

Thomas Wenzlaff

Beispiele zu Microsoft JScript

für den

Hobby-Programmierer



Hinweise zum Dokument

Autor: Thomas Wenzlaff
www.twseite.de

Stand des Dokumentes: Samstag, 24. März 2007

Rechtswahrung, Haftung und Verbesserungsvorschläge:

Das Dokument richtet sich an den Nutzerkreis der Hobby-Programmierer.

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen, Hardware- und Softwarebezeichnungen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und gehören - wenn eingetragen - dem jeweiligen Eigentümer, oder sollten als solche betrachtet werden.

Der Autor kann für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise ist der Autor dankbar.

Berlin, Samstag, 24. März 2007



Hinweise zur Kodierung von Quelltext in diesem Dokument

Die in diesem Dokument für die Kodierung von HTML-Quelltext verwendeten Einschachtelungen von HTML-Elementen wie Tags, Scripten usw. dienen ausschließlich der Übersichtlichkeit des Quelltextes.

HTML lässt Einschachtelungen wie in Java LEIDER nicht zu, da die Position der HTML-Elemente DIREKT in das Layout des HTML-Dokumentes übernommen werden.

Dazu ein Beispiel:

Für die Kodierung einer Tabelle wäre es übersichtlicher, jede Tabellenzelle als 1 Quelltext-Zeile zu kodieren. Eine Tabellenzelle wird syntaktisch an der Tag-Eingrenzung erkannt. Alle dazwischenliegenden Zeichen, also auch ein durch die Return-Taste (Enter-Taste) zum Zweck der Quelltextgestaltung erzeugter Zeilenumbruch, werden zum z.T. unerwünschten Zellinhalt. Leider ist es damit eventuell nötig, unübersichtliche Quelltextzeilen mit Tag-Folgen zu kodieren.

Wer strukturiert programmieren möchte, sollte sich mit JavaScript beschäftigen.



Inhaltsverzeichnis

1.	JAVASCRIPT ALS SPRACHE	19
1.1.	allgemeine Tipps	19
1.1.1.	Javascript-Version festlegen	19
1.1.2.	Verwendung JavaScript-Bibliothek (ab Javascript 1.1)	20
1.1.3.	Unterscheidung von Browsern, die Javascript können bzw. nicht können (SCRIPT> und <NOSCRIPT>)	20
1.1.4.	Feststellung des Dialektes zum aktiven Browser	21
1.1.5.	Browsererkennung beim Laden des HTML-Dokumentes	21
1.1.6.	Browsererkennung durch Ermittlung der Leistungsunterschiede	21
1.1.7.	Browsererkennung zur Unterscheidung browserinterner Objekte	27
1.2.	Variablen	33
1.2.1.	Variablen und ihre Gültigkeit	33
1.2.1.1.	Globale Variable	33
1.2.1.1.1.	Globale Variable Standard-Definition	33
1.2.1.1.2.	Globale Variable durch eine Funktion definieren	33
1.2.1.2.	Lokale Variable	34
1.2.2.	Array-Variablen	34
1.2.2.1.	Array anlegen und initialisieren	34
1.2.2.1.1.	eindimensionales Array	34
1.2.2.1.2.	mehrdimensionales Array	34
1.2.2.2.	Größe eines Arrays nach dem Anlegen ändern durch Hinzufügen von neuen Elementen	35
1.2.3.	Operationen mit Variablen	35
1.2.3.1.	numerische Variable	35
1.2.3.1.1.	berechnen	35
1.2.3.1.2.	prüfen	36
1.2.3.2.	Zeichenketten-Variable	36
1.2.3.2.1.	Zeicheneingabe prüfen ohne reguläre Ausdrücke (ohne reg-Methode)	36
1.2.3.2.2.	Teilstring suchen ohne reguläre Ausdrücke (ohne reg-Methode)	37
1.2.3.2.3.	Zeichenkette zu numerisch konvertieren	37
1.2.3.2.4.	Zeichenketten verknüpfen	37
1.2.3.2.5.	Dezimalwert zu Hexaziffern-Zeichenkette konvertieren	38
1.2.3.2.6.	Hexaziffern von bzw. zu numerisch	38
1.2.3.3.	Array-Variable	39
1.2.3.3.1.	Verketten von Arrayelementen des Zeichenkettentyps mit anderen Zeichenketten	39
1.2.3.3.2.	Arrayelemente nach deren Initialisierung erweitern (Methode join())	39
1.2.3.3.3.	Reihenfolge von Array-Elementen ändern	39
1.2.3.3.3.1.	umkehren (Methode reverse())	39
1.2.3.3.3.2.	Sortieren (Methode sort())	40
1.2.3.3.3.2.1.	nach dem Alphabet sortieren (Standardsortierung ohne Sortierfunktion)	40
1.2.3.3.3.2.2.	nach Zahlenwerten sortieren (Nicht-Standardsortierung mit Sortierfunktion)	40
1.3.	Verzweigungen und Schleifen	41
1.3.1.	if-else-Verzweigung	41
1.3.2.	switch-Verzweigung	42
1.3.3.	for -Schleife	42
1.3.4.	while-Schleife	44
1.3.5.	do-while-Schleife	44
1.3.6.	break-Anweisung für Schleifen	44
1.3.7.	continue-Anweisung für Schleifen	45
1.4.	Funktionen	45
1.4.1.	ohne Konstruktor (ohne Methode new)	45
1.4.2.	mit Konstruktor (mit Methode new)	46
1.4.2.1.	als Objekt anlegen	46
1.4.2.2.	als Objekt nachträglich erweitern (Prototyping)	46
1.4.3.	Übergabewerte an Funktion (Argumente)	46
1.4.4.	Rückgabewert einer Funktionen	47
1.4.4.1.	Rückgabe durch return	47
1.4.4.2.	Rückgabewert undefined durch void()	48
2.	BROWSER- UND SYSTEM-INFORMATIONEN (NAVIGATOR-OBJEKT)	48



2.1.	Browser-Informationen	48
2.1.1.	Name des Browsers	48
2.1.1.1.	Browserüberprüfung auf Microsoft Explorer ab 4.x	49
2.1.1.2.	Browserüberprüfung auf Netscape	49
2.1.2.	Codename Engine	49
2.1.3.	Browserversion: Hauptversion vor dem Punkt	49
2.1.4.	Browserversion: Updatenummer nach dem Punkt	49
2.1.5.	Useragent	49
2.1.6.	Sicherheitseinstellungen	49
2.1.7.	Browserplattform	51
2.1.8.	Java-Applet-Darstellungsmöglichkeit	51
2.2.	System-Informationen	51
2.2.1.	Betriebssystem unter dem der Browser läuft	51
2.1.2.	Prozessortyp	51
2.1.3.	Online- oder Offline-Zustand	51
2.1.3.1.	nicht Microsoft spezifisch	51
2.1.3.2.	Microsoft spezifisch	52
2.1.4.	User-Informationen	52
2.1.4.1.	Bildschirmauflösung des Users ermitteln	52
2.1.4.2.	User-Hinweis auf optimale Auflösung des HTML-Dokumentes erzeugen	52
2.1.4.3.	Url ermitteln, von der der User herkommt	52
3.	HTML-DOKUMENT	52
3.1.	Informationen ermitteln (location-Objekt)	52
3.2.	Möglichkeiten des Laden eines Dokumentes	53
3.2.1.	Laden eines fremden Dokumentes ohne Framedarstellung	53
3.2.2.	Eigenes Dokument wird durch fremde Webseite geladen	53
3.2.2.1.	CopyRight-Meldung auf fremden Host erzeugen	53
3.2.2.2.	Framedarstellung der eigenen Webseite sofort und ohne Bildschirmmeldung aktivieren	54
3.2.3.	Frame-Dokument mit zugehörigem Frameset laden	56
3.2.4.	Mehrere Frameinhalte gleichzeitig ändern	56
3.2.5.	Frame und Datenaustausch	57
3.2.6.	History-Eintrag ersetzen	57
3.3.	HTML-Code in einem Fenster erzeugen und zugleich ausführen	57
3.4.	Umleiten auf andere Url	59
3.5.	Update-Datum anzeigen	60
3.6.1.	für Microsoft Internet Explorer	61
3.6.2.	für Netscape	61
3.7.	Plugin-Prüfung	61
3.8.	E-Mail	62
3.8.1.	Email senden per Formular	62
3.8.2.	Spam-Schutz	67
3.8.3.	Email-Absendung mit Bestätigung für User	68
3.8.4.	Email und Stringkonvertierung ab Internet Explorer 5.5 (Encoding und Decoding URI's)	68
3.9.	Textdaten-Übergabe zwischen zwei HTML-Dokumenten	68
3.9.1.	Textdaten-Übergabe durch an Url angehängte Textdaten	68
3.9.2.	Textdaten-Übergabe durch Fenster-Handle	69
3.10.	Programmierung eines Java-Stacks zum temporären Zwischenspeichern von Daten in den RAM	71
3.11.	Microsoft Internet Explorer	71
3.11.1.	Abschaltung der automatischen Umrandung von angeklickten Objekten	71
3.11.2.	Bookmark per Script erzeugen	71
3.11.3.	Reload des Dokumentes nach Resize des Browserfenster	72
3.11.4.	Filter für HTML-Elemente ab IE 5.5	72
3.11.4.1.	Filterarten	72



3.11.4.2.	Prinzipien der Kodierung von Filtern	73
3.11.4.3.	Alpha-Filter für HTML-Grafik-Elemente ab IE 5.5	74
3.11.4.3.1.	Beispiel für transparentes INPUT-Element	74
3.11.4.3.2.	Beispiel für transparentes Vordergrund-Bild auf dem Hintergrundbild des Dokumentes	75
3.11.4.3.3.	Beispiel für transparente Überlagerung zweier Bilder	75
3.11.4.3.4.	Beispiele für Aus- und Einblende eines Bildes	75
3.11.4.3.5.	Beispiel für Aus- und Einblende von Bildern mit Bildwechsel	76
3.11.4.4.	Shadow-Filter für HTML-Grafik-Elemente ab IE 5.5	78
3.11.4.4.1.	Beispiel für Schatten eines Bildes	78
3.11.4.4.2.	Beispiel für 3D-Effekt eines Bildes	79
3.11.5.1.	Grafik zoomen ab IE 5.5	79
3.11.5.2.	Video abspielen	80
3.11.5.2.1.	Video abspielen per INPUT TYPE="image" (per IMG analog)	80
3.11.5.2.2.	Video und Sound abspielen per MediaPlayer	80
3.11.6.	PopUp-Fenster ab IE 5.5 per createPopup()	90
3.11.7.	Stackoperationen ab IE 5.5 für das Array-Objekt	91
3.11.8.	Auf ActiveX prüfen beim IE	91
3.11.9.	Linkrahmen unterdrücken beim IE	91
3.11.10.	Tooltip beim IE	91
3.11.11.	Webseite mit eigenem Icon ab IE 5.x	92
3.11.12.	Hintergrundbild beim IE als Wasserzeichen	92
3.11.13.	Uhr mit IE-Filter	92
3.12.	Verweilzeit des Users in Sekunden ermitteln	94
3.13.	History anwenden	94
3.13.1.	um 1 Seite zurück gehen	94
3.13.2.	um 1 Seite vorwärts gehen	94
3.13.3.	um mehrere Seiten wahlweise vorwärts bzw. rückwärts gehen	94
3.13.4.	History-Eintrag ersetzen	94
3.14.	Alle Textmarken (Anker) eines Dokumentes für deren Anwahl in einem Extra-Fenster anzeigen	94
3.15.	Inhalt des aktuellen Dokumentes als zusätzlichen Frame im Fenster darstellen	95
3.16.	Dokument nach dem Laden als oberstes Fenster anzeigen, also sichtbar	96
3.17.	Dokument drucken	96
3.18.	Dokument automatisch scrollen	97
3.19.	Reload des Dokumentes nach Resize des Browserfensters	98
3.20.	Festplattenverzeichnis unter Windows auslesen	98
3.21.	Dokumenten-Hintergrund	98
3.21.1.	Dokumenten-Hintergrund ein- bzw. ausblenden	98
3.21.2.	Dokumenten-Hintergrund auswählen	100
4.	FENSTER	100
4.1.	Textausgaben im aktuellen Fenster	100
4.1.1.	mit der Methode document.write()	100
4.1.2.	mit der Methode alert() (Alert-Fenster)	101
4.1.3.	schwebender Text von rechts nach links bei 640 Pixel Bildschirmbreite	101
4.1.4.	Fenster des Browser rausschieben und danach neu anzeigen	101
4.1.5.	Textstreifen senkrecht auf Bildschirm animieren	102
4.2.	Popup-Fenster per open()	110
4.3.	Nicht-Popup-Fenster per open()	110
4.3.1.	Fenster öffnen	110
4.3.1.1.	mit Standardgröße öffnen	110
4.3.1.2.	mit vordefinierter Größe öffnen	110



4.3.1.3.	ohne Eigenschaften eines Browserfensters öffnen	111
4.3.1.4.	Fenster, das andere Fenster aufruft, als oberstes also sichtbar anzeigen	111
4.3.2.	Fenster schliessen	111
4.3.2.1.	Fenster schließen, das nicht FRAME ist	111
4.3.2.2.	Fenster, das Frame ist: Oberstes Fenster schliessen	112
4.3.2.3.	Fenster schliesst sich selbst nach Wartezeit (Popup per open())	112
4.3.3.	Fensterauflösung ändern	112
4.3.4.	Fenstertitel nachträglich ändern	113
4.4.	Fenstertitelzeile mit scrollendem Text	113
5.	STATUSZEILE	115
5.1.	Standardtext in der Statuszeile	115
5.2.	Text Buchstabe für Buchstabe in Statuszeile anzeigen	115
5.3.	Automatisch wechselnder Text in der Statuszeile	115
5.4.	Scroller in der Statuszeile	116
5.5.	Text zu einem Hyperlink (HREF) für eine feste Zeitspanne in der Statuszeile anzeigen	117
5.6.	Text zu einem Hyperlink (HREF) für Überfahren mit Maus in der Statuszeile anzeigen bzw. mit fester Zeitspanne	118
5.7.	Hinweise zu Formularfelder in Statuszeile anzeigen	118
5.8.	Text in Statuszeile in fester Zeitspanne blinkend anzeigen	119
6.	FRAMES	119
6.1.	FRAMES füllen	119
6.2.	Reload eines Dokumentes mit allen seinen FRAMES	120
6.3.	Datei ohne FRAMESET in eine Datei mit FRAMESET laden	120
6.4.	Werbefenster als FRAME alle 3 Sekunden einblenden	120
6.5.	HTML-Inhalte von FRAMES tauschen	121
6.6.	Ein FRAME für mehrere HTML-Dokumente vom gemeinsamen Server	123
7.	Formulare	124
7.1.	Werte auslesen	124
7.1.1.	Auslesen eines Eingabefeldes	124
7.1.2.	Auslesen einer Checkbox	124
7.1.3.	Auslesen einer Auswahlliste	125
7.1.3.1.	Liste mit einer Auswahlmöglichkeit	125
7.1.3.2.	Liste mit mehreren Auswahlmöglichkeiten	125
7.2.	Werte zuweisen	126
7.3.	Werte überprüfen	126
7.3.1.	prüfen auf Leerkette	126
7.3.2.	prüfen auf Leerzeichen	127
7.3.3.	auf vorgegebene Zeichenanzahl prüfen	127
7.3.4.	auf Ziffern prüfen	127
7.3.5.	auf Integerwert aus einem Gültigkeitsbereich prüfen	127



7.4.	Werte konvertieren	127
7.4.1.	Numerischen Wert als Zeichenkette liefern und dabei Dezimalpunkt zu Dezimalkomma umwandeln	127
7.4.2.	Ziffern-Zeichenkettenwert nach numerisch und dabei Dezimalkomma zu Dezimalkomma umwandeln	127
7.4.3.	Vornull bei Ziffern-Zeichenkette, die numerisch <1 bzw. > -1 ist, vor dem Dezimalkomma, Dezimalpunkt oder Minus	128
7.5.	Formularprüfung für Submit	128
7.6.	Laufschrift in einem Formular	129
8.	MENÜ	129
8.1.	Menüleiste	129
8.1.1.	Realisierung einer Menüleiste mittels FRAMES	129
8.1.2.	Realisierung einer Menüleiste in einem separaten Fenster	129
8.2.	Formular-Text-Pulldown-Menü mit Button ein und ausschalten	130
8.3.	Formular-Text-Pulldown-Menü mit Aktivierung der Menüpunkte	130
8.4.	Formular-Option-Menü aus Hyperlinks	131
9.	BUTTONS	131
9.1.	Aussehen eine Button mit Mausklick verändern	131
9.1.	Aussehen eine Button-Objektes verändern	132
10.	MATHEMATISCHE BERECHNUNGEN (MATH-OBJEKT)	134
10.1.	Gibt den Wert von Math.PI aus	134
10.2.	Runden auf n Nachkommastellen	134
10.3.	Zufallszahl	134
10.4.	Größten gemeinsamen Teiler zweier ganzer Zahlen ab Javascript 1.2 per Rekursion ermitteln	135
10.5.	konvertieren zu ganzzahlig	135
10.6.	konvertieren zu ganzzahlig teilbar	135
10.7.	ganzzahlige Division	136
11.	DATENSTRUKTUREN	136
11.1.	HTML-Datenstrukturen anlegen und verwerten	136
11.1.1.	ausführliche Kodierung	136
11.1.2.	verkürzte Kodierung und Prototyping	137
11.2.	Datensätze als Teil einer Javascript Datei (*.js)	138
11.2.1.	*.js-Datei erzeugen	138
11.2.2.	*.js-Datei in das HTML-Dokument einbinden und auf die Daten zugreifen	138
11.2.3.	Beispiel Stichwortliste in obiger daten.js mit freier Suche nach Stichworten	139
11.3.	Programmierung eines Java-Stacks zum temporären Zwischenspeichern von Daten in den RAM	140
12.	EREIGNISSE (EVENTS)	140



12.1. Events allgemein	140
12.1.1. Der Event-Handler onAbort	140
12.1.2. Der Event-Handler onBlur	140
12.1.3. Der Event-Handler onChange	141
12.1.4. Der Event-Handler onClick	141
12.1.5. Der Event-Handler onError (Fehlermeldungen)	143
12.1.5.1. Fehlerbehandlung durch browsereigene Routinen abschalten	143
12.1.5.2. Fehler umleiten auf private Behandlungsroutine anstelle der browsereigenen Routinen	143
12.1.5.2.1. Fehlermeldung per alert()	143
12.1.5.2.2. Fehlermeldung in einem neuen Fenster	143
12.1.6. Der Event-Handler onFocus	144
12.1.7. Der Event-Handler onkeypress	145
12.1.8. Der Event-Handler onLoad	146
12.1.9. Der Event-Handler onMouseout	146
12.1.10. Der Event-Handler onMouseover	146
12.1.11. Der Event-Handler onReset	146
12.1.12. Der Event-Handler onSelect	147
12.2. Events im Netscape Communicator	147
12.2.1. Event onClick	147
12.2.2. Event onLoad	147
12.2.3. Das event-Objekt	148
12.2.4. Methode captureEvents()	148
12.2.5. Methode releaseEvents()	148
12.3. Events im Microsoft Internet Explorer	149
12.3.1. Das event-Objekt	149
12.3.2. Zusätzliche Event-Handler	149
12.3.3. Das Bubble-up Event	150
12.3.4. Das cancelBubble Event	150
13. REGULÄRE AUSDRÜCKE (REG-METHODE-SUCHALGORITHMEN)	150
13.1. Überprüfen einer E-Mail-Adresse	150
13.2. Genauere Überprüfung einer E-Mail-Adresse	151
13.3. Besondere Zeichen suchen	151
13.4. Teilketten suchen	152
13.5. Überprüfung einer Formular-Text-Eingabe	152
13.6. Anker suchen	153
14. TIMER, DATUM, UHRZEIT	153
14.1. Aktion nach einer Wartezeit starten	153
14.1.1. Erzeugt ein Popup-Fenster nach 10 Sekunden	153
14.1.2. Lädt eine WWW-Seite nach 10 Sekunden.	153
14.2. Tageberechnungen	153
14.2.1. Aktuellen Tag prüfen z.B. auf den 1. April	153
14.2.2. Wieviel Tage sind es noch bis Weihnachten?	154
14.3. Zeigt die Uhrzeit und das Datum an	154
14.4. Uhren anzeigen	156
14.4.1. Minuten-Eieruhr	156
14.4.2. Uhr im Formular anzeigen	157
14.4.3. Uhr in der Statuszeile	158



15. BILDER UND ANIMATIONEN 158

- 15.1. Bild preloaden (vorladen) in den Browser-Cache (ohne .readyState) 158
 - 15.1.1. Pixelgröße 1x1 als Platzhalter 158
 - 15.1.2. Mehrere Bilder mit Wartezeit preloaden 159
- 15.2. Zeigt zwei verschiedene Bilder abwechselnd an 159
- 15.3. Zeigt eine Bildfolge 160
- 15.4. Festes, nicht mitscrollendes Bild 161
- 15.5. Bild in einem neuen Fenster zentriert öffnen 165
- 15.6. Bild verschwinden lassen 167

16. SOUND 167

- 16.1. Dokument mit klingendem Sound laden am Beispiel Netscape 167
- 16.2. Sound für IE und NS 168

17. PASSWORD 169

- 17.1. Password in einer Variablen als Auswahlkriterium für eine HTML-Seite 169
 - 17.1.1. Eingabe per Formular 169
 - 17.1.2. Eingabe per Hyperlink (HREF) und prompt 170
 - 17.1.3. Zufallspasswort erzeugen 170
- 17.2. CGI-Script eines Providers, das auf dessen Server läuft, für eine geschützte Homepage auf gleichem Server 171

18. COOKIES 171

19. TASTATUR UND MAUS 173

- 19.1. Browserspezifische Ermittlung EINER eingegebenen Taste in einem Textarea 173
- 19.2. Rechte Maustaste belegen 174
- 19.3. Mauscursor permanent um ein mitlaufendes Bild erweitern 175
- 19.4. Mauscursor als Bild anhand cur-Datei erweitern 177
- 19.5. Mauscursor mit Uhr und Datum 177

20. STYLE SHEETS UND DHTML 186

- 20.1. Cascading Style Sheets (CSS) kodieren 186
 - 20.1.1. CSS-Klassen erzeugen 186
 - 20.1.2. Tag-unabhängige Kodierung (mit ID) 187
 - 20.1.2.1. allgemein 187
 - 20.1.2.2. mit Tag 187
 - 20.1.2.3. mit <DIV> Tag 188
 - 20.1.2.4. mit <Hx> Tag 188
 - 20.1.2.5. Clipping (Ausschnitt erzeugen) 190
 - 20.1.2.6. z-Index 193
 - 20.1.3. Tag-abhängige Kodierung (ohne ID) 194



20.1.3.1.	Style Sheet Definition direkt in einem Tag	194
20.1.3.2.	mit <LINK> Tag	194
20.2.	Das Layer-Objekt im Netscape Communicator	194
20.2.1.	Layer-Objekt erzeugen	194
20.2.2.	Layer-Objekt bewegen	196
20.2.3.	Die Layer-Eigenschaft document	196
20.2.4.	JavaScript-Code und das Layer-Objekt	197
20.2.5.	Laden neuer HTML-Seiten in ein Layer-Objekt	197
20.2.6.	Layer-Objekt clipping	198
20.2.7.	Verschachtelte Layer-Objekte mit dem Layer()-Konstruktor	199
20.2.8.	Das <LAYER>-Tag	199
20.3.	Das style-Objekt im Microsoft Internet Explorer	201
20.3.1.	style-Objekt erzeugen	201
20.3.2.	style-Objekt bewegen	201
20.3.3.	Instant Page Update	202
20.3.4.	style-Objekt clipping	202
20.4.	Unterscheidung Layer und Style je nach aktuellem Browser	203
21.	JAVA-JAVASCRIPT KOMMUNIKATION	205
21.1.	Gibt einen Text auf der Java Konsole aus	205
21.2.	Zeigt Informationen zum Betriebssystem und den Java-Klassen	206
21.3.	Programmierung eines Java-Stacks zum temporären Zwischenspeichern von Daten in den RAM	206
22.	TEXT	207
22.1.	Text-Kapitälchen	207
22.2.	Textmeldung in Statuszeile, wenn Maus über einen Link (Verweis) fährt	207
22.3.	Textticker	208
23.	SPIELE	209
23.1.	Break Out	209
24.	ACTIVEX-CONTROLS	231
24.1.	Analog-Uhr	237
24.2.	Video und Sound abspielen per Mediaplayer	250
25.	SONSTIGES	253
25.1.	Kalender	253
25.2.	Aktive Inhalte im Internet Explorer	262
25.3.	Aktive Inhalte im per .replace() geöffnetem Dokument	262
25.4.	ID-Attribut und .createElement()	263
25.5.	.focus() beim IE 7	264
(TWS)	Microsoft JScript -Beispiele für den Hobby-Programmierer	24.03.07



(TWS)	Microsoft JScript -Beispiele für den Hobby-Programmierer	24.03.07	Seite 12 von 311
25.6.	window.status beim IE 7		265
25.7.	Renderqualität und animierte Gif beim IE 7		265
25.8.	.createPopup() und Popupblocker beim IE 7		265
25.9.	.style.zIndex nach createElement() des Objektes mit .style im IE 6		266
25.10.	.eval()		266
25.11.	wichtige Events		268
25.12.	for-Schleife - Schleifenzähler manipulieren		283
25.13.	Funktion		285
25.14.	js-Datei dynamisch einbinden		286
25.15.	Anzahl der Tage zu einem Monat ermitteln		287
25.16.	Zeichen kodieren zur Form &#xx; erzeugen		288
25.17.	ASCII-Tabelle anzeigen		290
25.17.	Primzahlen ermitteln		290
25.17.1	Primzahlen ermitteln langsame Variante		290
25.17.1	Primzahlen ermitteln schnelle Variante		294
25.18.	Formatierte HTML-Text-Erzeugung		298
26.	INDEX		304



Hinweise zum Einrichten eines virtuellen Hosts per Apache-HTTP-Server:

Anstelle des HTTP-Webservers vom Internetprovider kann der eigene PC als Test-Server fungieren, um z.B. die Webseite so zu testen, als wäre sie gerade online auf dem Webserver.

Der Hobbyprogrammierer will u.a. kostengünstig testen, also bieten sich die kostenlosen HTTP-Server an, die einen virtuellen Host anhand eines beliebigen Ordners auf der Festplatte erzeugen können, wobei Local Host (127.0.0.1) als virtueller Host einrichtbar sein muss (der nicht von der Firewall-Software des PC allein verwaltet sein darf) und die zu testende Webseite im Browser per Domainnamen aktivierbar sein muss (anstelle der Eingabe von 127.0.0.1). Man google, um festzustellen, welche Produkte diesen Kriterien entsprechen. - Abkürzend fällt die Wahl nicht zufällig auf den HTTP-Server von Apache (www.apache.org), welcher ziemlich schwierig zu konfigurieren ist, wenn mehr als nur localhost genutzt werden soll (Für Hobbyzwecke reicht localhost aus).

Der Hobbyprogrammierer, welcher für den Microsoft Internet Explorer (ab IE 7 heißt der Windows Internet Explorer) in seinen Varianten je nach Windows-Version programmieren will, wird definitiv folgende Probleme bekommen:

Microsoft lässt u.a. die Installation des Internet Explorers in parallelen Versionen nicht zu, obwohl Browserversionen nachweislich nicht kompatibel sind und Microsoft Browserversionen innerhalb Windowsversionen supportet werden. Daher benötigt man pro Version des Internet Explorers eine Windows-Installation. Pro Windows-Installation wird eine Windows-Lizenz fällig, auch wenn auf anderer Festplatte am ansonsten identischen PC installiert wird (Windows-Online-Update erkennt solche Doppelversionen und verweigert den Support). Auch wer Windows unter VM emulieren will (oder auf Apple mit Intel-Technik Windows parallel mit Apple nutzen will), benötigt eine Lizenz. Mit anderen Worten: Auch wenn die Mehrfachinstallation nicht parallel nutzbar wäre, sondern immer nur genau 1, will Microsoft Geld haben - nicht umsonst ist Microsoft-Chef so beliebt wie reich und nicht umsonst migrieren immer mehr IT-Anwender zu Linux-Derivaten, die fast identische Browser bezüglich Windows haben.

Ergo, der Hobbyprogrammierer wird wohl sämtliche Bekannte mit Testwünschen nerven, oder illegal testen, oder auf andere Browser-Hersteller und deren HTML- sowie Script-Versionen ausweichen, die nicht nur Parallelinstallationen des Browsers zulassen (z.B. Opera unter Windows), sondern auch noch ziemlich gut kompatibel sind (weil identische Scriptmaschine nutzend).

Dass Microsoft eventuell keinen kostenlosen HTTP-Server anbietet (der ansonsten auch noch Javascript, Active-X-Control- Aufrufe per Script, DirectX-Zugriff kennen müsste), ist das kleinere - vor allem lösbarere Problem: Eben ohne Microsoft.

Pfade für Dateien im Script einer Webseite, die per HTTP-Server oder lokal von Festplatte gestartet wird:

Relative Pfade per '.' sind unabhängig davon, wie die Wurzel (Root) der Webseite heißt.
Absolute Pfade sind abhängig davon, wie die Wurzel (Root) der Webseite heißt.

Wird die Webseite (z.B. www.twseite.de) von einem HTTP-Server gestartet, dann sind absolute Pfade bezüglich <http://www.twseite.de> möglich, wobei <http://www.twseite.de> die Root der Webseite darstellt.

(Achtung: Auf einem HTTP-Server des Internetproviders liegt die Webseite natürlich auch auf einer Festplatte, also dort in einem Ordner. Dieser Ordner muss also mit www.twseite.de logisch verbunden sein - wie, das teilt der Internet-Provider mit.

Tipp: Namen des Ordners auf der Festplatte des Internet-Providers kann man genauso nennen wie den des Ordners auf der lokalen Festplatte des PC. Ordernamen mit Pfadzeichen wie '.' oder '/' sind natürlich nicht erlaubt)

Wird die Webseite (z.B. www.twseite.de) von lokaler Festplatte aus gestartet, dann sind absolute Pfade bezüglich <http://www.twseite.de> nicht möglich, da es solchen Ordernamen nicht geben kann.
Festplattenordner möglich z.B. c:\twseite\

Will man die Webseite identisch verwalten, egal ob man die Webseite von einem HTTP-Server oder von lokaler Festplatte aus startet, dann schaue man sich folgendes Beispiel für www.twseite.de an, das allerdings mit JavaScript oder JScript realisiert wird. Der Festplattenordner ist c:\twseite.

```
var BrowserAufOnlinePruefen=true;           // false für Browser nicht auf online prüfen
                                           // online: Webseite wurde auf HTTP-Server aktiviert
var DomainOhneHTTP='www.twseite.de';       // Host der Webseite ohne http:// und ohne Port
var BrowserIstOnline=false;                 // Annahme: Browser ist nicht online
var PfadDerDateien="";                      // Annahme: Browser ist nicht online also
                                           // alle Pfade unterhalb von c:\twseite\
                                           // wobei die Startdatei index.html der
                                           // Webseite eben in diesem Ordner liegt

if(BrowserAufOnlinePruefen)                 // wenn auf online geprüft werden soll
{if(window.location.hostname!=null);        // aktuell gefundener Host
 BrowserIstOnline=(window.location.hostname== DomainOhneHTTP);}
```



```

// Host www.xxx.yyy prüfen auf aktuelle gefundenen Host
// true so Browser online
}

if(BrowserIstOnline) {PfadDerDateien="http://" + DomainOhneHTTP;}
// Pfad wenn Browser online ist: Alle Pfade unterhalb von
// http://www.twseite.de
// Für den HTTP-Server liegt die Webseite natürlich
// auf einer Festplatte, also dort in einem Ordner.
// Jeder Pfad in der Webseite wird in den Ordnerpfad
// der Festplatte automatisch konvertiert (HTTP zu
// Festplatte per Dienst: Daher der Name HTTP-
// Server)

```

Nachfolgend die beispielhafte Einrichtung einer Webseite per Apache-HTTP-Server 2.2.2 bis 2.2.4. unter Windows XP ab SP 1:

Webseitendomain heißt www.twseite.de mit index.html als Startdatei
 Festplattenordner der Domain, die als virtueller Host über localhost (also 127.0.0.1) laufen soll
 c:\twseite\
 Apache wurde installiert unter e:\wsp\apache\
 wobei gelten muss
 DNS-Dienst von Windows muss aktiv sein (in der Regel ist der Dienst automatisch aktiv)
 Installationstyp All Users, on Port 80, as a Service - Recommended
 Domain und Servername 127.0.0.1 (nicht localhost kodieren)
 Email beliebig@localhost
 Mit der Installation wird der Dienst-Monitor von Apache bei Windowsstart ebenfalls starten.
 Es wird Apache als permanenter Dienst eingerichtet, der über den Dienst-Monitor von Apache
 aktivierbar / deaktivierbar ist. Es sind zwar mehrere Apache-Dienste einrichtbar, aber genau 1
 kann nur immer aktiv sein.

Apache-Software einrichten:

Dienst-Monitor von Apache: siehe oben
 aber: Da in Windows für jeden Aufruf der Webseite www.twseite.de unter Apache eine
 Einstellung getroffen werden muss, empfiehlt es sich, den Apache-Dienst
 grundsätzlich manuell zu starten und zu stoppen anhand nachfolgend
 vorgestellter BAT-Dateien.
 Dafür muss aber einmalig folgendes eingestellt werden:
 Der Apache-Dienst-Name ist per Dienst-Monitor zu sehen (und der Status ob
 aktiv oder deaktiv).
 Dienste sind unter Windows per Systemsteuerung-Verwaltung-Dienste
 verwaltbar (auch per Apache-Monitor ist die Dienstverwaltung
 aktivierbar).
 Der Apache-Dienst muss zuerst deaktiviert werden, dann auf Starttyp manuell
 gesetzt werden: Mit Windows-Start startet der Dienst nicht automatisch.

Windows anpassen an den Virtuellen Host

Die Anpassung erfolgt so, dass Apache per Batch-File, die unten erklärt werden, gestartet
 und deaktiviert wird (Batch-Files sind passend zur Anpassung von Windows).

Unter c:\windows\system32\drivers\etc\ liegt die Datei hosts

Copyright (c) 1993-1999 Microsoft Corp.

```

#
# Dies ist eine HOSTS-Beispieldatei, die von Microsoft TCP/IP
# für Windows 2000 verwendet wird.
#
# Diese Datei enthält die Zuordnungen der IP-Adressen zu Hostnamen.
# Jeder Eintrag muss in einer eigenen Zeile stehen. Die IP-
# Adresse sollte in der ersten Spalte gefolgt vom zugehörigen
# Hostnamen stehen.
# Die IP-Adresse und der Hostname müssen durch mindestens ein
# Leerzeichen getrennt sein.
#
# Zusätzliche Kommentare (so wie in dieser Datei) können in

```



```
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,
# aber müssen mit dem Zeichen '#' eingegeben werden.
#
# Zum Beispiel:
#
#      102.54.94.97      rhino.acme.com      # Quellserver
#      38.25.63.10      x.acme.com          # x-Clienthost
127.0.0.1      localhost
```

Diese Datei verwaltet -wie man sieht - auch den localhost.

Standardgemäß ist localhost auf 127.0.0.1 gelegt (nur deswegen sind localhost und 127.0.0.1 synonym)

Soll aber die Webseite www.twseite.de über 127.0.0.1 getestet werden, muss also die Zeile

```
127.0.0.1      localhost
```

ersetzt werden durch

```
127.0.0.1      www.twseite.de
```

Nach dem Test muss localhost wieder für 127.0.0.1 verfügbar gemacht werden, also der Standard gesetzt werden.

Genau dieses Ersetzen machen die Batch-Files siehe unten. Dafür benötigen sie nur 2 neue Ordner, die einmalig mit Inhalt manuell angelegt werden müssen in c:\windows\system32\drivers\etc\

Schritt 1: Ordner c:\windows\system32\drivers\etc_hosts_standard\

manuell erzeugen, und dorthin die bisher unveränderte, also originale hosts-Datei kopieren (Datei enthält 127.0.0.1 localhost)

```
# Die IP-Adresse und der Hostname müssen durch mindestens ein
# Leerzeichen getrennt sein.
#
# Zusätzliche Kommentare (so wie in dieser Datei) können in
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,
# aber müssen mit dem Zeichen '#' eingegeben werden.
#
# Zum Beispiel:
#
#      102.54.94.97      rhino.acme.com      # Quellserver
#      38.25.63.10      x.acme.com          # x-Clienthost
127.0.0.1      localhost
```

Schritt 2 Ordner c:\windows\system32\drivers\etc_hosts_mit_www_twseite_de\

manuell erzeugen, und dorthin die bisher unveränderte, also originale hosts-Datei kopieren

(Datei enthält 127.0.0.1 localhost)

dann diese Datei in diesem Ordner per notepad.exe (nicht Word etc.) auf

```
127.0.0.1      www.twseite.de
```

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Dies ist eine HOSTS-Beispieldatei, die von Microsoft TCP/IP
# für Windows 2000 verwendet wird.
#
# Diese Datei enthält die Zuordnungen der IP-Adressen zu Hostnamen.
# Jeder Eintrag muss in einer eigenen Zeile stehen. Die IP-
# Adresse sollte in der ersten Spalte gefolgt vom zugehörigen
# Hostnamen stehen.
```



```
# Die IP-Adresse und der Hostname müssen durch mindestens ein
# Leerzeichen getrennt sein.
#
# Zusätzliche Kommentare (so wie in dieser Datei) können in
# einzelnen Zeilen oder hinter dem Computernamen eingefügt werden,
# aber müssen mit dem Zeichen '#' eingegeben werden.
#
# Zum Beispiel:
#
#      102.54.94.97      rhino.acme.com      # Quellserver
#      38.25.63.10      x.acme.com          # x-Clienthost
127.0.0.1      www.twseite.de
```

Virtual-Host genau 1x einrichten in e:\wpx\apache\conf\ dort in der Datei httpd.conf

erst httpd.conf kopieren in einen Sicherungsordner freier Wahl

dann per notepad.exe (nicht per Word etc.) am Ende der httpd.conf folgenden Text einfügen

```
Achtung: Es muss natürlich dann
www.twseite.de      ersetzt werden durch zu
                    testende Domain
c:/twseite          ersetzt werden durch den
                    wirkliche Pfad
```

Virtual Host für www.twseite.de auf 127.0.0.1:80 gehostet

ERST Virtual Host für den Server, der damit alle Servernamen ungleich
www.twseite.de abfängt

also Angaben aus Serverinstalltion verwendet

```
NameVirtualHost 127.0.0.1
<VirtualHost 127.0.0.1>
    ServerName localhost
    ServerAlias 127.0.0.1
    DocumentRoot e:/wpx/apache/htdocs
    ErrorLog e:/wpx/apache/logs/error.log
    TransferLog e:/wpx/apache/logs/access.log
    ScriptAlias /cgi-bin/ e:/wpx/apache/cgi-bin/
</VirtualHost>
```

Virtual Host für www.twseite.de

```
<VirtualHost 127.0.0.1>
    ServerName www.twseite.de
    ServerAlias 127.0.0.1
    DocumentRoot c:/twseite
    DirectoryIndex index.html
    ErrorLog c:/twseite/apache_error.log
    TransferLog c:/twseite/apache_access.log
    <Directory c:/twseite>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Apache-Start und Stop per folgender BAT-Dateien, die z.B. in C:\ liegen (egal wo auf Festplatte):

Achtung: Falls der Start von Apache wegen nicht vorhandenem Dienst nicht erfolgen kann, dann gilt:

Mit der Apache-Installation wurde ein Dienst eingerichtet: siehe oben. Nur leider, der



Apache-Start verlangt einen weiteren Dienst mit anderem Namen. Da bekannt ist, dass aber nur 1 Dienst zu jedem Zeitpunkt aktiv sein kann, ist es sehr verwunderlich, wieso Apache-Start den bereits vorhandenen Dienst nicht nimmt.

Anstelle von wundern bitte folgendes genau 1 mal ausführen:

```
e:\wsp\apache\bin\ httpd.exe -k install
```

im bin-Verzeichnis liegen die ausführbaren Dateien von Apache.

httpd.exe ist die Hauptkomponente von Apache.

-k install installiert einen weiteren Apache-Dienst.

Logischerweise ist dieser Dienst analog zum Dienst, der bei Apache-Installation erzeugt wurde, ebenfalls per Dienste-Verwaltung zu deaktivieren und auf Starttyp manuell zu setzen. Dieser neue Dienst hat aber den passenden Namen, der vom Start von Apache akzeptiert wird.

Hier ein Beispiel für den Dienste-Wirrwarr:

Apache-Installation erzeugt Dienst 'Apache2.2'.

Apache-Start will aber den Dienst 'Apache 2'

Batch-File 'ApacheStarten.bat' für Start des Apache, wobei der Dienst 'Apache2' aktiviert wird

```
@echo off
cls
echo Apache als Anwendung starten bei inaktivem Apache-Dienst 'Apache2'
echo          (Apache-Dienst darf nicht bei Windows-Start aktiv sein)
echo          (Apache stoppen immer per ApacheStoppen.bat)
echo hosts-Datei mit www.twseite.de bereitstellen
echo.
copy /V /Y C:\WINDOWS\system32\drivers\etc\_hosts_mit_www_twseite_de\hosts
C:\WINDOWS\system32\drivers\etc\hosts > NUL
e:\wsp\apache\bin\httpd.exe -k start
echo.
echo.
echo      ..... Apache-Anwendung wurde nur gestartet, wenn oben keine
echo              Fehlermeldung angezeigt wurde !
echo              Falls Fehlermeldung dann im Monitor Apache Servers
echo              zu allen Eintraege STOP-Button klicken (falls klickbar)
echo              und Batch-File neu starten
echo.
echo.
echo      Apache kann keine ActiveX durchreichen (z.B. Logox WebSpeech)
echo              per JScript-erzeugte Popups nicht immer korrekt rendern
echo.
echo.
pause
echo.
echo.
echo aktuelle hosts-Datei lautet
echo.
type C:\WINDOWS\system32\drivers\etc\hosts
echo.
pause
```

Batch-File 'ApacheStoppen.bat' für Start des Apache, wobei der Dienst 'Apache2' de-aktiviert wird

```
@echo off
cls
echo Apache als Anwendung stoppen bei aktiven Apache-Dienst 'Apache2'
echo          (Apache-Dienst aktiviert per ApacheStarten.bat)
echo hosts-Datei ohne www.twseite.de bereitstellen
echo.
```



```
copy /V /Y C:\WINDOWS\system32\drivers\etc\_hosts_standard\hosts
C:\WINDOWS\system32\drivers\etc\hosts >NUL
e:\wsp\apache\bin\httpd.exe -k stop
echo.
echo.
echo      .... Apache-Anwendung wurde nur gestoppt, wenn oben keine
echo              Fehlermeldung angezeigt wurde !
echo      Falls Fehlermeldung dann im Monitor Apache Servers
echo              dann war kein Appache-Server aktiv.
echo.
echo.
pause
echo.
echo.
echo aktuelle hosts-Datei lautet
echo.
type C:\WINDOWS\system32\drivers\etc\hosts
echo.
pause
```

Batch-File 'ApacheHostsDateiAnzeigen.bat' für Anzeige der hosts-Datei

```
@echo off
cls
echo aktuelle hosts-Datei anzeigen
echo.
type C:\WINDOWS\system32\drivers\etc\hosts
echo.
echo.
echo.
pause
```

Webseite aktivieren per Apache:

Da die Webseite nur per o.g. Batch-Dateien gestartet bzw. gestoppt werden kann, kann der Apache-Dienst-Monitor nur noch zur Kontrolle benutzt werden, ob eine Apache-Dienst aktiv ist.

Schritt 1: Vor dem Start der Webseite darf kein Apache-Dienst aktiv sein

Schritt 2 ApacheHostsDateiAnzeigen.bat aktivieren. Die hosts-Datei muss enthalten

127.0.0.1 localhost

Schritt 3: ApacheStarten.bat aktivieren. Apache startet.

Wer will kann jetzt Schritt 2 wiederholen und sieht dann

127.0.01. www.twseite.de

Wenn jetzt die Firewall-Software sich meldet, dann Apache erlauben
am Port 80 (also HTTP) von 127.0.0.1

Achtung: Nutzt die zu testende Webseite andere Ports z.B. die von
Plugins, dann können die Plugins selbst einen
Firewalleintrag verlangen, aber Apache kann natürlich
nur HTTP (Port 80) und muss nur zwischen Webseite
und den Plugins vermitteln (eben über Port 80)

Schritt 4: Browser der Wahl aktivieren

Dort die Domain www.twseite.de eintippen die Webseite öffnet sich anhand
c:\twseite\index.html

Tipp: Sollte die Webseite bereits auf einem Webserver online sein, dann würde
ohne Apache-Start natürlich die online-Variante aktiviert werden
und nicht die Webseite von der Festplatte aus dem Ordner c:\twseite !



Wichtig: Sollte die Webseite im Ordner c:\twseite\ geändert worden sein, so muss der Browser-Cache gelöscht werden, bevor die die Webseite erneut aktiviert wird. Apache lässt den Browser genauso agieren, als ob der online die Webseite laden würde, also über den Browser-Cache. Manche Browser oder Webseiten unterbinden dann das Neuladen einer Datei, die im Browser-Cache dem Namen nach vorhanden ist, aber in c:\twseite verändert vorliegt also zwingend neu geladen werden muss.... Ergo Browser-Cache vorher löschen.

Browser-Cache im löschen im
Internet Explorer per Internetoptionen, die
im Browser per Menüpunkt Extra oder per Systemsteuerung
aktivierbar sind (ev. Verknüpfung auf Desktop manuell
erzeugen)
Opera per Menüpunkt Extra-Einstellungen-Erweitert

Schritt 5: Webseite getestet (man hat jetzt ein Smily-Gesicht oder graue Haare mehr, oder den Chef, Kunden, Frau am Hals *g)
also Zeit zum Deaktivieren von Apache per ApacheStoppen.bat

Schritt 6: Webseitendaten aus c:\twseite sichern.

Errormeldungen des HTTP-Servers per .htaccess-Datei verwalten:

HTTP-Server-Fehlercode auf die der Webmaster reagieren kann per .htaccess

xxx	Meldung	Bedeutung
204	No Content	Das angeforderte Dokument enthält keine Daten
301	Moved Permanently	Das angeforderte Dokument wurde verschoben
302	Moved Temporarily	Das angeforderte Dokument wurde zeitweise verschoben
400	Bad Request	Zugriff des Client auf Server ist unbekannt
401	Unauthorized	Zugriff des Client auf Server nicht erlaubt
404	Not Found	Das angeforderte Dokument ist nicht vorhanden
408	Request Time Out	Antwort des Servers hat Zeitüberschreitung
500	Internal Server Error	Server hat internen Fehler
501	Not Implemented	Zugriff des Client auf Server nicht ausführbar
503	Service Unavailable	Server ist nicht zu erreichen

Für jeden der obigen Fehler kann z.B. in die Root der Webseite ein HTML-Dokument hinterlegt werden, dass bei auftreten des Fehler aktiviert wird, den Fehler beschreibt und /oder Ersatzreaktion durchführt.

in .htaccess muss Zeile hinterlegt werden

ErrorDocument xxx /yyy.html

xxx siehe oben

yyy freier Dateiname in der Root, Achtung: Datei MUSS vorhanden sein !!!
Pfade auch möglich bezüglich Root /

Es ist auch möglich

ErrorDocument xxx "freier Text"

1. Javascript als Sprache

1.1. allgemeine Tipps

1.1.1. Javascript-Version festlegen

<HTML>

Dies ist normales HTML.

<SCRIPT LANGUAGE="JavaScript1.1">

<!--

document.write("Hallo Welt! Dies ist JavaScript 1.1");

//-->

</SCRIPT>

Wieder normales HTML.



</HTML>

Browser führt NUR das Script aus, dessen Version er kennt und belegt so die Variable korrekt.
Es wird angenommen, dass jeder Browser Javascript 1.0 kann

```
<SCRIPT LANGUAGE="JavaScript">
<!--
// Javascript 1.0 → wenn Browser 1.0 kann, führt er nachfolgende Anweisung aus
var script_version="0"; // können alle Browser
// -->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript1.1">
<!--
// Javascript 1.1 → wenn Browser 1.1 kann, führt er nachfolgende Anweisung aus
script_version="1"; // enthält auch Version 1.0
// -->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript1.2">
<!--
// Javascript 1.2 → wenn Browser 1.2 kann, führt er nachfolgende Anweisung aus
script_version="2"; // enthält auch Version 1.1
// -->
</SCRIPT>
```

script_version kann für die browser-spezifische Variantenkodierung von Javascript verwendet werden

1.1.2. Verwendung JavaScript-Bibliothek (ab Javascript 1.1)

Es wird die Javascript-Datei test.js mit folgendem Inhalt eingebunden:

```
document.write("Es funktioniert!<BR>");
document.write("Der MIME-Type ist gesetzt.");
```

Hier wird test.js eingebunden:

```
<HTML>
  Wenn Sie die folgenden Zeilen nicht
  sehen, dann ist der MIME-Type auf
  Ihrem Server nicht richtig eingestellt:
  <P>
  <SCRIPT SRC="test.js">
  </SCRIPT>
</HTML>
```

1.1.3. Unterscheidung von Browsern, die Javascript können bzw. nicht können (<SCRIPT> und <NOSCRIPT>)

```
<HTML>
  Dies ist normales HTML.<BR>

  <SCRIPT LANGUAGE="JavaScript">
  <!--
      vor alten Browser verstecken
      document.write("Hallo Welt! Dies ist JavaScript.");
  //-->
  </SCRIPT>

  <NOSCRIPT>
      Sie haben keinen JavaScript-fähigen Browser oder JavaScript ist deaktiviert !
  </NOSCRIPT>
  <BR>
  Wieder normales HTML.<BR>
</HTML>
```

Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden.



Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.

Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein, innerhalb dessen dann die neuen HTML-Elemente erzeugt werden.

1.1.4. Feststellung des Dialektes zum aktiven Browser

Netscape- und Microsoft-Browser interpretieren auf eigene Weise (z.T. sich gegenseitig ausschliessend)

1.1.5. Browsererkennung beim Laden des HTML-Dokumentes

Variante 1:

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
    var ie4=0;
    if ( (navigator.appName.indexOf("Explorer")>=0)
        && (navigator.appVersion.indexOf('4.0') >=0)
        )
      ie4=1;
  // -->
</SCRIPT>
```

für Netscape bitte "Netscape" verwenden und entsprechende Version

Variante 2:

```
<SCRIPT ...>
<!--
  var browser_typ="";
  var netscape_name="Netscape";
  var microsoft_name="Microsoft Internet Explorer";

  if (navigator.appName == netscape_name) {browser_typ="NN";}

  if (navigator.appName==microsoft_name) {browser_typ="IE";}

  if (browser_typ != "") {browser_typ+=navigator.appVersion.charAt(0);}
  // -->
</SCRIPT>
```

Variante 3:

```
<SCRIPT ...>
<!--
  var ns = document.layers ? true : false;
  var ie = document.all ? true : false;
  // -->
</SCRIPT>
```

1.1.6. Browsererkennung durch Ermittlung der Leistungsunterschiede

Bsp: Netscape kennt Layers; aber Explorer nicht (Achtung: Layers sind NICHT mehr im HTML-Standard !!!)

also Abfrage if (document.layers)	wenn true, so NUR Netscape;
	sonst irgendwas aber nicht gesichert der Explorer
also Abfrage if (document.all)	wenn true, so NUR Explorer;
	sonst irgendwas aber nicht gesichert der Netscape

Microsoft ändert fortlaufend die Active-X-Eigenschaften von Windows und somit auch des Internet Explorers

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.

Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.

Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Lizenzierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks. Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.



Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.

Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2). Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.

Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

Hinweis: Man suche doch mal im Internet nach einem kostenlosen HTTP-Server vom Microsoft, um IE-Seite testen zu können, die JScript nutzen (inklusive Debugger). Denn sollte kein kostenloses Angebot findbar sein, kommen die Kosten von Entwicklungssoftware zum IE hinzu. Ein Blick auf Konkurrenz-HTTP-Server klärt die Sachlage: Apache-HTTP-Server ist kostenlos, allerdings nicht einfach einzurichten (Hinweis: Der HTTP-Server sollte virtuelle Hosts einrichten können und korrekt mit der Firewall des Users zusammenarbeiten können).

Abänderungen wegen Sicherheitspatches der jeweiligen Windows-Versionen

Abschaltungen von Active-X-Controls erfolgen auch im Rahmen der Sicherheitspatches zu Windows-Versionen.

Es ist auch möglich, dass wegen Sicherheitslücken abgeschaltet wird und somit Komponenten einer Webseite je nach Windowsversion nicht mehr laufen.

Im Rahmen der Sicherheitspatches ist es Microsoft sogar gelungen, Webseiten, die den MS-Encoder zur Komprimierung von HTML- und JScript-Code nutzen, schlagartig unnutzbar zu machen: Ein Bug in einem Patch zu Windows XP - Q918899 Das Patch verursacht IE-Browser-Absturz bei per MS ScriptEncoder gepacktem JScript unter SP1 und 2 wenn HTTP 1.1 mit Kompression genutzt wird z.B. bei onclick-Handler auf IMG klick ins Fenster per aktivem Popup

Der Absturz ist "read" -Fehler von immer ein und derselben Speicherstelle.

User, die dieses Patch installiert haben, können ab sofort keine IE-Seiten mit codiertem Script mehr ansehen.

Microsoft stellt Abhilfe nach geraumer Zeit zur Verfügung, jedoch spezifisch nach Windows XP-Version:

Patch Q918899 für

Windows XP SP1
SP2

Download für jedermann bereitgestellt
nur auf kostenpflichtige telefonische Anfrage des Users per Downloadlink bereitgestellt, da Microsoft explizit die User registriert haben will, bei denen das Patchproblem auftritt (User muss sich Telefonnummer besorgen)
Solange also das Patch zum fehlerhaften Patch vom User nicht installiert wird, z.B. weil der User keine Ahnung hat, dass und wo er sich die Telefonnummer von Microsoft besorgen muss bzw. zu besorgen hat, wird der User IE-Seiten mit komprimierten Code dauerhaft nicht nutzen können. (Microsoft-Support ist z.T. nur in Englisch).

Abänderungen wegen Browser-Inkompatibilität

Popupblocker-Fehler

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()

Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.

Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.

Der Fehler tritt nicht auf, wenn ein Fenster per window.open() erzeugt wurde.

Bedingung:

Scriptfehleranzeige ist erlaubt im IE 7

Popupblocker ist im IE abgeschaltet

ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 window.popup per .show() erzeugt.

ein weiteres Fenster (Register) z.B. leere Seite (about:blank)

beide (Register) liegen in einer gemeinsamen IE-Instanz

Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per .show() erzeugt,

bricht der Browser das Dokument mit .show() ab (Scriptfehler).

Der Popupblocker für die leere Seite verursacht den Programmfehler im Dokument mit .show(). Es wird folgende Meldung angezeigt (in der Informationsleiste):

'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'

Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:

Der Popupblocker hat ein Populfenster geblockt. Sie können den Popupblocker deaktivieren oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.

Die Realität zur obigen Meldung ist völlig anders:

Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter

Popupblocker einschalten

weitere Informationen

jedoch keine Möglichkeit wie laut Bedeutung

Damit gilt: Der abgeschaltete Popupblocker ist in Wirklichkeit aktiv.

Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster



einer Windowsanwendung z.B. einer anderen IE-Instanz)
 Der Popupblocker des IE bemerkt aber NUR Webseite, die das Popup erzeugt.
 Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popupblocker verwaltet wird.
 Der Popupblocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen Webseiten (die nicht das Popup erzeugt haben).

Der Popupblocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.

Der Popupblockerfehler verändert die Eventverwaltung:

Es werden u.a. ignoriert

onfocus
 onblur
 onfocusin
 onfocusout

und viele andere, so dass trotz Events z.B. des Body der Popupblockerfehler entsteht.

```
// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell
window.focus();
window.document.focus();
if(document.body!=null)
{if(document.body.style!='hidden')          // wenn hidden so focus() nicht möglich (Scriptfehler erzeugt)
 {document.body.focus();}
}
// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt
popupzeiger.show(...);
```

Hinweis: Der Popupfehler ist so elementar, dass die vielen Beta-Testphasen des IE mehr als fragwürdig erscheinen, wie die Angabe von Microsoft, dass Code neu programmiert wurde, um den IE sicherer zu machen.

focus-Methode beim IE 7

windows.focus() document.focus() und body.focus() funktionieren NICHT
 zwischen Register in einem IE-Fenster
 zwischen Fensters z.B. in Taskleiste

Hinweis:

.focus() setzt Element aktiv, gibt dem Element den Focus und feuert dann onfocus
 .setActive() ist Teilmenge von .focus(): nur das aktiv setzen
 funktioniert nicht mit allen Elementen, mit denen .focus() funktioniert

Abänderungen wegen Streitigkeiten von Microsoft mit Fremdanbietern

Ein sehr bekanntes Beispiel ist die nachträglich eingeführte Einschränkung von Active-X-Controls wegen Patentwahrung durch Microsoft, wobei für den JScript-Programmierer massive Änderungen eintreten.

Wegen Patentwahrung hat Microsoft ein zunächst freiwilliges Patch herausgegeben, dass bei ActiveX-Control per APPLET, EMBED oder OBJECT, die auf dem Bildschirm rendern (mit oder ohne Userschnittstelle), dafür sorgt, dass bei mouseover über das Control eine Sprechblase erscheint, die darauf hinweist, dass das Objekt als ActiveX-Control klickbar ist.
 Diese Sprechblase erscheint auch, wenn das Control keine Userschnittstelle hat, also diese gar nicht klickbar ist.

Es wurde das Eventmodell gleichzeitig geändert:

Es werden alle Events solange unterdrückt, bis der User die Sprechblase geklickt hat.
 Das Klicken muss auf das Objekt im Sprechblasenrahmen erfolgen, der so groß ist, wie die Dimension, in der gerendert wurde.
 Es muss also ERST per Mausklick das Control aktiviert werden, ehe das Control klickbar und damit die Eventsteuerung aktiviert ist.
 Ein Control, dass programmtechnisch zwar was rendert, aber ansonsten ohne sichtbare programmtechnisch startet, muss ebenfalls geklickt werden, obwohl es bereits läuft und es nichts zu klicken gäbe (wenn keine Eventsteuerung eingebaut wurde).
 Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.
 Die Eventauslösung per nicht-objekteigenen Eventhandler, der für das Objekt per fireEvent() ein Event auslöst, ist solange blockiert, bis der User die Sprechblase geklickt hat.

style.visibility='hidden' wird ignoriert

Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert: Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.



Diese Sprechblase erscheint nur DANN NICHT, wenn die Userschnittstelle mit Breite == Höhe == 0 gerendert wird. Sollte die Userschnittstelle in einem Container liegen, z.B. DIV, dann wird der Container, wenn er in der Dimension kleiner ist, also die Userschnittstelle, angepasst. Daher muss der Container ebenfalls mit Breite == Höhe == 0 gerendert werden. Wegen Dimensionierung auf 0 sollte style.visibility="hidden" sein. Im Falle eines Containers reicht es, den style des Containers zu ändern, da visibility normalerweise vererbt wird an Kinder, also auch an das Control.

Abänderung wegen Abschaltungen

DirectX ist wegen Abschaltung von Active-X--Controls nicht mehr abwärtskompatibel:

Z.B. wurde bei Win XP SP2 Direct Animation aus DirectX schlagartig durch Abschaltung von Bibliotheken dezimiert, die es bei Win XP SP1 aber noch gibt.

Hier ein Beispiel aus dem Jahr 2004: Abschaltungen von Active-X-Controls

-Controls und Unterstützung/Verbot 20041215

erlaubt sind noch

Tabular Data-Steuerelement {333C7BC4-460F-11D0-BC04-0080C7055A83} Das TDC (Tabular Data-Steuerelement) ermöglicht die Weiterverarbeitung von Daten, die nur im Textformat vorliegen, beispielsweise durch Darstellung in einer Tabelle oder Sortierung. Weitere Informationen:•

http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp(http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp)

Microsoft Agent Control - Version 2.0 {D45FD31B-5C6E-11D1-9EC1-00C04FD7081F} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Weitere Informationen:•

<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>(<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Microsoft MSChat-Steuerelement-Objekt 2.0 - 2.5 {D6526FE0-E651-11CF-99CB-00C04FD64497}

Dieses Steuerelement wird von Webautoren verwendet, um text- und graphisch basierte Chatgemeinden für Echtzeitkonversationen im Web zu erstellen.

Microsoft ActiveX Upload-Steuerelement, Version 1.5 {886e7bf0-c867-11cf-b1ae-00aa00a3f2c3} Dieses Steuerelement kann auf vielerlei Art genutzt werden, um auf einfache Weise Webinhalte via Drag and Drop zu veröffentlichen. Weitere Informationen:• 230298 (<http://support.microsoft.com/kb/230298/DE/>) - Posting Acceptor Release Notes

• http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp
(http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp)

verboten sind

Datenbindung RDS {BD96C556-65A3-11D0-983A-00C04FC29E36} {BD96C556-65A3-11D0-983A-00C04FC29E33} Die RDS (Remote Data Service) Steuerelemente ermöglichen dem Browser, client-basierte SQL Abfragen an einen Webserver zu stellen. Inzwischen wurde RDS jedoch durch neuere Standards wie SOAP abgelöst, von einer weiteren Verwendung von RDS wird daher abgeraten. Weitere Informationen:• 184375 (<http://support.microsoft.com/kb/184375/DE/>) - Sicherheitsaspekte bei RDS 1.5, IIS 3.0 oder 4.0 und ODBC

<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>
(<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>)



http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp
(http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp)

XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer {550dda30-0541-11d2-9ca9-0060b0ec3d39} {CFC399AF-D876-11d0-9C10-00C04FC99C8E} {e54941b2-7756-11d1-bc2a-00c04fb925f3} {7108ECB4-AFDC-11D1-ADC1-00805FC752D8} XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer ermöglichen die Verarbeitung von XML Daten, etwa die Bindung von HTML Elementen an einen XML Datensatz, oder das Einlesen, Manipulieren, und Zurückschreiben von XML Daten.

Die Steuerelemente DOMDocument und XMLIslandPeer bzw. die dazugehörigen ClassIDs sind nicht mehr aktuell, so dass von einer generellen Freigabe dieser Steuerelementgruppe abgeraten wird. Weitere Informationen:•
http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp(http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp)

Internet Explorer

Active Setup / IE Active Setup-Steuerelement {F72A7B0E-0DD8-11D1-BD6E-00AA00B92AF1} Dieses Steuerelement enthält die in Microsoft Security Bulletin MS99-037 beschriebene Sicherheitsanfälligkeit. Um eine weitere Ausführung zu verhindern wurde im Rahmen dieses Security Bulletins ein Kill-Bit gesetzt, so dass selbst bei einer Freigabe dieses Controls eine Ausführung blockiert wird. Weitere Informationen:•

<http://www.microsoft.com/technet/security/bulletin/ms99-037.msp>(<http://www.microsoft.com/technet/security/bulletin/ms99-037.msp>)

<http://www.microsoft.com/technet/security/bulletin/fq99-037.msp>
(<http://www.microsoft.com/technet/security/bulletin/fq99-037.msp>)

240797 (<http://support.microsoft.com/kb/240797/DE/>) - So verhindern Sie die Ausführung von ActiveX-Steuerelementen in Internet Explorer

Media Player / Active Movie Runtime {A4001DE0-7075-11d0-89AB-00A0C9054129} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Runtime Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / ActiveMovie-Steuerelement {05589FA1-C356-11CE-BF01-00AA0055595A} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Microsoft NetShow Player {2179C5D3-EBFF-11CF-B6FD-00AA00B4E220} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das NetShow Player Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Windows Media Player {22D6F312-B0F6-11D0-94AB-0080C74C7E95} Dies ist das Steuerelement für Windows Media Player version 6.4 und war Installationsbestandteil bis einschließlich Windows Media Player Version 8. Ab Windows Media Player 9 wurde diese ClassID durch die neue ClassID {6BF52A52-394A-11D3-B153-00C04F79FAA6} abgelöst, deren Verwendung stattdessen empfohlen wird. Ab Windows Media Player Version 9 wird ferner die alte ClassID anhand eines Wrappers automatisch auf die neue ClassID umgeleitet. Die ClassID für Windows Media Player Version 9 ist jedoch nicht in der Liste der vom Administrator genehmigten Steuerelemente enthalten, und muss bei Bedarf manuell hinzugefügt werden.

Animierte Schaltflächen {0482B100-739C-11CF-A3A9-00A0C9034920} Dieses Steuerelement erlaubte in frühen Versionen des Internet Explorer die Verwendung animierter Schaltflächen auf Webseiten. Das Steuerelement wird



nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von der Freigabe des Steuerelements wird daher abgeraten.

IE Label-Steuerelement

{99B42120-6EC7-11CF-A6C7-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 auch kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

IE Menu-Steuerelement {74701400-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ermöglicht die Handhabung von Menüstrukturen in Webseiten, wird jedoch nicht mehr unterstützt und dürfte nur noch selten Verwendung finden. Von einer Freigabe des Steuerelements wird daher abgeraten.

IE Preloader-Steuerelement {16E349E0-702C-11CF-A3A9-00A0C9034920} Dieses Steuerelement ermöglichte das Vorladen von Webseiten, ist jedoch inzwischen nicht mehr aktuell, wird nicht mehr unterstützt und dürfte nicht mehr im Einsatz sein. Aufgrund einer potentiellen Sicherheitsanfälligkeit in diesem Steuerelement wird von einer Freigabe abgeraten. Weitere Informationen: • 231452 (<http://support.microsoft.com/kb/231452/DE/>) - Update Available for "Legacy ActiveX Control" Issue

IE Timer-Steuerelement {59CCB4A0-727D-11CF-AC36-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

MCSiMenü {275E2FE0-7486-11D0-89D6-00A0C90C9B67} Dieses Steuerelement dient der Anpassung von Pop-upmenüs, ist jedoch nicht mehr aktuell und wurde nach Windows 98 nicht mehr ausgeliefert. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten.

Pop-upmenüobjekt {7823A620-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

Microsoft Agent Control - Version 1.5 {F5BE8BD2-7DE6-11D0-91FE-00C04FD701A5} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Diese Version des Steuerelements ist jedoch nicht mehr aktuell und wird nicht mehr unterstützt. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • <http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm> (<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)



Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden. Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.
Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein, innerhalb dessen dann die neuen HTML-Elemente erzeugt werden.

1.1.7. Browsererkennung zur Unterscheidung browserinterner Objekte

Zur Verwendung der browserinternen Objekte muss die Unterscheidung der Browser erfolgen.

Z.B. kann die **pauschale** Differenzierung von Netscape und Microsoft mit folgender Abfrage erfolgen:

```
//      Prüfung auf Browsertyp
//      Dieser Quellcode muss VOR allen anderen Routinen kodiert sein, damit zuerst abgearbeitet
var ns = document.layers ? true : false;
var ie = document.all ? true : false;
if (ns)
{ // Code für Netscape }
else
{
    if (ie)
    { // Code für Internet Explorer }
}
```

Hinweis: Diese Darstellung lässt andere Browser außen vor, es sei denn, sie sind
objekt-kompatibel zu o.g. Browsern.

Im Falle von Netscape 6.x ist aber inzwischen eine Änderung gegenüber dessen Vorgänger 4.7x eingetreten. Es wurde mit Fähigkeiten des verbreiteten Netscape 4.7x konsequent gebrochen, was so mancher Internet-User nicht weiss, der zu dem eventuell noch hartnäckig auf "Netscape schwört" (Letzteres ist ein kontraproduktives Verhalten, dass im Kampf der Browserhersteller zum Zweck der Erzielung monopolistischer Internet-Marktanteile und Profite dem User regelrecht suggeriert wurde, wobei Microsoft langwieriger aber cleverer vorging: Marktanteile auf Basis der Browser-Integration in das Betriebssystem Windows und dessen Anwendungen (auch im Internet) **und** das konsequente aber z.Z. nicht standardisierte Design im Internet Explorer, das für die Integration in das objektorientierte Betriebssystem Windows wichtig ist, aber auch dem JavaScript-Programmierer beim Web-Design zum Vorteil gereicht, solange Windows benutzt wird (Linux wird hier nicht betrachtet da z.T. OpenSource). Nicht zu verachten ist auch der Hardware-Ressourcen-Bedarf: Während der IE mit Hochfahren von Windows z.T. instanziiert wird, muss Netscape eigene Ressourcen allokalieren, was nachteilig ist, wenn IE und Netscape auf dem Rechner aktiviert werden.)

Der Netscape 6.x ist dem Internet Explorer näher als nie zuvor. Netscape realisiert damit endlich Möglichkeiten des Web-Desing per JavaScript, die auch der Internet Explorer schon länger bedient.

Knackpunkt ist also die Browserprüfung auf unterstützte browereigene Objekte. Es könnte möglich sein, dass der Netscape auch ein document.all abbilden kann. Netscape 6.x hat definitiv keine <LAYER>- und <ILAYER> Unterstützung mehr, vermutlich auch keine Objekt-Unterstützung von layer. Die Verwendung des <DIV>-Tag wurde auch geändert, obwohl dieser Tag gerade für DHTML wichtig ist.

Aufgrund der unzähligen Browserversionen und -änderungen ist es nötig, die Objektfähigkeit des Browsers zu prüfen, wenn ein Objekt benutzt werden soll. Natürlich muss der Benutzer im Browser JavaScript eingeschaltet haben. Es wird tendenziell sichtbar, dass User ohne JavaScript-Zuschaltung mit dem rein HTML-basierenden Seiten ins Abseits gerät. Der Preis für JavaScript: Es kann missbräuchlich auf Basis der Unwissenheit des Internet-Users (der eigentlich "nur surfen will") benutzt werden. Der User muss sich entscheiden, für ihm vertraute Seiten JavaScript zuzuschalten, oder auf Desing zu verzichten.

Beispiel:

```
var ns = document.layers ? true : false;
var ie = document.all ? true : false;

var ie55=false;
if (ie)
{
    // auf IE-Version 5.5 prüfen
    ie55 = (navigator.appVersion.indexOf('5.5') != -1);
}

// Diese Abfrage hat einen Haken: Sie ist abhängig wie der String in ie55 aufgebaut ist.
// Und dieser Aufbau ist Sache des Browserherstellers, der nach Belieben vorgehen kann.
```



```

if (window.createPopup())    // Abfrage, die den Browserhersteller nicht verwendet
.....                      // Nachteil: Jedes zu benutzende Objekt und dessen gewünschte
                             // Eigenschaften muss geprüft werden
                             // window.createPopup ist ein browser-internes Objekt, das vom IE 5.5
                             // instanziiert wird, aber Hauptsache das Objekt IST instanziiert,
                             // egal von welchem Browser.
)
.....

```

Microsoft ändert fortlaufend die Active-X-Eigenschaften von Windows und somit auch des Internet Explorers

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.

Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.

Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Lizenzierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks

Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.

Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.

Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2). Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.

Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

Hinweis: Man suche doch mal im Internet nach einem kostenlosen HTTP-Server vom Microsoft, um IE-Seite testen zu können, die JScript nutzen (inklusive Debugger). Denn sollte kein kostenloses Angebot findbar sein, kommen die Kosten von Entwicklungssoftware zum IE hinzu. Ein Blick auf Konkurrenz-HTTP-Server klärt die Sachlage: Apache-HTTP-Server ist kostenlos, allerdings nicht einfach einzurichten (Hinweis: Der HTTP-Server sollte virtuelle Hosts einrichten können und korrekt mit der Firewall des Users zusammenarbeiten können).

Abänderungen wegen Sicherheitspatches der jeweiligen Windows-Versionen

Abschaltungen von Active-X-Controls erfolgen auch im Rahmen der Sicherheitspatches zu Windows-Versionen.

Es ist auch möglich, dass wegen Sicherheitslücken abgeschaltet wird und somit Komponenten einer Webseite je nach Windowsversion nicht mehr laufen.

Im Rahmen der Sicherheitspatches ist es Microsoft sogar gelungen, Webseiten, die den MS-Encoder zur Komprimierung von

HTML- und JScript-Code nutzen, schlagartig unnutzbar zu machen: Ein Bug in einem Patch zu Windows XP - Q918899

Das Patch verursacht IE-Browser-Absturz bei per MS ScriptEncoder gepacktem JScript unter SP1 und 2 wenn HTTP 1.1 mit Kompression genutzt wird z.B. bei onclick-Handler auf IMG klick ins Fenster per aktivem Popup

Der Absturz ist "read"-Fehler von immer ein und derselben Speicherstelle.

User, die dieses Patch installiert haben, können ab sofort keine IE-Seiten mit codiertem Script mehr ansehen.

Microsoft stellt Abhilfe nach geraumer Zeit zur Verfügung, jedoch spezifisch nach Windows XP-Version:

Patch Q918899 für

Windows XP SP1
SP2

Download für jedermann bereitgestellt

nur auf kostenpflichtige telefonische Anfrage des Users per Downloadlink bereitgestellt, da

Microsoft explizit die User registriert haben will, bei denen das

Patchproblem auftritt (User muss sich Telefonnummer besorgen)

Solange also das Patch zum fehlerhaften Patch vom User nicht installiert wird,

z.B. weil der User keine Ahnung hat, dass und wo er sich die Telefonnummer

von Microsoft besorgen muss bzw. zu besorgen hat, wird der User

IE-Seiten mit komprimierten Code dauerhaft nicht nutzen können.

(Microsoft-Support ist z.T. nur in Englisch).

Abänderungen wegen Browser-Inkompatibilität

Popupblocker-Fehler

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()

Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.



Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.
Der Fehler tritt nicht auf, wenn ein Fenster per `window.open()` erzeugt wurde.

Bedingung:

- Scriptfehleranzeige ist erlaubt im IE 7
- Popupblocker ist im IE abgeschaltet
- ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 `window.popup` per `.show()` erzeugt.
- ein weiteres Fenster (Register) z.B. leere Seite (`about:blank`)
- beide (Register) liegen in einer gemeinsamen IE-Instanz

Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per `.show()` erzeugt, bricht der Browser das Dokument mit `.show()` ab (Scriptfehler).

Der Popupblocker für die leere Seite verursacht den Programmfehler im Dokument mit `.show()`. Es wird folgende Meldung angezeigt (in der Informationsleiste):

'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'

Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:

Der Popupblocker hat ein Populfenster geblockt. Sie können den Popupblocker deaktivieren oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.

Die Realität zur obigen Meldung ist völlig anders:

Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter

Popupblocker einschalten

weitere Informationen

jedoch keine Möglichkeit wie laut Bedeutung

Damit gilt: Der abgeschaltete Popupblocker ist in Wirklichkeit aktiv.

Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster einer Windowsanwendung z.B. einer anderen IE-Instanz)

Der Popupblocker des IE bemeckert aber NUR Webseite, die das Popup erzeugt.

Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popupblocker verwaltet wird.

Der Popupblocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen Webseiten (die nicht das Popup erzeugt haben).

Der Popupblocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.

Der Popupblockerfehler verändert die Eventverwaltung:

Es werden u.a. ignoriert

`onfocus`

`onblur`

`onfocusin`

`onfocusout`

und viele andere, so dass trotz Events z.B. des Body der Popupblockerfehler entsteht.

// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell

`window.focus();`

`window.document.focus();`

`if(document.body!=null)`

`{if(document.body.style!='hidden') // wenn hidden so focus() nicht möglich (Scriptfehler erzeugt)`

`{document.body.focus();}`

`}`

// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt

`popupzeiger.show(...);`

Hinweis: Der Popupfehler ist so elementar, dass die vielen Beta-Testphasen des IE mehr als fragwürdig erscheinen, wie die Angabe von Microsoft, dass Code neu programmiert wurde, um den IE sicherer zu machen.

focus-Methode beim IE 7

`windows.focus()` `document.focus()` und `body.focus()` funktionieren NICHT
zwischen Register in einem IE-Fenster
zwischen Fensters z.B. in Taskleiste

Hinweis:

`.focus()` setzt Element aktiv, gibt dem Element den Focus und feuert dann `onfocus`

`.setActive()` ist Teilmenge von `.focus()`: nur das aktiv setzen

funktioniert nicht mit allen Elementen, mit denen `.focus()` funktioniert

Abänderungen wegen Rechtstreitigkeiten von Microsoft mit Fremdanbietern

Ein sehr bekanntes Beispiel ist die nachträglich eingeführte Einschränkung von Active-X-Controls wegen Patentwahrung durch Microsoft, wobei für den JScript-Programmierer massive Änderungen eintreten.

Wegen Patentwahrung hat Microsoft ein zunächst freiwilliges Patch herausgegeben, dass bei ActiveX-Control per `APPLET`, `EMBED` oder `OBJECT`, die auf dem Bildschirm rendern (mit oder ohne Userschnittstelle), dafür sorgt, dass bei `mouseover` über das Control



eine Sprechblase erscheint, die darauf hinweist, dass das Objekt als ActiveX-Control klickbar ist.
Diese Sprechblase erscheint auch, wenn das Control keine Userchnittstelle hat, also diese gar nicht klickbar ist.

Es wurde das Eventmodell gleichzeitig geändert:

Es werden alle Events solange unterdrückt, bis der User die Sprechblase geklickt hat.
Das Klicken muss auf das Objekt im Sprechblasenrahmen erfolgen, der so groß ist, wie die Dimension, in der gerendert wurde.
Es muss also ERST per Mausklick das Control aktiviert werden, ehe das Control klickbar und damit die Eventsteuerung aktiviert ist.
Ein Control, dass programmtechnisch zwar was rendert, aber ansonsten ohne sichtbare programmtechnisch startet, muss ebenfalls geklickt werden, obwohl es bereits läuft und es nichts zu klicken gäbe (wenn keine Eventsteuerung eingebaut wurde).
Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.
Die Eventauslösung per nicht-objekteigenen Eventhandler, der für das Objekt per fireEvent() ein Event auslöst, ist solange blockiert, bis der User die Sprechblase geklickt hat.

style.visibility='hidden' wird ignoriert

Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert: Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.

Diese Sprechblase erscheint nur DANN NICHT, wenn die Userchnittstelle mit Breite == Höhe == 0 gerendert wird. Sollte die Userchnittstelle in einem Container liegen, z.B. DIV, dann wird der Container, wenn er in der Dimension kleiner ist, also die Userchnittstelle, angepasst. Daher muss der Container ebenfalls mit Breite == Höhe == 0 gerendert werden. Wegen Dimensionierung auf 0 sollte style.visibility="hidden" sein. Im Falle eines Containers reicht es, den style des Containers zu ändern, da visibility normalerweise vererbt wird an Kinder, also auch an das Control.

Abänderung wegen Abschaltungen

DirectX ist wegen Abschaltung von Active-X--Controls nicht mehr abwärtskompatibel:

Z.B. wurde bei Win XP SP2 Direct Animation aus DirectX schlagartig durch Abschaltung von Bibliotheken dezimiert, die es bei Win XP SP1 aber noch gibt.

Hier ein Beispiel aus dem Jahr 2004: Abschaltungen von Active-X-Controls

ActiveX-Controls und Unterstützung/Verbot 20041215

erlaubt sind noch

Tabular Data-Steuerelement{333C7BC4-460F-11D0-BC04-0080C7055A83} Das TDC (Tabular Data-Steuerelement) ermöglicht die Weiterverarbeitung von Daten, die nur im Textformatvorliegen, beispielsweise durch Darstellung in einer Tabelle oder Sortierung. Weitere Informationen:•

http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp(http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp)

Microsoft Agent Control - Version 2.0 {D45FD31B-5C6E-11D1-9EC1-00C04FD7081F} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Weitere Informationen:•

<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>(<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Microsoft MSChat-Steuerelement-Objekt 2.0 - 2.5 {D6526FE0-E651-11CF-99CB-00C04FD64497}

Dieses Steuerelement wird von Webautoren verwendet, um text- und graphisch basierte Chatgemeinden für Echtzeitkonversationen im Web zu erstellen.

Microsoft ActiveX Upload-Steuerelement, Version 1.5 {886e7bf0-c867-11cf-b1ae-00aa00a3f2c3} Dieses Steuerelement kann auf vielerlei Art genutzt werden, um auf einfache Weise Webinhalte via Drag and Drop



zuveröffentlichen. Weitere Informationen:• 230298 (<http://support.microsoft.com/kb/230298/DE/>) - Posting Acceptor Release Notes

• http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp
(http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp)

verboten sind

Datenbindung RDS {BD96C556-65A3-11D0-983A-00C04FC29E36} {BD96C556-65A3-11D0-983A-00C04FC29E33} Die RDS (Remote Data Service) Steuerelemente ermöglichen dem Browser, client-basierte SQL Abfragen an einen Webserver zu stellen. Inzwischen wurde RDS jedoch durch neuere Standards wie SOAP abgelöst, von einer weiteren Verwendung von RDS wird daher abgeraten. Weitere Informationen:• 184375 (<http://support.microsoft.com/kb/184375/DE/>) - Sicherheitsaspekte bei RDS 1.5, IIS 3.0 oder 4.0 und ODBC

<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>
(<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>)

http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp
(http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp)

XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer {550dda30-0541-11d2-9ca9-0060b0ec3d39} {CFC399AF-D876-11d0-9C10-00C04FC99C8E} {e54941b2-7756-11d1-bc2a-00c04fb925f3} {7108ECB4-AFDC-11D1-ADC1-00805FC752D8} XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer ermöglichen die Verarbeitung von XML Daten, etwa die Bindung von HTML Elementen an einen XML Datensatz, oder das Einlesen, Manipulieren, und Zurückschreiben von XML Daten.

Die Steuerelemente DOMDocument und XMLIslandPeer bzw. die dazugehörigen ClassIDs sind nicht mehr aktuell, so dass von einer generellen Freigabe dieser Steuerelementgruppe abgeraten wird. Weitere Informationen:• http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp(http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp)

Internet Explorer

Active Setup / IE Active Setup-Steuerelement {F72A7B0E-0DD8-11D1-BD6E-00AA00B92AF1} Dieses Steuerelement enthält die in Microsoft Security Bulletin MS99-037 beschriebene Sicherheitsanfälligkeit. Um eine weitere Ausführung zu verhindern wurde im Rahmen dieses Security Bulletins ein Kill-Bit gesetzt, so dass selbst bei einer Freigabe dieses Controls eine Ausführung blockiert wird. Weitere Informationen:•

<http://www.microsoft.com/technet/security/bulletin/ms99-037.msp>
(<http://www.microsoft.com/technet/security/bulletin/ms99-037.msp>)

<http://www.microsoft.com/technet/security/bulletin/fq99-037.msp>
(<http://www.microsoft.com/technet/security/bulletin/fq99-037.msp>)

240797 (<http://support.microsoft.com/kb/240797/DE/>) - So verhindern Sie die Ausführung von ActiveX-Steuerelementen in Internet Explorer

Media Player / Active Movie Runtime {A4001DE0-7075-11d0-89AB-00A0C9054129} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Runtime Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / ActiveMovie-Steuerelement {05589FA1-C356-11CE-BF01-00AA0055595A} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.



Media Player / Microsoft NetShow Player {2179C5D3-EBFF-11CF-B6FD-00AA00B4E220} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das NetShow Player Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Windows Media Player {22D6F312-B0F6-11D0-94AB-0080C74C7E95} Dies ist das Steuerelement für Windows Media Player version 6.4 und war Installationsbestandteil bis einschließlich Windows Media Player Version 8. Ab Windows Media Player 9 wurde diese ClassID durch die neue ClassID {6BF52A52-394A-11D3-B153-00C04F79FAA6} abgelöst, deren Verwendung stattdessen empfohlen wird. Ab Windows Media Player Version 9 wird ferner die alte ClassID anhand eines Wrappers automatisch auf die neue ClassID umgeleitet. Die ClassID für Windows Media Player Version 9 ist jedoch nicht in der Liste der vom Administrator genehmigten Steuerelemente enthalten, und muss bei Bedarf manuell hinzugefügt werden.

Animierte Schaltflächen {0482B100-739C-11CF-A3A9-00A0C9034920} Dieses Steuerelement erlaubte in frühen Versionen des Internet Explorer die Verwendung animierter Schaltflächen auf Webseiten. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von der Freigabe des Steuerelements wird daher abgeraten.

IE Label-Steuerelement

{99B42120-6EC7-11CF-A6C7-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 auch kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

IE Menu-Steuerelement {74701400-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ermöglicht die Handhabung von Menüstrukturen in Webseiten, wird jedoch nicht mehr unterstützt und dürfte nur noch selten Verwendung finden. Von einer Freigabe des Steuerelements wird daher abgeraten.

IE Preloader-Steuerelement {16E349E0-702C-11CF-A3A9-00A0C9034920} Dieses Steuerelement ermöglichte das Vorladen von Webseiten, ist jedoch inzwischen nicht mehr aktuell, wird nicht mehr unterstützt und dürfte nicht mehr im Einsatz sein. Aufgrund einer potentiellen Sicherheitsanfälligkeit in diesem Steuerelement wird von einer Freigabe abgeraten. Weitere Informationen: • 231452 (<http://support.microsoft.com/kb/231452/DE/>) - Update Available for "Legacy ActiveX Control" Issue

IE Timer-Steuerelement {59CCB4A0-727D-11CF-AC36-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

MCSiMenü {275E2FE0-7486-11D0-89D6-00A0C90C9B67} Dieses Steuerelement dient der Anpassung von Pop-upmenüs, ist jedoch nicht mehr aktuell und wurde nach Windows 98 nicht mehr ausgeliefert. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten.



Popupmenüobjekt {7823A620-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

Microsoft Agent Control - Version 1.5 {F5BE8BD2-7DE6-11D0-91FE-00C04FD701A5} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Diese Version des Steuerelements ist jedoch nicht mehr aktuell und wird nicht mehr unterstützt. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • <http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm> (<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden. Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.
Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein, innerhalb dessen dann die neuen HTML-Elemente erzeugt werden.

1.2. Variablen

1.2.1. Variablen und ihre Gültigkeit

1.2.1.1. Globale Variable

1.2.1.1.1. Globale Variable Standard-Definition

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--      // x global definieren
                var x = 17;

                function ausgabe()
                {alert("Der Wert von x ist " + x); }

      // -->
    </SCRIPT>
  </HEAD>
</HTML>
```

1.2.1.1.2. Globale Variable durch eine Funktion definieren

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
                function definieren()
                {x = 17; // lokal zu dieser Funktion definiert!}

                function anzeigen()
                {
                  definieren(); // x wird nun global, also auch in anzeigen() verfügbar
                  alert("Der Wert von x ist " + x);}

                anzeigen(); // kein Fehler, da x global ist

      // -->
    </SCRIPT>
  </HEAD>
</HTML>
```



1.2.1.2. Lokale Variable

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function definieren()
        {var x = 17; // lokal zu dieser Funktion}

        function anzeigen()
        {alert("Der Wert von x ist " + x);}          // x ist nicht definiert !

        anzeigen(); // Fehler: x ist nicht global also in anzeigen() nicht verfügbar !
      // -->
    </SCRIPT>
  </HEAD>
</HTML>

```

1.2.2. Array-Variablen**1.2.2.1. Array anlegen und initialisieren****1.2.2.1.1. eindimensionales Array****Variante 1**

```

<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var meinArray = new Array(3); // Anzahl immer ab 1
      meinArray[0] = "Tom";        // Index immer ab 0
      meinArray[1] = "Jerry";
      meinArray[2] = "Tom und Jerry";

      document.write(meinArray[0] + "<BR>");
      document.write(meinArray[1] + "<BR>");
      document.write(meinArray[2]);
    // -->
  </SCRIPT>
</HTML>

```

Variante 2

```

<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var meinArray = new Array("Tom", "Jerry", "Tom und Jerry");
      document.write(meinArray[0] + "<BR>"); // Index immer ab 0
      document.write(meinArray[1] + "<BR>");
      document.write(meinArray[2]);        // Länge immer ab 1
    // -->
  </SCRIPT>
</HTML>

```

1.2.2.1.2. mehrdimensionales Array

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function mehr_dim_array_erzeugen( ein_dim_array_anzahl,
                                           ein_dim_array_elemente_anzahl
                                           )
        {
          // ist Konstruktor für Erzeugung eines mehrdimensionalen Arrays
          // als Menge von eindimensionalen Arrays

          for ( var ein_dim_array_zahler = 0;
                ein_dim_array_zahler < ein_dim_array_anzahl;
                ein_dim_array_zahler++
              )
          {

```



```

        this[ein_dim_array_zahler] = new Array(ein_dim_array_elemente_anzahl);
        // ein_dim_array_zahler immer ab 1
        // je ein eindimensionales Array erzeugen und zu this zuordnen
        // this ist Zeiger des Konstruktors auf das per new
        // anzulegende mehrdimensionale Feld
    }

    this.breite = ein_dim_array_anzahl;
    this.hoehe = ein_dim_array_elemente_anzahl;
}
// -->
</SCRIPT>
</HEAD>

<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--
    zwei_dim_array = new mehr_dim_array_erzeugen (2, 3);
    // zwei Arrays zu je 3 Feldelementen,
    // zweidimensionale Matrix mit
    // 2 Zeilen zu je 3 Spalten
    // erzeugen
    // Anzahl immer ab 1

    zwei_dim_array[1][2] = 17;
    // letztes eindimensionale Feld in dessen
    // Element 2 (vorletztes) mit Wert 17 initialisieren
    // Indexe immer ab 0
    document.write("Feld 1-2: " + zwei_dim_array[1][2]);

// -->
</SCRIPT>
</BODY>
</HTML>

```

1.2.2.2. Größe eines Arrays nach dem Anlegen ändern durch Hinzufügung von neuen Elementen

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var meinArray = new Array(1); // Anzahl immer ab 1, also 1 Element
    // Datentyp von Elementen steht erst
    // nach Initialisierung fest

    document.write("Das Feld meinArray hat ");
    document.write(meinArray.length + " Elemente.<BR>");

    meinArray[1] = "zeichenkette"; // Index immer ab 0
    // Element 2 initialisieren und damit
    // 2 Feldelemente erzeugen (0 bis 1)
    // wobei
    // NUR das zweite Element mit
    // dem Index 1 initialisiert wird
    // NUR auf das zweite Element
    // zugegriffen werden darf,
    // da nur dieses einen Datentyp hat

    document.write("Das Feld meinArray hat jetzt ");
    document.write(meinArray.length + " Elemente.");

// -->
</SCRIPT>
</HTML>

```

1.2.3. Operationen mit Variablen

1.2.3.1. numerische Variable

1.2.3.1.1. berechnen

Beispiel 1: numerische Operationen



```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    document.write(17 + 5 * 3); // ergibt 32, da * höhere Priorität als + hat
//-->
</SCRIPT>
</HTML>
```

Beispiel 2: *rechnet Zoll-Angabe in Zentimeter um*

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var zentimeter, zoll, umrechnungsfaktor;
    zoll = 1;
    umrechnungsfaktor = 2.54;
    zentimeter = umrechnungsfaktor * zoll;

    document.write(zoll + " Zoll sind " + zentimeter + " Zentimeter");
    // 1 Zoll sind 2.54 Zentimeter
// -->
</SCRIPT>
</HTML>
```

1.2.3.1.2. prüfen

Beispiel: *prüfen auf nicht-numerisch (Not an Number)*

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    zeichenkette = "xyz";

    pruefergebnis= parseInt(zeichenkette);

    document.write(isNaN(pruefergebnis)); // true
// -->
</SCRIPT>
</HTML>
```

1.2.3.2. Zeichenketten-Variable

1.2.3.2.1. Zeicheneingabe prüfen ohne reguläre Ausdrücke (ohne reg-Methode)

Beispiel: *Zeichenkette auf Wertebereich der Zeichen prüfen*

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function pruefe_zeichenkette(zeichenkette, wertebereich)
    {
        //      wertebereich ist Zeichenkette z.B. "0123456789 -+/,()"

        var rueckgabewert = true; // Datentyp Boolean
                                // Annahme: Zeichenkette hat NUR Zeichen
                                // aus dem Wertebereich

        var zeichen_aus_zeichenkette;

        //      Zeichenkette zeichenweise analysieren:
        //      jedes Zeichen mit Wertebereich vergleichen
        for (var i = 0; i < zeichenkette.length; i++)
        {
            zeichen_aus_zeichenkette = zeichenkette.charAt(i);
            if (wertebereich.indexOf(zeichen_aus_zeichenkette) == -1)
            {
                rueckgabewert = false; break;
            }
            // ungültiges Zeichen gefunden, also abbrechen
        }

        return rueckgabewert;
    }
    function pruefe_eingabe(zu_pruefende_zeichenkette)
    {
        if ( pruefe_zeichenkette(zu_pruefender_zeichenkette, "0123456789 -+/,()") )
```



```

        {alert("Eingabe ist korrekt !");}
        else
        {alert("Eingabe ist nicht korrekt !"); }
    }
    //-->
</SCRIPT>
</HEAD>
<BODY>
    <FORM>
        Telefon:
        <INPUT  TYPE="text"
                NAME="Telefon"
                VALUE="">
        <INPUT  TYPE="button"
                VALUE="Ueberpruefen"
                onClick="pruefe_eingabe(this.form.Telefon.value)">
    </FORM>
</BODY>
</HTML>

```

1.2.3.2.2. Teilstring suchen ohne reguläre Ausdrücke (ohne reg-Methode)

Beispiel: E-Mail-Adresse auf "@" prüfen

```

<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JavaScript">
            <!--
                function pruefe_zeichenkette(zeichenkette)
                {
                    if (zeichenkette.indexOf('@') = -1)
                    {alert("@ fehlt !");}
                    else
                    {alert("@ gefunden !");}
                }
            //-->
        </SCRIPT>
    </HEAD>
    <BODY>
        <FORM>
            e-Mail:
            <INPUT  TYPE="text"
                    NAME="Mail"
                    VALUE="">
            <INPUT  TYPE="button"
                    VALUE="Ueberpruefen"
                    onClick="pruefe_zeichenkette (this.form.Mail.value)">
        </FORM>
    </BODY>
</HTML>

```

1.2.3.2.3. Zeichenkette zu numerisch konvertieren

```

<HTML>
    <SCRIPT LANGUAGE="JavaScript">
        <!--
            zeichenkette= "17.50 DM";
            num_wert = parseFloat(zeichenkette);
            document.write(num_wert);
        // -->
    </SCRIPT>
</HTML>

```

1.2.3.2.4. Zeichenketten verknüpfen

Variante 1

```

<HTML>
    <SCRIPT LANGUAGE="JavaScript">
        <!--
            var name = "Du";
            var begruessung;

```



```

        begruessung = "Hallo " + name + " !";
        document.write(begruessung);
    // -->
</SCRIPT>
</HTML>

```

Variante 2

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var name = "Du";
    document.write("Hallo " + name + " !");
    // -->
</SCRIPT>
</HTML>

```

1.2.3.2.5. Dezimalwert zu Hexaziffern-Zeichenkette konvertieren

```

<SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
<!--
    var hexaziffern_feld = [
        "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
        "A", "B", "C", "D", "E", "F"
    ];

    function dezimalwert_zu_hexaziffern(dezimal_wert)
    {
        var high = Math.floor(dezimal_wert / 16);
        var low = Math.floor(dezimal_wert - (high * 16));

        return(hexaziffern_feld[high] + hexaziffern_feld[low]);
    }
    //-->
</SCRIPT>

```

1.2.3.2.6. Hexaziffern von bzw. zu numerisch

Hexaziffer zu numerisch:

```

var X00=0;
var X01='09AF';
X01=X01.toUpperCase();
X00=X01.charCodeAt(0);X00-=48;if(X00>=17){X00-=7;}alert(X00);
X00=X01.charCodeAt(1);X00-=48;if(X00>=17){X00-=7;}alert(X00);
X00=X01.charCodeAt(2);X00-=48;if(X00>=17){X00-=7;}alert(X00);
X00=X01.charCodeAt(3);X00-=48;if(X00>=17){X00-=7;}alert(X00);

```

// nur '0' bis '9' und 'A' bis 'F' grossbuchstaben
// nach Grossbuchstaben
// 0
// 9
// 10
// 15

numerisch 0 bis 15 zu Hexaziffer '0' bis 'F'

```

var X00=0;X01=0;
var X02="";
X00=0; X01=X00;X01+=7;if(X01>=17){X00=X01;}X00+=48;X02=String.fromCharCode(X00);alert(X02); // '0'
X00=9; X01=X00;X01+=7;if(X01>=17){X00=X01;}X00+=48;X02=String.fromCharCode(X00);alert(X02); // '9'
X00=10;X01=X00;X01+=7;if(X01>=17){X00=X01;}X00+=48;X02=String.fromCharCode(X00);alert(X02); // 'A'
X00=15;X01=X00;X01+=7;if(X01>=17){X00=X01;}X00+=48;X02=String.fromCharCode(X00);alert(X02); // 'F'

```

numerisch 0 bis 255 zu Hexaziffern paar

```

var X00=0;X01=0;X02=0;X03=0;X04="";
X00=51;
X01=X00/16;X02=Math.floor(X01);
X01=X02*16;X00-=X01;
X03=X02;X03+=7;if(X03>=17){X02=X03;}X02+=48;X04=String.fromCharCode(X02);
X03=X00;X03+=7;if(X03>=17){X00=X03;}X00+=48;X04+=String.fromCharCode(X00);
alert(X04);

```

// ganzzahlige Division
// X02 num. Wert der 1. Hexaziffer
// X00 num. Wert der 2. Hexaziffer
// 1. Hexaziffer
// 2. Hexaziffer
// '33'

Hexaziffern paar zu numerisch 0 bis 255

```

var X00=0;X01=0;
var X02='33';
X02=X02.toUpperCase();
X00=X02.charCodeAt(0);X00-=48;if(X00>=17){X00-=7;}

```

// nur '0' bis '9' und 'A' bis 'F' grossbuchstaben
// nach Grossbuchstaben
// 1. Ziffer zu numerisch 3



```
X01=X02.charCodeAt(1);X01-=48;if(X01>=17){X01-=7;} // 2. Ziffer zu numerisch 3
X00*=16;X00+=X01;
alert(X00); // 51 numerisch
```

1.2.3.3. Array-Variable

1.2.3.3.1. Verketteten von Arrayelementen des Zeichenkettentyps mit anderen Zeichenketten

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var meinArray = new Array();
    meinArray["Tom"] = "Auto";
    meinArray["Jerry"] = "Motorrad";
    meinArray["Tom und Jerry"] = "Roller";

    document.write("Tom: " + meinArray["Tom"] + "<BR>");
    document.write("Jerry: " + meinArray["Jerry"] + "<BR>");
    document.write("Tom und Jerry: " + meinArray["Tom und Jerry"] + "<BR>");
// -->
</SCRIPT>
</HTML>
```

1.2.3.3.2. Arrayelemente nach deren Initialisierung erweitern (Methode join())

Beispiel 1

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var meinArray = new Array();

    meinArray[0] = "Tom";
    meinArray[1] = "Jerry";
    meinArray[2] = "Tom und Jerry";

    erweiterung = meinArray.join("/");
    // meinArray.join() entspricht meinArray.join(",")

    document.write(erweiterung ); // Tom/Jerry/Tom und Jerry
// -->
</SCRIPT>
</HTML>
```

Beispiel 2

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var meinArray = [17, 100.5, "Jerry"];
    document.write(meinArray.join()); // 17,100.5,Jerry
    // meinArray.join() entspricht meinArray.join(",")
// -->
</SCRIPT>
</HTML>
```

1.2.3.3.3. Reihenfolge von Array-Elementen ändern

1.2.3.3.3.1. umkehren (Methode reverse())

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var meinArray = new Array(1,2,3);
    document.write(meinArray.join() + "<BR>"); // 1,2,3
    // meinArray.join() entspricht meinArray.join(",")

    meinArray.reverse();
    document.write(meinArray.join()); // 3,2,1
    // meinArray.join() entspricht meinArray.join(",")
// -->
```



```
</SCRIPT>
</HTML>
```

1.2.3.3.3.2. Sortieren (Methode sort())

1.2.3.3.3.2.1. nach dem Alphabet sortieren (Standardsortierung ohne Sortierfunktion)

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var meinArray = new Array();
    meinArray[0] = "Tom";
    meinArray[1] = "Jerry";
    meinArray[2] = "Tom und Jerry";

    document.write(meinArray.join() + "<BR>");           // Tom,Jerry,Tom und Jerry

    meinArray.sort();
    document.write(meinArray.join());                  // Jerry,Tom,Tom und Jerry
// -->
</SCRIPT>
</HTML>
```

1.2.3.3.3.2.2. nach Zahlenwerten sortieren (Nicht-Standardsortierung mit Sortierfunktion)

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function sortier_funktion(a,b)
    //      muss zwei Parameter haben:
    //      die paarweise zu vergleichenden Größen gleichen Typs
    //      wenn a < b so negative Zahl < 0 liefern
    //      wenn a = b so 0 liefern
    //      wenn a > b so positive Zahl > 0 liefern
    {return a - b;} // Subtraktion

    var meinArray = new Array();

    meinArray[0] = 3;
    meinArray[1] = 17;
    meinArray[2] = 0.004;
    document.write(meinArray.join(", ") + "<BR>");       // 3, 17, 0.004

    //      Standardsortierung also alphabetisch
    meinArray.sort();
    document.write(meinArray.join(", ") + "<BR>");       // 0.004, 17, 3

    //      Nicht-Standardsortierung, also mit Sortierfunktion
    meinArray.sort(sortier_funktion);
    document.write(meinArray.join(", "));              // 0.004, 3, 17
// -->
</SCRIPT>
```



1.3. Verzweigungen und Schleifen

1.3.1. if-else-Verzweigung

Beispiel 1

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var x = 10;
      var y = 17;

      if (x == y)
      { document.write("x ist gleich y<BR>")}
      else
      { document.write("x ist nicht gleich y<BR>");}

      if (x > y) { document.write("x ist groesser als y<BR>");}
      if (x < y) { document.write("y ist groesser als x<BR>");}

    // -->
  </SCRIPT>
</HTML>
```

Beispiel 2

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var boolean_variable = true;

      if (boolean_variable)
      { document.write("Die Variable bool ist true.");}
      else
      { document.write("Die Variable bool ist false.");}

    // -->
  </SCRIPT>
</HTML>
```

Beispiel 3

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var eingabe = "xyz";
      var x = parseInt(eingabe);
      if (isNaN(x))
      { document.write("Die Variable x ist keine Zahl.");}
      else
      { document.write("Die Variable x ist eine Zahl.");}

    // -->
  </SCRIPT>
</HTML>
```

Beispiel 4

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var x = 17;
      var y = 10;

      if (x == y)
      { document.write("x ist gleich y");}
      else
      {
        if (x > y)
        { document.write("x ist groesser als y");}
        else
        { document.write("y ist groesser als x");}
      }

    </SCRIPT>
</HTML>
```



```
// -->
</SCRIPT>
</HTML>
```

1.3.2. switch-Verzweigung

Beispiel 1

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var name = "Tom";

    switch (name)
    {
        case "Tom":      document.write("Hallo Tom !");
                        break;
        case "Jerry":    document.write("Hallo Jerry !");
                        break;
        default:         document.write("Hallo Unbekannter !");
    }
//-->
</SCRIPT>
</HTML>
```

Beispiel 2

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var x = 17;

    switch (x)
    {
        case 2:  document.write("2 erkannt");
                break;
        case 8:  document.write("8 erkannt");
                break;
        default: document.write(x);
    }
//-->
</SCRIPT>
</HTML>
```

1.3.3. for -Schleife

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function anzeigen(zahler)
    {
        document.write(zahler + "<BR>");
    }
// -->
</SCRIPT>
</HEAD>

<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--
    for (var i = 0; i < 10; i++)
    {
        if (i==5)
            {continue;} // bei 5 nichts tun
        else
        {
            if (i == 8)
            {break;}      // vorzeitiger Abbruch bei 8
            else
            {
                document.write("Wiederholung Nr. ");
                anzeigen(i);
            }
        }
    }
// -->
</SCRIPT>
</BODY>
```



</HTML>



1.3.4. while-Schleife

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var fortsetzen = true;
      var x = 0;

      while (fortsetzen)
      {
        x++;
        fortsetzen = confirm("x hat den Wert " + x + " \nSchleife fortsetzen?");
      }
    // -->
  </SCRIPT>
</HTML>
```

1.3.5. do-while-Schleife

Beispiel Berechnung einer Fakultät durch eine rekursive Funktion

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function rekursion(n)
        {
          if (n != 1)
            {return n * rekursion(n - 1);}
          else
            {return 1;}
        }

        function berechnen()
        {
          var zahl;
          zahl = prompt("Startwert: ", "");
          alert("Ergebnis " + rekursion(zahl));
        }
      // -->
    </SCRIPT>
  </HEAD>

  <BODY>
    <FORM>
      <INPUT TYPE="button"
        VALUE="Berechnung starten"
        onClick="berechnen()">
    </FORM>
  </BODY>
</HTML>
```

1.3.6. break-Anweisung für Schleifen

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function anzeigen(zahler)
        {
          document.write(zahler + "<BR>");
        }
      // -->
    </SCRIPT>
  </HEAD>

  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        for (var i = 0; i < 10; i++)
        {
          if (i==5)
            {continue;} // bei 5 nichts tun
          else
            {
              if (i == 8)

```



```

        {break;}           // vorzeitiger Abbruch bei 8
        else
        {
            document.write("Wiederholung Nr. ");
            anzeigen(i);}
        }
    }
// -->
</SCRIPT>
</BODY>
</HTML>

```

1.3.7. continue-Anweisung für Schleifen

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function anzeigen(zahler)
    {
        document.write(zahler + "<BR>");}

// -->
</SCRIPT>
</HEAD>

<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--
    for (var i = 0; i < 10; i++)
    {
        if (i==5)
        {continue;} // bei 5 nichts tun
        else
        {
            if (i == 8)
            {break;}           // vorzeitiger Abbruch bei 8
            else
            {
                document.write("Wiederholung Nr. ");
                anzeigen(i);}
            }
        }
    }
// -->
</SCRIPT>
</BODY>
</HTML>

```

1.4. Funktionen

1.4.1. ohne Konstruktor (ohne Methode new)

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function anzeigen()
    {
        document.write("Text <BR>");}

</SCRIPT>
</HEAD>

<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!--
    // 3 Funktionsaufrufe
    anzeigen();
    anzeigen();
    anzeigen();

// -->
</SCRIPT>

<FORM>
<INPUT TYPE="button"
        VALUE="Anzeigen "

```



```

                                onClick="anzeigen()")>
        </FORM>
    </BODY>
</HTML>

```

1.4.2. mit Konstruktor (mit Methode new)

1.4.2.1. als Objekt anlegen

```

<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      addieren = new Function("x", "y", "return x + y");
      alert(addieren(12, 5));
    // -->
  </SCRIPT>
</HTML>

```

1.4.2.2. als Objekt nachträglich erweitern (Prototyping)

```

<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      function datumsAusgabe()
      {
          tag = this.getDate();
          monat = this.getMonth() + 1;
          jahr = this.getYear() + 1900;

          return tag + "." + monat + "." + jahr;
      }

      // datumsAusgabe() als Prototyp von Date
      Date.prototype.ausgabe = datumsAusgabe;

      heute = new Date();

      document.write("Das heutige Datum in ");
      document.write("deutscher Schreibweise:");

      // Aufruf der neuen Methode
      document.write(heute.ausgabe());
    // -->
  </SCRIPT>
</HTML>

```

Das heutige Datum in deutscher Schreibweise:25.5.1999

1.4.3. Übergabewerte an Funktion (Argumente)

Beispiel 1

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function begruessung(name)
        {
            var text = "Hallo " + name + "!";
            alert(text);
        }
      //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    Namen auswählen:<P>
    <FORM>
      <INPUT TYPE="button"
        VALUE="Tom"
        onClick="begruessung('Tom')">
      <INPUT TYPE="button"
        VALUE="Jerry"
        onClick="begruessung('Jerry')">
    </FORM>
  </BODY>
</HTML>

```



```

</BODY>
</HTML>

```

Beispiel 2

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function addition( x, y)
        {alert(x + y);}
      //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT TYPE="button"
        VALUE="Addieren"
        onClick="addition(12,5)">
    </FORM>
  </BODY>
</HTML>

```

Beispiel 3

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function summe()
        {
          var ergebnis = 0;

          for (var i = 0; i < summe.arguments.length; i++)
          {
            ergebnis += summe.arguments[i];
          }

          alert(ergebnis);
        }
      //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT TYPE="button"
        VALUE="23 + 33 + 53 + 12"
        onclick="summe(23,33,53,12)">
      <INPUT TYPE="button"
        VALUE="Addiere 12 und 5"
        onclick="summe(12,5)">
    </FORM>
  </BODY>
</HTML>

```

1.4.4. Rückgabewert einer Funktionen**1.4.4.1. Rückgabe durch return**

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function multiplizieren(x,y)
        {
          return x*y;
        }
      //-->
    </SCRIPT>
  </HEAD>

```



```

<BODY>
  <FORM>
    <INPUT TYPE="button"
      VALUE="Produkt bilden aus 17 und 3"
      onclick="multiplizieren(17,3))">
    </FORM>
  </BODY>
</HTML>

```

1.4.4.2. Rückgabewert undefined durch void()

```

<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      function test()
      {
        return "xxx";
      }
    //-->
  </SCRIPT>

  <A HREF="javascript:test()">Test 1</A><BR>      // test liefert "xxx"
  <A HREF="javascript:void(test())">Test 2</A>      // test() liefert undefined
</HTML>

```

2. Browser- und System-Informationen (navigator-Objekt)

2.1. Browser-Informationen

```

<SCRIPT LANGUAGE="JavaScript1.2">
<!--
  function UserInfo()
  {
    var info = navigator.userAgent.toLowerCase();
    this.major = parseInt(navigator.appVersion);
    this.minor = parseFloat(navigator.appVersion);
    this.ns = (
      (info.indexOf('mozilla')!=-1)
      && (
        (info.indexOf('spoofer')==-1)
        && (info.indexOf('compatible') == -1)
      )
    );
    this.ns2 = (this.ns && (this.major == 2));
    this.ns3 = (this.ns && (this.major == 3));
    this.ns4b = (this.ns && (this.minor < 4.04));
    this.ns4 = (this.ns && (this.major >= 4));
    this.ie = (info.indexOf("msie") != -1);
    this.ie3 = (this.ie && (this.major == 3));
    this.ie4 = (this.ie && (this.major >= 4));
    this.op3 = (info.indexOf("opera") != -1);
    this.win = (info.indexOf("win") != -1);
    this.mac = (info.indexOf("mac") != -1);
    this.unix = (info.indexOf("x11") != -1);
    this.r640x480 = (screen.width == 640);
    this.r800x600 = (screen.width == 800);
    this.r1024x768 = (screen.width == 1024);
    this.r1280x1024 = (screen.width == 1280);
  }

  var user = new UserInfo();
-->
</SCRIPT>

.....
if(user.ie) //... IE erkannt
if(user.ns) //... Netscape erkannt

```

2.1.1. Name des Browsers

```

<SCRIPT LANGUAGE="JavaScript">
<!--
  (navigator.appName) ? document.writeln(navigator.appName) :
    document.writeln("navigator.appName wird vom Browser nicht unterstuetzt");

```




```
//-->
</SCRIPT>
```

liefert für den IE folgende Zeichenkette "Microsoft Internet Explorer" (ohne "" natürlich)

2.1.1.1. **Browserüberprüfung auf Microsoft Explorer ab 4.x**

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      if (
        &&      (navigator.appVersion.indexOf("MSIE") >0)
                  (parseInt(navigator.appVersion) >= 4)
        )
      { document.write("Microsoft Internet Explorer.");}
      else
      { document.write("Dies ist nicht der Microsoft Internet Explorer.");}
    //-->
  </SCRIPT>
</HTML>
```

Hinweis zu indexOf: "MSIE 3" > 0 so Microsoft Internet Explorer 3
 "MSIE 4" > 0 so Microsoft Internet Explorer 4
 "MSIE 5" > 0 so Microsoft Internet Explorer 5

2.1.1.2. **Browserüberprüfung auf Netscape**

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      if (navigator.appName.indexOf("Netscape") != -1)
      { document.write("Netscape Communicator/Navigator.");}
      else
      { document.write("Dies ist nicht der Netscape Communicator.");}
    </SCRIPT>
  </HTML>
```

Hinweis: browser_name=navigator.appName.subString(0,8); // für Netscape wird "Netscape" geliefert

2.1.2. **Codename Engine**

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  (navigator.appCodeName) ? document.writeln(navigator.appCodeName) :
    document.writeln("navigator.appCodeName wird vom Browser nicht unterstuetzt");
//-->
</SCRIPT>
```

2.1.3. **Browserversion: Hauptversion vor dem Punkt**

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  (navigator.appVersion) ? document.writeln(navigator.appVersion) :
    document.writeln("navigator.appVersion wird vom Browser nicht unterstuetzt");
//-->
</SCRIPT>
```

2.1.4. **Browserversion: Updatenummer nach dem Punkt**

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  (navigator.appMinorVersion) ? document.writeln(navigator.appMinorVersion) :
    document.writeln("navigator.appMinorVersion wird vom Browser nicht unterstuetzt");
//-->
</SCRIPT>
```

2.1.5. **Useragent**

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  (navigator.userAgent) ? document.writeln(navigator.userAgent) :
    document.writeln("navigator.UserAgent wird vom Browser nicht unterstuetzt");
//-->
</SCRIPT>
```

2.1.6. **Sicherheitseinstellungen**

```
<SCRIPT LANGUAGE="JavaScript">
```



```
<!--  
    (navigator.securityPolicy) ? document.writeln(navigator.securityPolicy) :  
        document.writeln("navigator.securityPolicy wird vom Browser nicht unterstuetzt");  
//-->  
</SCRIPT>
```



2.1.7. Browserplattform

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    (navigator.platform) ? document.writeln(navigator.platform) :
        document.writeln("navigator.platform wird vom Browser nicht unterstuetzt");
//-->
</SCRIPT>
```

liefert "Win32" oder "Win16" oder "WinCE"

andere Variante falls Useragent unterstützt wird:

```

    navigator.userAgent.indexOf("Win98") > 0
oder navigator.userAgent.indexOf("Windows 98") > 0      so Win98 erkannt

    navigator.userAgent.indexOf("Win95") > 0
oder navigator.userAgent.indexOf("Windows 95") > 0      so Win95 erkannt

    navigator.userAgent.indexOf("Linux") > 0              so Linux erkannt
```

2.1.8. Java-Applet-Darstellungsmöglichkeit

wenn navigator.javaEnabled() true liefert, so Applet-Darstellung möglich

2.2. System-Informationen

2.2.1. Betriebssystem unter dem der Browser läuft

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    (navigator.platform) ? document.writeln(navigator.platform) :
        document.writeln("navigator.platform wird vom Browser nicht unterstuetzt");
//-->
</SCRIPT>
```

liefert "Win32" oder "Win16" oder "WinCE"

andere Variante falls Useragent unterstützt wird:

```

    navigator.userAgent.indexOf("Win98") > 0
oder navigator.userAgent.indexOf("Windows 98") > 0      so Win98 erkannt

    navigator.userAgent.indexOf("Win95") > 0
oder navigator.userAgent.indexOf("Windows 95") > 0      so Win95 erkannt

    navigator.userAgent.indexOf("Linux") > 0              so Linux erkannt
```

2.1.2. Prozessortyp

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    (navigator.cpuClass) ? document.writeln(navigator.cpuClass) :
        document.writeln("navigator.cpuClass wird vom Browser nicht unterstuetzt");
//-->
</SCRIPT>
```

liefert	"x86"	für Intel
	"68K"	für Motorola
	"Alpha"	für Digital
	"PPC"	für Motorola

2.1.3. Online- oder Offline-Zustand

2.1.3.1. nicht Microsoft spezifisch

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    (navigator.onLine) ? document.writeln("online !") :
        document.writeln("navigator.onLine wird vom Browser nicht unterstuetzt oder offline");
//-->
</SCRIPT>
```



2.1.3.2. Microsoft spezifisch

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    if (navigator.appName == "Microsoft Internet Explorer")
    {
        (oCliCap.connectionType) ? dokument.writeln(oCliCap.connectionType) :
        document.writeln("oCliCap.connectionType wird vom IE nicht unterstuetzt");
    }
//-->
</SCRIPT>
```

liefert "lan" für mit Netzwerkverbindung online
 liefert "modem" für mit Modemverbindung online
 liefert "offline" für keine Onlineverbindung, also für offline

2.1.4. User-Informationen**2.1.4.1. Bildschirmauflösung des Users ermitteln**

```
fenster_breite=screen.width;
fenster_hoehe=screen.hight;
fenster_farbtiefe=screen.colorDepth
fenster_pixeltiefe=screen.pixelDepth
```

liefert z.B. 800 für Breite

600 für Höhe

Umrechnung screen.pixelDepth in Farben per Math.pow(2,screen.pixelDepth

Umrechnung screen.colorDepth in Farben per Math.pow(2,screen.colorDepth

2.1.4.2. User-Hinweis auf optimale Auflösung des**HTML-Dokumentes erzeugen**

```
<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function usermeldung()
        {
            var optimale_breite=800;
            var optimale_hoehe=600;

            if (document.all || document.layers)
            {
                if (
                    (screen.width != optimale_breite)
                    ||
                    (screen.height != optimale_hoehe)
                )
                {alert("Diese Seite ist optimiert auf "+optimale_breite+
                    "x"+optimale_hoehe+ "! Sie haben "+screen.width+
                    'x'+screen.heigt+" Wenn möglich bitte anpassen !");
                }
            }
        }
    //-->
    </SCRIPT>
</HEAD>

<BODY onload="usermeldung()">
</BODY>
</HTML>
```

2.1.4.3. Url ermitteln, von der der User herkommt

```
document.referrer
```

Hinweis: Per Firewall ist die Weitergabe der Url des Users unterbindbar.

3. HTML-Dokument**3.1. Informationen ermitteln (location-Objekt)**

```
<HTML>
<P>
```



```

<CENTER>
<SCRIPT LANGUAGE="JavaScript">
<!--
    with (document)
    {
        write("<TABLE WIDTH=60% border=1>");
        write("<TD WIDTH=30%>href</TD><TD WIDTH=70%>" + location.href
+ "</TD><TR>");
        write("<TD>protocol</TD><TD>" + location.protocol
+ "</TD><TR>");
        write("<TD>hostname</TD><TD>" + location.hostname
+ "</TD><TR>");
        write("<TD>host</TD><TD>" + location.host
+ "</TD><TR>");
        write("<TD>port</TD><TD>" + location.port
+ "</TD><TR>");
        write("<TD>pathname</TD><TD>" + location.pathname
+ "</TD><TR>");
        write("<TD>search</TD><TD>" + location.search
+ "</TD><TR>");
        write("<TD>hash</TD><TD>" + location.hash
+ "</TD><TR>");
        write("</TABLE>");
    }
// -->
</SCRIPT>
</CENTER>
<P>
</HTML>

```

Datum der letzten Änderung:

```

var Datum = document.lastModified;
var Tag = datum.substr(3,2);
var Monat = datum.substr(0,2);
var Jahr = datum.substr(6,4);

```

3.2. Möglichkeiten des Laden eines Dokumentes

innerhalb eines Frame: rechtlich nicht zulässig (Abmahnungsgefahr)

3.2.1. Laden eines fremden Dokumentes ohne Framedarstellung

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function laden()
    { location.href = "http://www.test.de"; }
// -->
</SCRIPT>
</HEAD>
<BODY>
<FORM>
<INPUT TYPE="button"
        VALUE="www.test.de anwaehlen "
        onClick="laden()">
</FORM>
</BODY>
</HTML>

```

3.2.2. Eigenes Dokument wird durch fremde Webseite geladen

3.2.2.1. Copyright-Meldung auf fremden Host erzeugen

Variante 1:

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript1.2">
<!--
    // In diesem Beispiel schreibt das Script Text auf die "entführte" Seite!
    var HostUrl="www.test.de";

```



```

    if(
        &&      (parent !=null)
                (parent != self)
    )
    {
        var host=parent.location.hostname;

        if(host != HostUrl)
        {
            document.write( " <CENTER>Diese Seite wurde ausgeliehen bei "
                            + "<A HREF=\"\"
                            + location.href
                            + \"\" TARGET=\"_parent\">"
                            + HostUrl
                            + "</A></CENTER>" )

        }
    }
}
//-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

Variante 2:

<BODY>

```

.....
if (
    &&      (parent != null)
            (parent != self)
)
{
    var  meine_url = "http://www.meine_angebot.de";
    var  mein_host_name = "http://" + meine_url;

    var  fremder_host_name = parent.location.hostname;

    if ( fremder_host_name != mein_host_name)
    {
        document.write(
            " Diese Seite liegt auf "
            + "<A "
            + "HREF=\"" + location.href + "\" "
            + "TARGET=\"_parent\" "
            + ">"
            + " und stammt von "
            + mein_host_name
            + "</A>"
        );
    }
}
.....
</BODY>

```

3.2.2.2. **Framedarstellung der eigenen Webseite sofort und ohne Bildschirmmeldung aktivieren**

Dieses Coding muss in jedes HTML-Dokument, das in einem Frame geladen wird. Bei Einzelaufwurf des Dokumentes wird automatisch die Seite mit dem FRAMESET aktiviert, also die vollständige Framedarstellung.

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript1.2">
<!--
    var EigenerHost="http://www.test.de";
    // window.location.hostname ist Leerkette, wenn Browser offline
    var StartSeite="_start.html";

    function InaktiveFrameDarstellungAktivieren()
    {
        if (top.frames.length == 0)
        {
            // keine Frame-Darstellung aktiv, also diese aktivieren

```



```
        top.location.href = StartSeite; // muss das FRAMESET enthalten !  
    }  
}
```



```

    if (      (parent != null) // Elternobjekt existiert
        && (parent != self) // Eltern sind vorhanden Eltern: dieses Dokument (self) ist ein Kind
    )
    {
        // aktuellen ElternHost ermitteln
        var ElternHost=parent.location.hostname;

        // ElternHost prüfen ob eigener und nicht leer, also Browser online ist
        if (      (ElternHost != "") // Browser ist online
            && (ElternHost != EigenerHost)
        )
        {
            // fremder Host, also als oberstes Fenster nun die Startseite anzeigen
            // und damit den eigenen Host aktivieren
            top.location.href=EigenerHost + '/' + Startseite;
        }
        else
        {
            // eigener Host und/oder Browser ist offline
            InaktiveFrameDarstellungAktivieren();
        }
    }
    else
    {
        // Elternobjekt existiert nicht und/oder dieses Dokument (self) ist kein Kind
        InaktiveFrameDarstellungAktivieren();
    }
}
//-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

3.2.3. Frame-Dokument mit zugehörigem Frameset laden

Sollte eine Frame per Suchmaschine indiziert sein und der User die Url des Frames bekommt, die er anklickt, so soll dieser Frame mit dem zugehörigen Frameset geladen werden.

```

<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript1.2">
<!--
    var Startseite="_start.html";

    if (top.frames.length == 0)
    {
        // keine Frame-Darstellung aktiv, also diese aktivieren
        top.location.href = Startseite; // muss das FRAMESET enthalten !
    }
}
//-->
</SCRIPT>

```

3.2.4. Mehrere Frameinhalte gleichzeitig ändern

HTML-Kodierung:

```

.....
<FRAMESET ...>
    <FRAME NAME="frame_name_1 SRC=html_datei_in_frame_1">
    .....
    <FRAME NAME="frame_name_n SRC=html_datei_in_frame_n">
</FRAMESET>

```

Auslösung der Frameänderung kann nur von einem Frame aus erfolgen
z.B. von html_datei_in_frame_1

```

....
<SCRIPT LANGUAGE="Javascript">
<!--
    function aenderung_variante1()
    {parent.frame_name_1.location="neue_html_datei_1";
    ....
    parent.frame_name_n.location="neue_html_datei_n";}
}

```




```

oder      function aenderung_variante2
            (neue_html_datei_1,...,neue_html_datei_n)
            {parent.frame_name_1.location=neue_html_datei_1+".html";
            .....
            parent.frame_name_n.location=neue_html_datei_n+".html";}
            // -->
</SCRIPT>
....
<INPUT TYPE="button"
        VALUE="Start der Aenderung" onClick="aenderung_variante1()">
oder      <A HREF="Javascript:aenderung_variante1()"> Start der Aenderung </A>
oder      <A HREF="Javascript:aenderung_variante1()">
            <IMG SRC=start_button.gif">
            </A>
bzw.      <A HREF=Javascript:aenderung_variante2
            ('neue_html_datei_1_ohne_suffix', .... 'neue_html_datei_n_ohne_suffix')
            > Start der Aenderung
            </A>

```

Hinweis: Standardsuffix einer HTML-Datei ist html oder htm.

3.2.5. Frame und Datenaustausch

Zwischen Frames können Daten ausgetauscht werden.

Bps: Adressierung eines Frame über das FRAMESET

```

<FRAMESET .... >
    <FRAMESET ..... >'
        <FRAME SRC="test.html"
            +NAME="test"
        >
        <FRAME SRC="test1.html"
            +NAME="test1"
        >
    </FRAMESET>
    <FRAME SRC="test2.html"
        NAME="test2"
    >
</FRAMESET>

```

parent.test2.location.href="neu.html"; // Laden von neu.html in den Frame test2

Bsp: auf Objekte im FRAMESET-Dokument durch ein FRAME-Dokument zugreifen

```

im FRAMESET-Dokument wird kodiert      var Kette="Hallo !";

im FRAME-Dokument wird auf Kette zugegriffen:  alert(parent.Kette);

```

Bsp: auf Objekte im FRAME-Dokument durch das FRAMESET-Dokument zugreifen

```

im FRAME-Dokument wird kodiert      var Kette="Hallo !";
                                     <FRAME ...NAME ="test2" ...>

im FRAMESET-Dokument wird auf Kette zugegriffen: alert(parent.test2.Kette);

```

3.2.6. History-Eintrag ersetzen

Bsp: in test.html kodieren window.location.replace("test.html");

3.3. HTML-Code in einem Fenster erzeugen und zugleich ausführen

Beispiel 1

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function erzeugeFenster()
        {

```



```

        neuesFenster = open("", "displayWindow", "HEIGHT=200,WIDTH=600");
        neuesFenster.document.open();
        with (neuesFenster.document)
        {
            write("<HTML><HEAD><TITLE>On the fly</TITLE></HEAD>");
            write("<H1>Dieses Fenster wurde on-the-fly erzeugt...</H1>");
            write("</HTML>");
        }
        neuesFenster.document.close();
    }

    function FensterSchliessen()
    { neuesFenster.close();}

// -->
</SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <INPUT  YPE="button"
            VALUE="Oeffnen"
            onClick="erzeugeFenster()">
        <INPUT  YPE="button"
            VALUE="Schliessen"
            onClick="FensterSchliessen()">
    </FORM>
</BODY>
</HTML>

```

Beispiel 2

```

<SCRIPT LANGUAGE="JavaScript">
<!--      document.write("<TABLE BORDER='1'><TR><TD>");
           document.write("Letzte Aenderung: " + document.lastModified);
           document.write("</TD></TR></TABLE>");

```

//-->

</SCRIPT>

Netscape vertragt max. 256 Zeichen pro Zeile.

Beispiel 3:

```

<SCRIPT ...>
<!--
    function eigenes_fehler_fenster(meldungs_text,url,zeilen_nr)
    {
        // Fehlermeldung bilden
        var url_als_kette=url.toString();

        var zeilen_nr_als_kette=zeilen_nr.toString();

        var meldung_komplett=meldungs_text + url_als_kette + zeilen_nr_als_kette;

        var fenster=window.open();

        with (fenster.document)
        {
            open("text/html"); // HTML-Dokument im Fenster erzeugen

            writeln(
                "<HTML><HEAD><TITLE>Private Errormeldung</TITLE></HEAD>");

            writeln("<BODY><H1>Fehlermeldung</H1>");

            writeln(meldung_komplett);

            writeln(
                "<FORM><INPUT TYPE=BUTTON VALUE='OK'"
                + " onClick='self.close()'"
            );
            // Button anklicken, damit das Meldungs-Fenster geschlossen wird

            close(); //HTML-Dokument schliessen
        }
    }

```



```

    }
    return true;          // muss true liefern für window.onerror
}

window.onerror= eigenes_fehler_fenster; // NICHT onError kodieren !
// keine () kodieren, da sonst die Funktion
// sofort ausgeführt wird !!
// window.onerror verlangt die Zuweisung
// von true für Abschaltung des
// Standard-onerror

// -->
</SCRIPT>
----
<BODY>
...
</BODY>

```

Hinweis: Innerhalb von “ “ ist für eine Zeichenketten ‘ ‘ zu kodieren bzw. “ mit \“ zu entwerten !!

3.4. Umleiten auf andere Url

Beispiel 1

```

<META HTTP.EQUIV="Refresh"
      CONTENT="3;URL=http://....."
>

```

// 3 Sekunden warten, dann umleiten nach http://

Beispiel 2

```

<SCRIPT LANGUAGE="JavaScript">
<!--
      top.location.HREF="http:// ....."
//-->
</SCRIPT>

```

Beispiel 3

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
      function laden()
      { location.href = "http://www.test.de";}
    // -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT TYPE="button"
            VALUE="www.test.de anwaehlen "
            onClick="laden()">
    </FORM>
  </BODY>
</HTML>

```

Beispiel 4:

```

<BODY onLoad="setTimeout('location.reload(true),anzahl_millisekunden');">

```

1s = 1000 ms; 1min = 60000 ms; 5min = 300000 ms; etc.

Beispiel 5:

```

<A HREF="http://www.test.de" onClick="return confirm('Wirklich?')">Zu www.test.de gehen</A>

```



3.5. *Update-Datum anzeigen*

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    document.write(          "<FONT color=ff0000>"
+                           "Letztes Update am: "
+                           "</FONT>" + document.lastModified);
//-->
</SCRIPT>
```



3.6. Suchmaschine einbinden

3.6.1. für Microsoft Internet Explorer

```
markierter_text=document.selection.createRange().text;           oder markierter_text=prompt('Suchbegriff: ');
suchmaschinen_url='http:// .....';
suchmaschinen_parameter='.....';

if (markierter_text)
{
    location.HREF=suchmaschinen_url + suchmaschinen_parameter + escape(markierter_text);}
else
{
    location.HREF=suchmaschine_url;
}
```

Beispiel für altavista: suchmaschinen_url 'http://altavista.de/'

suchmas chinen_parameter 'cgi-bin/query?pg=q&what=web&q='

Funktion escape setzt Umlaute und Leerzeichen in korrekte Darstellung um

3.6.2. für Netscape

```
markierter_text=document.getSelection();           oder markierter_text=prompt('Suchbegriff: ');
suchmaschinen_url='http:// .....';
suchmaschinen_parameter='.....';

if (markierter_text)
{
    location.HREF=suchmaschinen_url + suchmaschinen_parameter + escape(markierter_text);}
else
{
    location.HREF=suchmaschine_url;
}
```

Beispiel für altavista: suchmaschinen_url 'http://altavista.de/'

suchmas chinen_parameter 'cgi-bin/query?pg=q&what=web&q='

Funktion escape setzt Umlaute und Leerzeichen in korrekte Darstellung um

3.7. Plugin-Prüfung

Beispiel: auf Adobe Acrobat-Reader-Plugin prüfen

```
<HTML>
  <HEAD>
    <TITLE> ..... </TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript1.2">
      <!--
        if (navigator.plugin["Adobe Acrobat"] != null) // eckige Klammern !
        {
          document.write("<EMBED SRC='test.pdf' WIDTH=600 HEIGHT=800>");
          document.write("<NOEMBED> .....</NOEMBED>");
        }
        else
        {
          document.write("Kein Adobe-Plugin !"); }
      //-->
    </SCRIPT>
  </BODY>
</HTML>
```

Beispiel: alle Plugins auflisten

```
<HTML>
  <HEAD>
    <TITLE> ..... </TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript1.2">
      <!--
        for (i=0; i < navigator.plugins.length; i++)
        {
          document.writeln(navigator.plugins[i].name);
        }
      //-->
    </SCRIPT>
  </BODY>
</HTML>
```



```

        document.writeln(navigator.plugins[i].description);
        document.writeln(navigator.plugins[i].filename);
    }
    //-->
</BODY>
</HTML>

```

3.8. E-Mail

Beispiele für mailto-Formen

```

Standard      mailto:aaa@bbb
carbon copy    mailto:aaa@bbb?cc=mailto:ccc@ddd
                (mit Kopie an anderen Empfänger)
blind copy     mailto:aaa@bbb?bcc=mailto:ccc@ddd
                (mit Blind-Kopie an anderen Empfänger)
carbon copy und Betreff    mailto:aaa@bbb?cc=mailto:ccc@ddd&subject=betreff_text
carbon copy und Betreff und Mailtext
                        mailto:aaa@bbb?cc=mailto:ccc@ddd&subject=betreff_text&body=mail_text

```

Betreff/Mailtext analog für Standard und blind copy

3.8.1. Email senden per Formular

Beispiel 1

```

<form ACTION="mailto:test@web.de" method=GET>
  <input name=subject type=hidden value="Testemail">
  <input type=submit value="Email-Programm mit Standardkonto aktivieren">
</form>

```

action belegt das Zielkonto, an das gemailt werden soll.

input-subject ist unsichtbar und füllt Betreff-Zeile der Email mit Wert.

ohne das submit-Button wird aktion nicht ausgelöst

Das unter windows eingebundene Standard-Emailprogramm wird aktiviert,
wobei das Standardkonto benutzt wird.

Es ist also nicht möglich, im Form das Konto zu selektieren, da
das Emailprogramm per Form nicht ansteuerbar für Sende-Kontenwahl

ob fireEvent() auf input-subject per onclick möglich ist, muss geprüft werden:

Wenn ja, so ist input-subject mit style.visibility="hidden"
und onclick umleiten von anderem Element auf das Button möglich

Beispiel 2

```

<HTML>
<HEAD>
<STYLE TYPE="text/css">
<!--
body { background:#FFFFFF; margin-left: 5%; margin-right: 5% }
-->
</STYLE>

```

```

<SCRIPT LANGUAGE="JavaScript">
<!--

```

```

function MailToPruefen(Kette)
{
    // true wenn alles okay
    var ReturnWert=false;

    var Laenge=Kette.length;
    var Position=0;

    // Länge prüfen
    if (Laenge == 0)
    { alert('Mail-To ist leer !'); }
    else
    {
        // Position des @ ermitteln
    }
}

```



```

        Position=Kette.indexOf("@");

        if (Position == -1)
        {alert('Mail-To @ fehlt !');}
        else
        {
            if (
                (Laenge < 3)                // nur 2 Zeichen, also fehlt vor oder hinter dem @ ein Zeichen
                || (Position == 0)          // vor dem @ fehlt ein Zeichen
                || (Laenge == (Position +1)) // hinter dem @ fehlt ein Zeichen
            )
            {alert('Mail-To ist falsch kodiert !');}
            else
            {ReturnWert=true;}
        }
    }

    return ReturnWert;
}

function EmailStringPruefen(Kette, Parameter)
{
    // true wenn alles okay
    var ReturnWert=false;

    var Laenge=Kette.length;
    var Position=0;

    // Länge prüfen
    if (Laenge > 0)
    {
        // Position des @ ermitteln
        Position=Kette.indexOf("@");

        if (Position == -1)
        {alert(Parameter + ': @ fehlt !');}
        else
        {
            if (
                (Laenge < 3)                // nur 2 Zeichen, also fehlt vor oder hinter dem @ ein Zeichen
                || (Position == 0)          // vor dem @ fehlt ein Zeichen
                || (Laenge == (Position +1)) // hinter dem @ fehlt ein Zeichen
            )
            {alert(Parameter + ' falsch kodiert !');}
            else
            {ReturnWert=true;}
        }
    }
    else
    {ReturnWert=true;}

    return ReturnWert;
}

function TrennZeichenEinfuegen(Kette)
{
    // kopieren in den Rückgabewert
    var RueckgabeString = Kette;

    // prüfen auf vorhandenes ?, das als ERSTES Trennzeichen kommen muss
    if (RueckgabeString.indexOf("?") == -1)
    {
        // kein ? vorhanden, also anhängen
        RueckgabeString += "?";
    }
    else
    {
        // vorhanden, also nur noch Trennzeichen & anhängbar für Aufzählung
        RueckgabeString += "&";
    }
}

```



```

    return RueckgabeString;
}

function ErzeugeEmailUrlEventuellImLink(sourceForm, targetField, urlType)
{
    // Beispiele:
    // Standard                                mailto:aaa@bbb
    // carbon copy also mit Kopie an anderen Empfänger mailto:aaa@bbb?cc=mailto:ccc@ddd
    // blind copy                               mailto:aaa@bbb?bcc=mailto:ccc@ddd
    // carbon copy und Betreff                  mailto:aaa@bbb?cc=mailto:ccc@ddd&subject=betreff_text
    // carbon copy und Betreff und Mailtext     mailto:aaa@bbb?cc=mailto:ccc@ddd&subject=betreff_text&body=mail_text

    var MailTO    = sourceForm.FormularTO.value;
    var MailCC    = sourceForm.FormularCC.value;
    var MailBCC   = sourceForm.FormularBCC.value;
    var MailBETREFF = sourceForm.FormularBETREFF.value;
    var MailTEXT   = sourceForm.FormularTEXT.value;
    var MailLINKTEXT = sourceForm.FormularLINKTEXT.value;

    var EmailUrl = "";

    // Pflichteingabe prüfen
    if (! MailToPruefen(MailTO))
    {
        sourceForm.FormularTO.focus();

        // onclick verlangt return true;
        return true;
    }
    else
    {
        // mit @ also als Anfangswert
        EmailUrl = 'mailto:' + MailTO;

        // Prüfungen aller optionalen Parameter in der notwendigen Reihenfolge

        if (! EmailStringPruefen(MailCC, "Cc"))
        {
            sourceForm.FormularCC.focus();

            // onclick verlangt return true;
            return true;
        }

        if ( MailCC != "" )
        {
            EmailUrl = TrennZeichenEinfuegen(EmailUrl);
            EmailUrl += "CC=" + MailCC;
        }

        if (! EmailStringPruefen(MailBCC, "Bcc"))
        {
            sourceForm.FormularBCC.focus();

            // onclick verlangt return true;
            return true;
        }

        if ( MailBCC != "" )
        {
            EmailUrl = TrennZeichenEinfuegen(EmailUrl);
            EmailUrl += "BCC=" + MailBCC;
        }

        if (MailBETREFF != "")
        {
            EmailUrl = TrennZeichenEinfuegen(EmailUrl);
            EmailUrl += "Subject=" + escape(MailBETREFF); // Hexakodierung nötig
        }
    }
}

```




```

        if (MailTEXT != "")
        {
            EmailUrl = TrennZeichenEinfuegen(EmailUrl);
            EmailUrl += "Body=" + escape(MailTEXT);           // Hexakodierung nötig
        }

        // eventuell Email-Url im Link erzeugen
        if (urlType != "url")
        { EmailUrl = '<A HREF=' + EmailUrl + '>' + MailLINKTEXT + '</A>'; }

        // Email-Url nach Formularfeld
        targetField.value = EmailUrl;

        // und Focus auf dieses Formularfeld
        targetField.focus();
        targetField.select();

        // onclick verlangt return true;
        return true;
    }
}

function TesteEmailUrl(Kette)
{
    // kopieren in die WindowUrl
    var WindowUrl = Kette;

    // prüfen ob HREF enthalten
    if (WindowUrl.indexOf("HREF=") != -1)
    {
        WindowUrl = WindowUrl.substring(WindowUrl.indexOf("HREF=")+6, WindowUrl.indexOf(">")-1);
    }

    // durch das @ wird das Standard-Email-Programm aufgerufen
    window.location = WindowUrl;
}

function ZeigeEmailUrl(linkText)
{
    alert("komplette Email-URL lautet:\n\n" + linkText);
}

// -->
</SCRIPT>

</HEAD>
</HEAD>
<BODY BGCOLOR="#ffffff">
<FONT FACE="Arial, Helvetica" SIZE=-1>
<FORM NAME="Formular">
    <CENTER>
        <TABLE BORDER=0 CELLSPACING=0 CELLPADDING=2 BGCOLOR="#ffff99" WIDTH="100%" >
            <TR ALIGN="LEFT" VALIGN="TOP">
                <TD COLSPAN="4">
                    <CENTER>
                        <B>
                            <FONT COLOR="#408080" SIZE=+1>Gib hier die Message-Detail ein:</FONT>
                        </B>
                    </CENTER>
                </TD>
            </TR>
            <TR ALIGN="LEFT" VALIGN="TOP">
                <TD>
                    <FONT FACE="Arial, Helvetica" SIZE=-1>
                        <B>Mail-To (Pflichteingabe) :</B>
                    </FONT>
                </TD>
            </TR>
        </TABLE>
    </CENTER>
</FORM>

```



```

        <TD>
            <INPUT TYPE="text" NAME="FormularTO" SIZE=30 STYLE="background-color:#ffcc66">
        </TD>
        <TD>
            <FONT FACE="Arial, Helvetica" SIZE=-1>
                <B>Cc:</B>
            <FONT>
        </TD>
        <TD>
            <INPUT TYPE="text" NAME="FormularCC" SIZE=30 STYLE="background-color:#ffcc66">
        </TD>
    </TR>
    <TR ALIGN="LEFT" VALIGN="TOP">
        <TD ALIGN="LEFT">
            <FONT FACE="Arial, Helvetica" SIZE=-1>
                <B>Bcc:</B>
            <FONT>
        </TD>
        <TD>
            <INPUT TYPE="text" NAME="FormularBCC" SIZE=30 STYLE="background-color:#ffcc66">
        </TD>
        <TD>
            <FONT FACE="Arial, Helvetica" SIZE=-1>
                <B>Betreff:</B>
            <FONT>
        </TD>
        <TD>
            <INPUT TYPE="text" NAME="FormularBETREFF" SIZE=30
            STYLE="background-color:#ffcc66">
        </TD>
    </TR>
    <TR ALIGN="LEFT" VALIGN="TOP">
        <TD ALIGN="LEFT">
            <FONT FACE="Arial, Helvetica" SIZE=-1>
                <B>Mail-Text:</B>
            <FONT>
        </TD>
        <TD COLSPAN="4">
            <TEXTAREA NAME="FormularTEXT" WRAP="VIRTUAL" COLS=63
            ROWS=3 STYLE="background-color:#ffcc66"></TEXTAREA>
        </TD>
    </TR>
    <TR ALIGN="LEFT" VALIGN="TOP">
        <TD ALIGN="LEFT">
            <FONT FACE="Arial, Helvetica" SIZE=-1>
                <B>Link Text:</B>
            <FONT>
        </TD>
        <TD COLSPAN="3">
            <INPUT TYPE="text" NAME="FormularLINKTEXT" SIZE=73
            STYLE="background-color:#ffcc66">
        </TD>
    </TR>
    <TR>
    <TR>
        <TD COLSPAN="4">
            <CENTER>
                <INPUT TYPE="Reset"
                    NAME="Clear"
                    Value="Daten loeschen"
                >
                <INPUT TYPE="button"
                    NAME="CreateURL"
                    VALUE="Email-URL erzeugen"
                    onClick="ErzeugeEmailUrlEventuellImLink(document.Formular, document.Formular.text, 'url')"
                >
                <INPUT TYPE="button"
                    NAME="CreateAnchor"
                    VALUE="Link mit Email-Url erzeugen"
                    onClick="ErzeugeEmailUrlEventuellImLink(document.Formular, document.Formular.text, 'anchor')"

```



```

        >
        </CENTER>
    </TD>
</TR>
<TR>
    <TD COLSPAN="4">
        <CENTER>
            <FONT COLOR="#408080" SIZE="+1">
                <B>Ergebnis:</B>
            </FONT>
        </CENTER>
    </TD>
</TR>
<TR>
    <TD ALIGN="LEFT">
        <FONT FACE="Arial, Helvetica" SIZE="-1">
            <B>mailto URL:</B>
        </FONT>
    </TD>
    <TD COLSPAN="3">
        <INPUT TYPE="text" NAME="text" SIZE=73 STYLE="background-color:#ffcc66">
    </TD>
</TR>
<TR>
    <TD COLSPAN="4">
        <CENTER>
            <INPUT TYPE="button"
                NAME="Test"
                VALUE="Standard-Email-Programm mit dieser Email-Url starten"
                onClick="TesteEmailUrl(document.Formular.text.value)"
            >
            <INPUT TYPE="button"
                NAME="View"
                VALUE="komplette Email-Url ansehen"
                onClick="ZeigeEmailUrl(document.Formular.text.value)"
            >
        </CENTER>
    </TD>
</TR>
</TABLE>
</CENTER>
</FORM>
</FONT>
</BODY>
</HTML>

```

3.8.2. Spam-Schutz

Suchroboter grasen E-Mail-Adressen auf Seiten ab, um an diese Spams zu schicken
E-Mail-Adressen werden aber von Suchmaschinen nicht erkannt, wenn

Variante 1: alle Zeichen der E-Mail in der Form
&#xxx; kodiert sind
mit xxx als ALT-Kode
wobei zwischen den &#xxx keine Blanks kodiert werden dürfen

Bsp.: Doppelpunkt :
 @ @
 m m

Variante 2:

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript1.2">
    <!--
        // test@test.de
        var user_name="test";
        var host_name="test.de";

        document.write(
            '<A HREF="mailto:'
            + user_name

```



```

        + '@'
        + host_name
        + '>'
        + user_name
        + '@'
        + host_name
        + '</A>'
    );

    // -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

3.8.3. Email-Absendung mit Bestätigung für User

<INPUT TYPE="SUBMIT" ... onClick="alert('Die eMail wurde abgeschickt.')">

3.8.4. Email und Stringkonvertierung ab Internet Explorer 5.5

(Encoding and Decoding URI's)

URI: Universal Resource Index als Standardformat im ISO Latin-1 Zeichensatz (8859-1)
 Ein nicht-alphanumerisches Zeichen muss als "%xy" kodiert werden.
 Hinweis: # ist ein alphanumerisches Zeichen (Nummernkreuz)

Ab dem Internet Explorer ab 5.5 können folgende Methoden für die Konvertierung benutzt werden:

encodeURIComponent(string), Konvertierung von String zu URI

Bsp: var URI-String = encodeURIComponent("My phone # is 123-456-7890");
 erzeugt "My%20phone%20#%20is%20123-456-7890"

decodeURI(uri)

Konvertierung von URI zu String

Bsp: var String = decodeURI("My%20phone%20#%20is%20123-456-7890");
 erzeugt "My phone # is 123-456-7890"

3.9. Textdaten-Übergabe zwischen zwei HTML-Dokumenten (Daten aus TEXTAREA)

3.9.1. Textdaten-Übergabe durch an Url angehängte Textdaten

quelle.htm: lädt ziel.htm
 übergibt Textdaten an ziel.htm

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function ziel_seite_laden_mit_datenuebergabe(uebergabe_datan)
    {location.href = "ziel.htm?" + escape(uebergabe_datan);}
    // Url von ziel.htm als Suchbegriff mit angehangenen Daten merken,
    // die per escape() in das Url-Format konvertiert wurden
    // -->
</SCRIPT>
</HEAD>

<BODY>
    An ziel.htm zu &uuml;bergabenden Text eingeben:
    <FORM>
        <TEXTAREA      NAME=eingabe
                        ROWS=5
                        COLS=40>

        </TEXTAREA>
        <BR>
        <INPUT  TYPE=button
                VALUE="Zielseite laden"
                onClick=" ziel_seite_laden_mit_datenuebergabe
                    (this.form.eingabe.value)">
    </FORM>
</BODY>
</HTML>

```



ziel.htm: Wird durch **quelle.htm** geladen
erhält Textdaten von **quelle.htm**

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        datenuebernahme()
        {
          uebernahme_datens= location.search;
            // Url mit angehangenen Daten holen
            // search liefert ?daten
          uebernahme_datens= uebernahme_datens.substring(1, uebernahme_datens.length);
            // ? abschneiden
          document.ausgabe.ausgabefeld.value=unescape(uebernahme_datens);
            // unescape: von Url-Format nach Zeichenkette
        }
      // -->
    </SCRIPT>
  </HEAD>

  <BODY onLoad=" datenuebernahme ()">
    Von quelle.htm &uuml;bernommener Text:
    <FORM NAME=ausgabe>
      <TEXTAREA      NAME=ausgabefeld
        ROWS=10 COLS=40>
      </TEXTAREA>
    </FORM>
  </BODY>
</HTML>
```

3.9.2. Textdaten-Übergabe durch Fenster-Handle

Die Textdaten und die der Javascript-Code, der die Textdaten verarbeitet, sind **getrennte** HTML-Dokumente.

HTML-Dokument, das die Textdaten enthält:

```
<HTML>
<HEAD>
</HEAD>
<BODY>
  <FORM NAME=formular>
    <TEXTAREA NAME=text_area>
      // Hier die Textdaten als Wert von TEXTAREA
    </TEXTAREA>
  </FORM>
</BODY>
</HTML>
```

Dokument, das die Textdaten verarbeitet:

```
<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      function TextAreaWertLesen(url)
      {
        // url enthält den Pfad und den Namen des Dokumentes, dass die
        // Textarea-Werte enthält
        var handle = window.open(url_der_datendatei, "", "width=100,height=100");
          // Handle erzeugen und Fenster mit dem Textarea anzeigen
          // (Fenstergröße ist egal)
        var data = handle.document.forms[formular].elements[text_area].value;
          // Textarea-Element-Wert lesen
        handle.close();
        return data;
      }
    // -->
  </SCRIPT>
</HEAD>
```



```
<BODY>
</BODY>
</HTML>
```

Nachteil dieser Variante ist, dass in beiden Dokumenten die logischen Namen vom Formular und dem Textarea identisch kodiert werden müssen, also eine doppelte Verwaltung nötig ist.

Nachfolgende Beispiele zeigen den Zugriff innerhalb eines **gemeinsamen** HTML-Dokumentes **ohne Fensterhandle**:

Beispiel 1

```
<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      function TextAreaWertLesen()
      {
        var data = document.forms[formular].elements[text_area].value;
        // Textarea-Element-Wert lesen
        return data;
      }
    // -->
  </SCRIPT>
</HEAD>
<BODY>
  <FORM NAME=formular>
    <TEXTAREA NAME=text_area>
      // Hier die Textdaten als Wert von TEXTAREA
    </TEXTAREA>
  </FORM>
</BODY>
</HTML>
```

Hinweis: Die Funktion TextAreaWertLesen muss natürlich noch an gewünschter Stelle aufgerufen werden durch z.B. TextAreaWert=TextAreaWertLesen();

Da der Body-Teil zuerst ausgeführt wird, sind die logischen Namen für das Formular und den Textarea beim Aufruf der Funktion TextAreaWertLesen bekannt, also gültig.

Beispiel 2

```
<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var      TextAreaWert="test"; // hier mit konkreten Werten zu belegen

      function TextAreaWertLesen()
      {
        var data = document.forms[formular].elements[text_area].value;
        // Textarea-Element-Wert lesen
        return data;
      }

      document.write(
        '<FORM NAME=formular>'
        + '<TEXTAREA NAME=text_area>'
        + '      TextAreaWert'
        + '</TEXTAREA>'
        + '</FORM>'
      );
    // -->
  </SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Hinweis: Die Funktion TextAreaWertLesen muss natürlich noch an gewünschter Stelle aufgerufen werden durch z.B. TextAreaWert=TextAreaWertLesen();



Da im Head-Teil die Anweisung document.write SOFORT ausgeführt wird, sind die logischen Namen für das Formular und den Textareabeim Aufruf der Funktion TextAreaWertLesen bekannt, also gültig.

3.10. **Programmierung eines Java-Stacks zum temporären Zwischenspeichern von Daten in den RAM**

Die Reihenfolge der Daten auf dem Stack ist **UNBEDINGT** zu beachten:
Der zuletzt auf dem Stack abgelegte Wert ist zuerst auszulesen.
(Last in First Out = LiFo)

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        // Neues Stack-Objekt erzeugen
        java_objekt_stack = new java.util.Stack();

        function in_stack_ablegen(wert)
        {java_objekt_stack.push(wert);}

        function aus_stack_holen()
        {return java_objekt_stack.pop();}

      // -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT TYPE="text"
        NAME="wert">
      <INPUT TYPE="button"
        VALUE="In Stack ablegen"
        onClick=" in_stack_ablegen (this.form.wert.value)">
      <INPUT TYPE="button"
        VALUE="aus Stack holen"
        onClick="alert(aus_stack_holen ())">
    </FORM>
  </BODY>
</HTML>
```

3.11. **Microsoft Internet Explorer**

3.11.1. **Abschaltung der automatischen Umrandung von angeklickten Objekten**

für gesamtes Dokument:

```
<BODY onclick="if (document.all) body.focus();"> ..... </BODY>
```

im Link:

```
<A HREF="...." onclick="if (document.all) body.focus();"> ....
```

3.11.2. **Bookmark per Script erzeugen**

Beispiel 1:

```
<A HREF="javascript:window.external.AddFavorite('http:.....','freier_text')"></A>
```

Achtung: Diese Zeile MUSS immer eine einzige Zeile sein, da kein Umbruch möglich !

Beispiel 2:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  function Bookmark_Setzen()
  {
    if (window.external()
    {
      external.AddFavorit(location.href,document.title)}
    else {alert("Kein IE")}
  }
//-->
</SCRIPT>
```

....

```
<FORM>
```



```

        <INPUT                TYPE="button"
                           VALUE="Bookmark setzen"
                           onClick="Bookmark_Setzen()"
        >
    </FORM>

```

Beispiel 3:

```

<SCRIPT LANGUAGE="JavaScript">
<!--
    if (
        (navigator.appName == "Microsoft Internet Explorer")
        &&    (parseInt(navigator.appVersion) >= 4)
    )
    {
        var url="http://www.test.com.com";
        var title="TEST.HTML";
        document.write('<A HREF="javascript:window.ext';
        document.write('external.AddFavorite(url,title);" ');
        document.write('onMouseOver=" window.status=';
        document.write('"Unsere Seite als Favorit!'; return true ');
        document.write('"onMouseOut=" window.status=';
        document.write('"'; return true ');
        document.write('"> Unsere Seite als Favorit!</A>');
    }
    else
    {
        var msg = "Netscape: Unsere Seite als Bookmark: ";
        if(navigator.appName == "Netscape") {msg += " (CTRL-D)";}
        document.write(msg);
    }
// -->
</SCRIPT>

```

3.11.3. Reload des Dokumentes nach Resize des Browserfenster

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function neuLaden()
        {
            if (document.all)
            {history.go(0);}
        }
    // -->
    </SCRIPT>
</HEAD>
<BODY onResize=neuLaden();">
</BODY>

```

3.11.4. Filter für HTML-Elemente ab IE 5.5

Filter für HTML-Elemente werden in Verbindung mit Style-Sheet eingesetzt.

Es können beim IMG-Tag alle Bildformate, die der Browser kennt, verarbeitet werden, also z.B. GIF und JPG. Speziell für das DIV-Tag sind Filter sehr sinnvoll anwendbar.

3.11.4.1. Filterarten

```

filter: alpha(opacity=deck_kraft)
filter: blur(add= ...,strength=..., direction=....)
        add=0    für Original nicht überlagern
        add=1    für Original überlagern
filter: chroma(color=....)
filter: dropShadow(color=schatten_farbe, offX=..., offY=...)    Doppelschatten
filter: flipV            spiegeln am linken Objektrand
filter: flipH            spiegeln am unteren Objektrand
filter: glow(color=..., strength= ....)                        glühen
filter: gray eingrauen
filter: invert            invers
filter: mask(color=...)
filter: revealTrans(duration=..., transition=...)
        zu transition:    0 Box in
                        1 Box out

```



2 Circle in
 3 Circle out
 4 Wipe up
 5 Wipe down
 6 Wipe right
 7 Wipe left
 8 Vertical blinds
 9 Horizontal blinds
 10 Checkerboard across
 11 Checkerboard down
 12 Random dissolve
 13 Split vertical in
 14 Split vertical out
 15 Split horizontal in
 16 Split horizontal out
 17 Strips left down
 18 Strips left up
 19 Strips right down
 20 Strips right up
 21 Random bars horizontal
 22 Random bars vertical
 23 Random filter from list above

filter: shadow(color=schatten_farbe, direction=.....)
 filter: wave(strength=..., freq=...)
 filter: xray eingrauen in invers
 filter: progid:DXImageTransform.Microsoft.MotionBlur(strength=..., direction=...)
 filter: progid:DXImageTransform.Microsoft.Emboss
 filter: progid:DXImageTransform.Microsoft.Wheel(duration=...)

Hinweis:

direction=grad_0_bis_360_als_richtung
 strength für Stärke des Effektes

3.11.4.2. Prinzipien der Kodierung von Filtern

Variante 1

Im HTML-Element wird das CLASS-Attribut verwendet.Innerhalb <HEAD> ..</HEAD> wird mit dem <STYLE>-Tag die CSS-Klasse mit den Alpha-Filtern kodiert.

```
<HTML>
<HEAD>
  <STYLE>
    tag_name_1.filter_name_1 { filter: .....; }
    .....
    tag_name_1.filter_name_n { filter: .....; }
    ....
    tag_name_n.filter_name1 { filter: .....; }
    .....
    tag_name_n.filter_namen { filter: .....; }
  </STYLE>
</HEAD>
<BODY>
  <tag_name_1 ... CLASS="filter_name_1"> .... </tag_name_1>
  .....
</BODY>
</HTML>
```

Es können pro Tag mehrere Filter kodiert werden.

Jedes HTML-Element laut Tag kann aber nur einen Filter besitzen. Es können also mehrere HTML-Elemente vom gemeinsamen Tag aber mit verschiedenen Filtern kodiert werden.

Der IE bietet aber die Layer-Technik z.B. anhand des DIV-Tags, so dass Überlagerungen an einer gemeinsamen Position möglich sind.

Variante 2

Direkt im HTML-Element wird das STYLE-Attribut verwendet. Damit entfällt die Klassendeklaration per <STYLE> innerhalb <HEAD>. Es wird aber ein internes Filterprogramm benötigt z.B. für Fade. Typisches HTML-Element ist der <DIV>-Tag, der auch die Layer-Technik unterstützt.

```
<HTML>
```



```

<HEAD>
</HEAD>
<BODY>
    <DIV ID="DIV_ID"
        STYLE=".....;filter:progid:DXImageTransform.Microsoft.Fade(duration=y);">
        ....
    </DIV>
</BODY>
</HTML>

```

y ist numerischer Wert der Dauer

Zugriffvarianten anhand eines IMG:

```

<IMG ID="Bild" STYLE="filter: wave(strength=0, freq=1)" SRC="test.gif" BORDER="0">

document.all.Bild.filters[0]
document.all.Bild.filters["wave"]
document.all.Bild.filters.wave
document.all.Bild.filters.item(0)
document.all.Bild.filters.item("wave")

```

Variante 3:

Im Falle von Filter-Effekt-Sequenzen als Animation und für progid:DXImageTransform.Microsoft.filter:.....

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE= "JavaScript">

function playTransition()
{
    document.all.Bild.filters.blendTrans.Apply();
    document.all.Bild.src = "test.gif";
    document.all.Bild.filters.blendTrans.Play();
}
</SCRIPT>
</HEAD>

<BODY onLoad="playTransition()">
    <IMG ID="Bild" STYLE="filter:blendTrans(duration=3)" SRC="square.gif" BORDER="0">
</BODY>
</HTML>

```

Variante 4:

im META-Tag innerhalb <HEAD> ... </HEAD>

```

<META HTTP-EQUIV=      "Page-Enter"
      oder              "Page-Exit"
      oder              "Site-Enter"
      oder              "Site-Exit"
      CONTENT="revealTrans(duration=10, transition=8)">

```

3.11.4.3. Alpha-Filter für HTML-Grafik-Elemente ab IE 5.5

Grafiken können transparent dargestellt werden: Die Deckungsstärke ist einstellbar.

3.11.4.3.1. Beispiel für transparentes INPUT-Element

Das INPUT-Element kann den Hintergrund des Dokumentes in der INPUT-Dimension stufig abdecken.

```

<HTML>
<HEAD>
<STYLE>
    INPUT.filter_name {filter:Alpha(opacity=y);}
</STYLE>
</HEAD>
<BODY>
    <INPUT TYPE="button" .... CLASS="filter_name">
</BODY>
</HTML>

```

opacity: Deckkraft



```

y von 0 bis 100
0           keine Deckkraft
100        alles abdeckend

```

3.11.4.3.2. Beispiel für transparentes Vordergrund-Bild auf dem Hintergrundbild des Dokumentes

Ziel ist es, eine per Filter gesteuerte Deckkraft zu erreichen. Beispiel dafür: Ein Bild im Vordergrund soll sich in den Hintergrund aus einem Bild einprägen, also mit dem Hintergrund per Deckkraft verschmelzen. Auch andersrum wäre es möglich: Der Hintergrund erscheint optisch vor dem Vordergrundbild.

```

<HTML>
<HEAD>
<STYLE>
    IMG.filter_name {filter:Alpha(opacity=y);}
</STYLE>
</HEAD>
<BODY BACKGROUND="hintergrund_bild.gif">
    <IMG SRC="bild.gif".... CLASS="filter_name">
</BODY>
</HTML>

```

```

opacity:      Deckkraft
               y von 0 bis 100
               0           keine Deckkraft
               100        alles abdeckend

```

3.11.4.3.3. Beispiel für transparente Überlagerung zweier Bilder

Ziel ist es, eine per Filter gesteuerte Deckkraft zu erreichen. Beispiel dafür: Ein Bild im Vordergrund soll mit einem zweiten in der Dimension kleineren Bild überlagert werden, so dass dieses kleinere mit dem größeren per Deckkraft verschmilzt. Auch andersrum wäre es möglich: Das größere Bild erscheint optisch vor dem kleineren.

```

<HTML>
<HEAD>
<STYLE>
    IMG.filter_name {filter:Alpha(opacity=y);}
</STYLE>
</HEAD>
<BODY>
    // je nach Variante die Reihenfolge der IMG verändern.
    <IMG SRC="hintergrund_bild.gif"....>
    <BR>
    <IMG SRC="bild.gif".... CLASS="filter_name">
</BODY>
</HTML>

```

```

opacity:      Deckkraft
               y von 0 bis 100
               0           keine Deckkraft
               100        alles abdeckend

```

3.11.4.3.4. Beispiele für Aus-und Einblende eines Bildes

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        var Sichtbar=true;    // DIV ist natürlich nach dem Dokument-Laden sichtbar
        var StandardFarbe="green";

        function Blenden()
        {
            // Fade zurücksetzen
            DIV_ID.filters[0].Apply();

            if (Sichtbar)
            {
                Sichtbar=false;
                DIV_ID.style.visibility="hidden";
            }
            else
            {

```



```

        Sichtbar=true;
        DIV_ID.style.visibility="visible";
        DIV_ID.style.backgroundColor=StandardFarbe;
    }

    // Fade starten
    DIV_ID.filters[0].Play();
}
// -->
</SCRIPT>
</HEAD>
<BODY>
    <BUTTON onclick="Blenden()">
        Aus- und Einblenden
    </BUTTON>
    <BR>
    <BR>
    <DIV ID="DIV_ID"
        STYLE="height:250px;width:250px;background-color:red;
            filter:progid:DXImageTransform.Microsoft.Fade(duration=1);"
        // alles EINE Zeile !!
    >
        <IMG SRC="Bild.gif" .....HEIGHT=250 WIDTH=250>
        // Bild-Dimension muss in die DIV-Dimension reinpassen !
    </DIV>
</BODY>
</HTML>

```

3.11.4.3.5. Beispiel für Aus- und Einblende von Bildern mit Bildwechsel

Variante 1

Es werden zwei DIV an absoluter Position überlagert. Jedes DIV hat einen eigenen Inhalt, hier im Beispiel sein eigenes Bild. Durch das Ein- und Ausblenden wird ein Slide-Show-Effekt der beiden Bilder erreicht.

Nachteil: Pro Bild muss ein eigenes DIV erzeugt werden.

Es ist möglich, nur mit einem DIV zu arbeiten, da der IE den Ersatz den inneren DIV-Bereiches zwischen <DIV> und </DIV> zulässt. Pro Ersatz wird ein Bild eingebunden.

DIV_ID.innerHTML=

Da es sich nur um einen einzigen <DIV> handelt, ist dieser per Layertechnik des IE auch bewegbar.

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        var DIV_ID1_Sichtbar=true;    // DIV_ID1 ist nach dem Dokument-Laden sichtbar
                                    // DIV_ID2 aber nicht

        function BildWechsel()
        {
            // Fade zurücksetzen
            DIV_ID1.filters[0].Apply;
            DIV_ID2.filters[0].Apply;

            if (DIV_ID1_Sichtbar)
            {
                DIV_ID1_Sichtbar = false;
                DIV_ID1.style.visibility="hidden";
                DIV_ID2.style.visibility="visible";
            }
            else
            {
                DIV_ID1_Sichtbar = true;
                DIV_ID1.style.visibility="visible";
                DIV_ID2.style.visibility="hidden";
            }
            // Fade starten
            DIV_ID1.filters[0].Play();
            DIV_ID2.filters[0].Play();
        }
    // -->

```



```

</SCRIPT>
</HEAD>
<BODY>
  <BUTTON onclick="BildWechsel()">
    Bild wechseln mit Aus- und Einblenden
  </BUTTON>
  <BR>
  <BR>
  <DIV
    ID="DIV_ID1"
    STYLE="height:250px;position:absolute;top:50;width:250px;
      background-color:red; visibility:visible
      filter:progid:DXImageTransform.Microsoft.Fade(duration=1);"
    // alles EINE Zeile
  >
    <IMG SRC="DIV_ID1_Bild.gif" ....HEIGHT=250 WIDTH=250>
    // Bild-Dimension muss in die DIV-Dimension reinpassen !
  </DIV>

  <DIV
    ID="DIV_ID2"
    STYLE="height:250px;position:absolute;top:50;width:250px;
      background-color:red; visibility:hidden
      filter:progid:DXImageTransform.Microsoft.Fade(duration=1);"
    // alles EINE Zeile
  >
    <IMG SRC="DIV_ID2_Bild.gif" ....HEIGHT=250 WIDTH=250>
    // Bild-Dimension muss in die DIV-Dimension reinpassen !
  </DIV>
</BODY>
</HTML>

```

Variante 2

Es wird der innere DIV-Bereich zwischen <DIV> und </DIV> ersetzt und damit je ein Bild dargestellt. Durch das Ein- und Ausblenden wird ein Slide-Show-Effekt der Bilder erreicht.

Da es sich nur um einen einzigen <DIV> handelt, ist dieser per Layertechnik des IE auch bewegbar.

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
  var BildUrlFeld=Array
  (
    // hier beliebig viele Bilder eintragen
    "test1.jpg",
    "test2.jpg",
    "test3.jpg"
  );
  var BildUrlFeld_Laenge= BildUrlFeld.length;

  // fuer alle Bilder die identischen Dimensionen !!!
  var BildHoehe=444;
  var BildBreite=640;

  // Zeigerfeld zum Vorladen der Grafiken
  var BildZeigerFeld = Array();

  var Index=1;          // Starten mit test2.jpg, da test1.jpg bereits angezeigt mit Laden des DIV

  function BildWechsel()
  {
    // Fade zurücksetzen
    DIV_ID.filters[0].Apply();

    // Bild im DIV anzeigen
    document.AktuellesBild.src= BildZeigerFeld[Index].src;

    // Fade starten
    DIV_ID.filters[0].Play();

    // nächstes Bild einstellen

```



```

        Index++;

        // Index korrigieren
        if (Index >= BildUrlFeld_Laenge)
        { Index=0; }
    }

    // nachfolgender Code wird mit dem Laden des HEAD-Teiles abgearbeitet

    // Bildobjekte vorladen: allokieren und Zeiger merken per Prototyping
    for (i=0; i<= BildUrlFeld_Laenge; i++)
    {
        // Image-Zeiger bilden als Feldeintrag
        BildZeigerFeld[i] = new Image();

        // Url dem Zeiger zuweisen und Bild somit vorladen
        BildZeigerFeld[i].src= BildUrlFeld[i];
    }

    // DIV erzeugen mit test1.gif als Startbild
    //                               in Dimension aller Bilder
    document.write(
        '<DIV '
        + 'ID="DIV_ID" '
        + 'STYLE="height:' + BildHoehe + 'px;width:' + BildBreite + 'px;'
        + 'filter:progid:DXImageTransform.Microsoft.Fade(duration=1);'
        + '""
        + '>\n'
    );

    document.write(
        '<IMG NAME="AktuellesBild"'
        + 'SRC="' + BildZeigerFeld[0].src + '" '
        + 'WIDTH=' + BildBreite + ' '
        + 'HEIGHT=' + BildHoehe
        + '>\n'
    );

    document.write(
        '</DIV>\n');

    // Button zum Bildwechsel anzeigen
    document.write(
        '<BUTTON onclick="BildWechsel()">'
        + 'Naechstes Bild'
        + '</BUTTON>\n'
    );

    // -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

3.11.4.4. Shadow-Filter für HTML-Grafik-Elemente ab IE 5.5

3.11.4.4.1. Beispiel für Schatten eines Bildes

```

<HTML>
<HEAD>
    <STYLE>
        DIV.filter_name { filter:shadow(color=#000000,direction=120); width:100; color: #FF0000; }
    </STYLE>
</HEAD>
<BODY>
    <DIV CLASS="filter_name" STYLE="width:400;">
        <IMG SRC="bild.gif">
        Das Bild und dieser Text haben einen Schatten
        <BR>
        Die Bild-Dimension muss mit dem des DIV uebereinstimmen !
    </DIV>
</BODY>
</HTML>

```

direction: Schattenrichtung in Grad; von 0 bis 360 Grad



3.11.4.4.2. Beispiel für 3D-Effekt eines Bildes

Dieser Effekt dient dem verschobenen Überlagern ein und desselben Bildes, so dass ein 3D-Effekt mit Schatten erzeugt wird. Die Verschiebung ist damit die Schattendicke bzw. die Effekt-Stärke des 3D-Effektes. Besonders günstig sieht man diesen Effekt, wenn die Grafik einen Schriftzug enthält.

Die Überlagerung erfolgt anhand zweier DIV's, die je eine CSS-Klasse erhalten und ansonsten identisch sind.

```
<HTML>
<HEAD>
  <STYLE>
    DIV.DIV1_filter:{ filter:shadow(color=#FFFF00,direction=300, strength=5);
                      width:100; color: #FF0000;}
                      // alles EINE Zeile
    DIV.DIV2_filter:{ filter:shadow(color=#000000,direction=120, strength=8);
                      width:100; color: #FF0000;}
                      // alles EINE Zeile
  </STYLE>
</HEAD>
<BODY>
  <DIV CLASS="DIV1_filter " STYLE="width:400; position:absolute; top:50px; left:50px;">
    <BLOCKQUOTE>
      <IMG SRC="bild.gif">
      Das Bild und dieser Text haben einen Schatten
      <BR>
      Die Bild-Dimension muss mit dem des DIV uebereinstimmen !
    </BLOCKQUOTE>
  </DIV>
  <DIV CLASS="DIV2_filter " STYLE="width:400; position:absolute; top:50px; left:50px;">
    <BLOCKQUOTE>
      <IMG SRC="bild.gif">
      Das Bild und dieser Text haben einen Schatten
      <BR>
      Die Bild-Dimension muss mit dem des DIV uebereinstimmen !
    </BLOCKQUOTE>
  </DIV>
</BODY>
</HTML>
```

direction: Schattenrichtung in Grad; von 0 bis 360 Grad

strength: Verschiebung in Pixel, also Schattenbreite und somit 3D-Effekt-Stärke

3.11.5.1. Grafik zoomen ab IE 5.5

Natürlich kann die Bildauflösung nicht geändert werden. Je höher der Zoom, um so pixeliger das Bild, welches von sich aus schon hochauflösend sei sollte.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
  var BildUrl="zoom.jpg";
  var BildBreite=250;
  var BildHoehe=372;

  // Bildobjekte vorladen
  // Image-Zeiger
  BildZeiger = new Image();

  // Url dem Zeiger zuweisen und Bild somit vorladen
  BildZeiger.src= BildUrl;

  // .style.zoom
  //          ändert leider die DIV-Dimensionen, so dass die Input-Button VOR dem
  //          zu zommenden Bild liegen müssen und somit ein DIV nicht nötig wird
  //          kann nicht mit einem Zeichenkettenwert belegt werden, sondern muss direkt kodiert
  //          werden !
  document.write(
'<FORM NAME="Formular">'
+ '<INPUT NAME="Button20" TYPE=button VALUE="20%" onclick="Bild_ID.style.zoom=\'20%\''
+ '<INPUT NAME="Button40" TYPE=button VALUE="40%" onclick="Bild_ID.style.zoom=\'40%\''
+ '<INPUT NAME="Button60" TYPE=button VALUE="60%" onclick="Bild_ID.style.zoom=\'60%\''
```



```

+ '<INPUT NAME="Button80" TYPE=button VALUE="80%" onclick="Bild_ID.style.zoom=\'80%\'">'
+ '<INPUT NAME="Button100" TYPE=button VALUE="100%" onclick="Bild_ID.style.zoom=\'100%\'">'
+ '<INPUT NAME="Button120" TYPE=button VALUE="120%" onclick="Bild_ID.style.zoom=\'120%\'">'
+ '<INPUT NAME="Button140" TYPE=button VALUE="140%" onclick="Bild_ID.style.zoom=\'140%\'">'
+ '<INPUT NAME="Button160" TYPE=button VALUE="160%" onclick="Bild_ID.style.zoom=\'160%\'">'
+ '<INPUT NAME="Button180" TYPE=button VALUE="180%" onclick="Bild_ID.style.zoom=\'180%\'">'
+ '<INPUT NAME="Button200" TYPE=button VALUE="200%" onclick="Bild_ID.style.zoom=\'200%\'">'
+ '<INPUT NAME="Button300" TYPE=button VALUE="300%" onclick="Bild_ID.style.zoom=\'300%\'">'
+ '<INPUT NAME="Button400" TYPE=button VALUE="400%" onclick="Bild_ID.style.zoom=\'400%\'">'
+ '</FROM>\n'
+ '<BR>\n'
);

```

3.11.5.2. Video abspielen

3.11.5.2.1. Video abspielen per INPUT TYPE="image" (per IMG analog)

```

<HTML>
<HEAD>
<SCRIPT>
</SCRIPT>
</HEAD>
<BODY>
Das Video 1 startet sofort
<INPUT TYPE="image" ID="TEST1"><BR><BR>
Das Video 2 startet nur bei mouseover
<IMG ID="TEST2">
<SCRIPT>
// funktioniert mit IMG und INPUT image
// Url muss mit dynsrc zugewiesen werden und nicht per src
TEST1.dynsrc='0.wmv';
// TEST1.start='mouseover';
TEST2.dynsrc='1.wmv';
TEST2.start='mouseover';
</SCRIPT>
</BODY>
</HTML> document.write(
    '<IMG ID="Bild_ID" SRC="' + BildZeiger.src + "' '
    + 'HEIGHT=' + BildHoehe
    + 'WIDTH=' + BildBreite
    + '>\n');

document.all.Bild_ID.style.zoom='100%'; // .style ist eine document.all-Eigenschaft
// Bild_ID ist Objekt von document
// entspricht document.Formular.Button100.click();

document.Formular.Button100.focus(); // entspricht document.Formular.elements[4].focus();
// funktioniert leider nur, wenn Dokument nicht aus IE-Cache
// gelesen wird (sonst Übernahme der letzten
// Fokusposition)

// -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

3.11.5.2.2. Video und Sound abspielen per MediaPlayer

Beispiele für Active-X-Control des MediaPlayer zum Abspielen eines Videos oder Sounds:

Achtung: CLASSID wird inzwischen nicht mehr sicher unterstützt (siehe unten Microsoft ändert die Active-X-Eigenschaften

```

<BODY>
<OBJECT ID="ActiveMovie1"
    WIDTH=356
    HEIGHT=328
    CLASSID="CLSID:05589FA1-C356-11CE-BF01-00AA0055595A"
>
    <PARAM NAME="MovieWindowWidth" VALUE="352">
    <PARAM NAME="MovieWindowHeight" VALUE="247">
    <PARAM NAME="FileName" VALUE="0.wmv">
</OBJECT>

```



Das ActiveX-Control umfasst folgende PARAM

```
<PARAM NAME="Appearance" VALUE="0">
<PARAM NAME="AutoStart" VALUE="0">
<PARAM NAME="AllowChangeDisplayMode" VALUE="-1">
<PARAM NAME="AllowHideDisplay" VALUE="0">
<PARAM NAME="AllowHideControls" VALUE="-1">
<PARAM NAME="AutoRewind" VALUE="-1">
<PARAM NAME="Balance" VALUE="0">
<PARAM NAME="CurrentPosition" VALUE="0">
<PARAM NAME="DisplayBackColor" VALUE="0">
<PARAM NAME="DisplayForeColor" VALUE="16777215">
<PARAM NAME="DisplayMode" VALUE="0">
<PARAM NAME="Enabled" VALUE="-1">
<PARAM NAME="EnableContextMenu" VALUE="-1">
<PARAM NAME="EnablePositionControls" VALUE="-1">
<PARAM NAME="EnableSelectionControls" VALUE="0">
<PARAM NAME="EnableTracker" VALUE="-1">
<PARAM NAME="Filename" VALUE="">
<PARAM NAME="FullScreenMode" VALUE="0">
<PARAM NAME="MovieWindowSize" VALUE="0">
<PARAM NAME="PlayCount" VALUE="1">
<PARAM NAME="Rate" VALUE="1">
<PARAM NAME="SelectionStart" VALUE="-1">
<PARAM NAME="SelectionEnd" VALUE="-1">
<PARAM NAME="ShowControls" VALUE="-1">
<PARAM NAME="ShowDisplay" VALUE="-1">
<PARAM NAME="ShowPositionControls" VALUE="0">
<PARAM NAME="ShowTracker" VALUE="-1">
<PARAM NAME="Volume" VALUE="-600">
```

VALUE-Angaben sind nur Beispiele

Bezeichner der Eigenschaften für JScript sind IDENTISCH mit den Werten aus NAME-Attribut von PARAM !
Gross-Kleinschreibung beachten !

AllowChangeDisplayMode	Änderung des Anzeigemodus Zeitanzeige bzw. Framesanzeige durch User boolean true oder false
AllowHideControls	Änderung des Anzeigemodus der Steuerungstasten während Wiedergabe durch User boolean true oder false false (0) für nicht erlauben
AllowHideDisplay	Änderung des Anzeigemodus der Wiedergabe-Anzeige während Wiedergabe durch User boolean true oder false false (0) für nicht erlauben
AutoRewind	Automatisches Rückspulen nach Stop boolean true oder false false (0) für nicht rückspulen
AutoStart	Autostart der Wiedergabe ein oder aus boolean true oder false false (0) für nicht automatische Wiedergabe: siehe Aktionen zum Player
Balance	numerisch 0 für zentriert default -1000 ganz links +1000 ganz rechts
CurrentPosition	unklar
DisplayBackColor	Hintergrundfarbe der Control-Elemente numerischer Farbwert, kein Hexastring 0 Default für Black 16777215 für Weiss
DisplayForeColor	Hintergrundfarbe der Control-Elemente



	numerischer Farbwert, kein Hexastring 0 Default für Black 16777215 für Weiss
DisplayMode	Anzeigeart der Position 0 für Time 1 für Frames
Enabled	Control-Anzeige ein, aus boolean true oder false false (0) für nicht erlauben
EnableContextMenu	Rechte-Maus-Kontextmenü ein, aus boolean true oder false false (0) für nicht erlauben
EnablePositionControls	Positionierungs-Button im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
EnableSelectionControls	Selektions-Button im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
EnableTracker	Tracker-Balken im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
FileName	Mediendatei mit Pfad oder Url
FullScreenMode	Vollbildschirm ein, aus boolean true oder false false (0) für kein Vollbildmodus
MovieWindowSize	unklar, Grösse des Anzeigefensters
PlayCount	Anzahl der Wiedergaben der Mediendatei ab 1
Rate	Wiedergabefaktor numerisch 1 für 1-fache Geschwindigkeit 10 für 20-fache Geschwindigkeit
SelectionEnd	Bis-Position, bis zu der widergegeben werden soll in Sekunden bezüglich Anfang der Mediendatei (0)
SelectionStart	Von-Position, bis zu der widergegeben werden soll in Sekunden bezüglich Anfang der Mediendatei (0)
ShowControls	Start Stop buttons and tracker controls sichtbar ein, aus boolean true oder false false (0) für aus
ShowDisplay	Status-Anzeige-Panel sichtbar ein, aus boolean true oder false false (0) für aus
ShowPositionControls	Previous, Rewind, Forward and Next buttons sichtbar ein, aus boolean true oder false false (0) für aus
ShowSelectionControls	Selektions-Button im Control-Panel sichtbar ein, aus boolean true oder false false (0) für aus
ShowTracker	Trackerbar im Controlpanel sichtbar ein, aus: nur lesen boolean true oder false false (0) für aus



uiMode	Sichtbarkeit anstelle STYLE-visible "invisible" für unsichtbar
Volume	Laustärke als Basis dient die aktuelle Windows-Lautstärke integer 0 für maximale, also aktuelle Windows-Lautstärke (höher geht nicht !!!) -10000 für total still

Den ganzen Mediaplayer unsichtbar machen: per STYLE-Attribut im OBJECT-Tag

document.ID-Bezeichner.readyState (Mediaplayer 6.4, funktioniert aber auch mit höheren Versionen)

.readyState hat folgende Werte

0	The FileName property has not been initialized.
1	Windows Media Player control is asynchronously loading a file.
3	Windows Media Player control loaded a file, and downloaded enough data to play the file, but has not yet received all data.
4	All data has been downloaded.

Methoden:

.IsSoundCardEnabled()	
	liefert boolean true Soundkarte erkannt false keine Soundkarte erkannt
.Run();	Erzeugt und startet Instanz des Mediaplayers und ermöglicht danach weitere Aktionen wie .Play etc.
.Mute	ohne ()
	lesen, schreiben true Player volume is muted. false (Default) Player volume is not muted.
.Play	ohne ()
	Wiedergabe starten
.AutoStart	ohne ()
	true (Default) Clip automatically starts. false Disables automatic start.
.Stop	ohne ()
.Pause	ohne ()

Audio wiedergeben

```
<HEAD>
<SCRIPT>
var SoundStarten_TimeoutID=0;

function SoundStarten()
{ // auf Soundkarte prüfen
  if (ID_PlayerObjekt.IsSoundCardEnabled())
  { // Soundkarte vorhanden
    // Musik geladen und abspielfähig ?
    if (ID_PlayerObjekt.readyState == 4)
    { // geladen, also Playerinstanz starten
      ID_PlayerObjekt.Run();
      // Musik wiedergeben
      ID_PlayerObjekt.Play;
      alert('Musik ist gestartet !');
    }
  }
}
```



```

    }
    else{ SoundStarten_TimeoutID=window.setTimeout('SoundStarten()', 100);}
  }
  else{alert('Keine Soundkarte gefunden !');}
}
</SCRIPT>
</HEAD>
<BODY>
<OBJECT ID="ID_PlayerObjekt" CLASSID="CLSID:05589FA1-C356-11CE-BF01-00AA0055595A">
  <PARAM NAME="FileName" VALUE="a0.mp3">
  <PARAM NAME="AutoStart" VALUE="0">      <!-- kein Autoplay -->
</OBJECT>
<SCRIPT>
if (ID_PlayerObjekt != null){SoundStarten();}
</SCRIPT>
</BODY>
</HTML>

```

Microsoft ändert fortlaufend die Active-X-Eigenschaften von Windows und somit auch des Internet Explorers

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.

Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.

Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Lizensierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks

Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.

Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.

Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2). Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.

Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

Hinweis: Man suche doch mal im Internet nach einem kostenlosen HTTP-Server vom Microsoft, um IE-Seite testen zu können, die JScript nutzen (inklusive Debugger). Denn sollte kein kostenloses Angebot findbar sein, kommen die Kosten von Entwicklungssoftware zum IE hinzu. Ein Blick auf Konkurrenz-HTTP-Server klärt die Sachlage: Apache-HTTP-Server ist kostenlos, allerdings nicht einfach einzurichten (Hinweis: Der HTTP-Server sollte virtuelle Hosts einrichten können und korrekt mit der Firewall des Users zusammenarbeiten können).

Abänderungen wegen Sicherheitspatches der jeweiligen Windows-Versionen

Abschaltungen von Active-X-Controls erfolgen auch im Rahmen der Sicherheitspatches zu Windows-Versionen.

Es ist auch möglich, dass wegen Sicherheitslücken abgeschaltet wird und somit Komponenten einer Webseite je nach Windowsversion nicht mehr laufen.

Im Rahmen der Sicherheitspatches ist es Microsoft sogar gelungen, Webseiten, die den MS-Encoder zur Komprimierung von

HTML- und JScript-Code nutzen, schlagartig unnutzbar zu machen: Ein Bug in einem Patch zu Windows XP - Q918899

Das Patch verursacht IE-Browser-Absturz bei per MS ScriptEncoder gepacktem JScript unter SP1 und 2 wenn HTTP 1.1 mit Kompression genutzt wird z.B. bei onclick-Handler auf IMG klick ins Fenster per aktivem Popup

Der Absturz ist "read" -Fehler von immer ein und derselben Speicherstelle.

User, die dieses Patch installiert haben, können ab sofort keine IE-Seiten mit codiertem Script mehr ansehen.

Microsoft stellt Abhilfe nach geraumer Zeit zur Verfügung, jedoch spezifisch nach Windows XP-Version:

Patch Q918899 für

Windows XP SP1
SP2

Download für jedermann bereitgestellt

nur auf kostenpflichtige telefonische Anfrage des Users per Downloadlink bereitgestellt, da

Microsoft explizit die User registriert haben will, bei denen das

Patchproblem auftritt (User muss sich Telefonnummer besorgen)

Solange also das Patch zum fehlerhaften Patch vom User nicht installiert wird,

z.B. weil der User keine Ahnung hat, dass und wo er sich die Telefonnummer

von Microsoft besorgen muss bzw. zu besorgen hat, wird der User

IE-Seiten mit komprimierten Code dauerhaft nicht nutzen können.



(Microsoft-Support ist z.T. nur in Englisch).

Abänderungen wegen Browser-Inkompatibilität

Popupblocker-Fehler

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()

Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.

Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.

Der Fehler tritt nicht auf, wenn ein Fenster per window.open() erzeugt wurde.

Bedingung:

Scriptfehleranzeige ist erlaubt im IE 7

Popupblocker ist im IE abgeschaltet

ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 window.popup per .show() erzeugt.

ein weiteres Fenster (Register) z.B. leere Seite (about:blank)

beide (Register) liegen in einer gemeinsamen IE-Instanz

Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per .show() erzeugt,

bricht der Browser das Dokument mit .show() ab (Scriptfehler).

Der Popupblocker für die leere Seite verursacht den Programmfehler im Dokument mit .show(). Es wird folgende

Meldung angezeigt (in der Informationsleiste):

'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'

Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:

Der Popupblocker hat ein Populfenster geblockt. Sie können den Popupblocker deaktivieren

oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.

Die Realität zur obigen Meldung ist völlig anders:

Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter

Popupblocker einschalten

weitere Informationen

jedoch keine Möglichkeit wie laut Bedeutung

Damit gilt: Der abgeschaltete Popupblocker ist in Wirklichkeit aktiv.

Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster einer Windowsanwendung z.B. einer anderen IE-Instanz)

Der Popupblocker des IE bemeckert aber NUR Webseite, die das Popup erzeugt.

Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popupblocker verwaltet wird.

Der Popupblocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen Webseiten (die nicht das Popup erzeugt haben).

Der Popupblocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.

Der Popupblockerfehler verändert die Eventverwaltung:

Es werden u.a. ignoriert

onfocus

onblur

onfocusin

onfocusout

und viele andere, so dass trotz Events z.B. des Body der Popupblockerfehler entsteht.

// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell

window.focus();

window.document.focus();

if(document.body!=null)

{if(document.body.style!='hidden') // wenn hidden so focus() nicht möglich (Scriptfehler erzeugt)

{document.body.focus();}

}

// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt

popupzeiger.show(...);

Hinweis: Der Populfehler ist so elementar, dass die vielen Beta-Testphasen des IE mehr als fragwürdig erscheinen, wie die Angabe von Microsoft, dass Code neu programmiert wurde, um den IE sicherer zu machen.

focus-Methode beim IE 7

windows.focus() document.focus() und body.focus() funktionieren NICHT

zwischen Register in einem IE-Fenster

zwischen Fensters z.B. in Taskleiste

Hinweis:

.focus() setzt Element aktiv, gibt dem Element den Focus und feuert dann onfocus

.setActive() ist Teilmenge von .focus(): nur das aktiv setzen

funktioniert nicht mit allen Elementen, mit denen .focus() funktioniert



animierte Gif (mit Timer)

Animierte Gifs (mit Timer), die unter IE 6 korrekt laufen, müssen unter IE 7 im Timer nicht mehr laufen:
z.B. garnicht mehr sichtbar, oder Timer nicht verwendet.

Dann müssen animierte Gif-Bilder nach IE-Version bereitgestellt werden.

Abänderungen wegen Rechtstreitigkeiten von Microsoft mit Fremdanbietern

Ein sehr bekanntes Beispiel ist die nachträglich eingeführte Einschränkung von Active-X-Controls wegen Patentwahrung durch Microsoft, wobei für den JScript-Programmierer massive Änderungen eintreten.

Wegen Patentwahrung hat Microsoft ein zunächst freiwilliges Patch herausgegeben, dass bei ActiveX-Control per APPLLET, EMBED oder OBJECT, die auf dem Bildschirm rendern (mit oder ohne Userschnittstelle), dafür sorgt, dass bei mouseover über das Control eine Sprechblase erscheint, die darauf hinweist, dass das Objekt als ActiveX-Control klickbar ist.

Diese Sprechblase erscheint auch, wenn das Control keine Userschnittstelle hat, also diese gar nicht klickbar ist.

Es wurde das Eventmodell gleichzeitig geändert:

Es werden alle Events solange unterdrückt, bis der User die Sprechblase geklickt hat.

Das Klicken muss auf das Objekt im Sprechblasenrahmen erfolgen, der so groß ist, wie die Dimension, in der gerendert wurde.

Es muss also ERST per Mausklick das Control aktiviert werden, ehe das Control klickbar und damit die Eventsteuerung aktiviert ist.

Ein Control, dass programmtechnisch zwar was rendert, aber ansonsten ohne sichtbare programmtechnisch startet, muss ebenfalls geklickt werden, obwohl es bereits läuft und es nichts zu klicken gäbe (wenn keine Eventsteuerung eingebaut wurde).

Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.

Die Eventauslösung per nicht-objekteigenen Eventhandler, der für das Objekt per fireEvent() ein Event auslöst, ist solange blockiert, bis der User die Sprechblase geklickt hat.

style.visibility='hidden' wird ignoriert

Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert: Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.

Diese Sprechblase erscheint nur DANN NICHT, wenn die Userschnittstelle mit Breite == Höhe == 0 gerendert wird. Sollte die Userschnittstelle in einem Container liegen, z.B. DIV, dann wird der Container, wenn er in der Dimension kleiner ist, also die Userschnittstelle, angepasst. Daher muss der Container ebenfalls mit Breite == Höhe == 0 gerendert werden. Wegen Dimensionierung auf 0 sollte style.visibility="hidden" sein. Im Falle eines Containers reicht es, den style des Containers zu ändern, da visibility normalerweise vererbt wird an Kinder, also auch an das Control.

Abänderung wegen Abschaltungen

DirectX ist wegen Abschaltung von Active-X--Controls nicht mehr abwärtskompatibel:

Z.B. wurde bei Win XP SP2 Direct Animation aus DirectX schlagartig durch Abschaltung von Bibliotheken dezimiert, die es bei Win XP SP1 aber noch gibt.

Hier ein Beispiel aus dem Jahr 2004: Abschaltungen von Active-X-Controls

ActiveX-Controls und Unterstützung/Verbot 20041215

erlaubt sind noch

Tabular Data-Steuerelement{333C7BC4-460F-11D0-BC04-0080C7055A83} Das TDC (Tabular Data-Steuerelement) ermöglicht die Weiterverarbeitung von Daten, die nur im Textformatvorliegen, beispielsweise durch Darstellung in einer Tabelle oder Sortierung. Weitere Informationen:•

http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp(http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp)

Microsoft Agent Control - Version 2.0 {D45FD31B-5C6E-11D1-9EC1-00C04FD7081F} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb



eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Weitere Informationen:•

<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>(<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Microsoft MSChat-Steuerelement-Objekt 2.0 - 2.5 {D6526FE0-E651-11CF-99CB-00C04FD64497}

Dieses Steuerelement wird von Webautoren verwendet, um text- und graphisch basierte Chatgemeinden für Echtzeitkonversationen im Web zu erstellen.

Microsoft ActiveX Upload-Steuerelement, Version 1.5 {886e7bf0-c867-11cf-b1ae-00aa00a3f2c3} Dieses Steuerelement kann auf vielerlei Art genutzt werden, um auf einfache Weise Webinhalte via Drag and Drop zu veröffentlichen. Weitere Informationen:• 230298 (<http://support.microsoft.com/kb/230298/DE/>) - Posting Acceptor Release Notes

• http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp
(http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp)

verboten sind

Datenbindung RDS {BD96C556-65A3-11D0-983A-00C04FC29E36} {BD96C556-65A3-11D0-983A-00C04FC29E33} Die RDS (Remote Data Service) Steuerelemente ermöglichen dem Browser, client-basierte SQL Abfragen an einen Webserver zu stellen. Inzwischen wurde RDS jedoch durch neuere Standards wie SOAP abgelöst, von einer weiteren Verwendung von RDS wird daher abgeraten. Weitere Informationen:• 184375 (<http://support.microsoft.com/kb/184375/DE/>) - Sicherheitsaspekte bei RDS 1.5, IIS 3.0 oder 4.0 und ODBC

<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>
(<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>)

http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp
(http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp)

XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer {550dda30-0541-11d2-9ca9-0060b0ec3d39} {CFC399AF-D876-11d0-9C10-00C04FC99C8E} {e54941b2-7756-11d1-bc2a-00c04fb925f3} {7108ECB4-AFDC-11D1-ADC1-00805FC752D8} XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer ermöglichen die Verarbeitung von XML Daten, etwa die Bindung von HTML Elementen an einen XML Datensatz, oder das Einlesen, Manipulieren, und Zurückschreiben von XML Daten.

Die Steuerelemente DOMDocument und XMLIslandPeer bzw. die dazugehörigen ClassIDs sind nicht mehr aktuell, so dass von einer generellen Freigabe dieser Steuerelementgruppe abgeraten wird. Weitere Informationen:• http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp(http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp)

Internet Explorer

Active Setup / IE Active Setup-Steuerelement {F72A7B0E-0DD8-11D1-BD6E-00AA00B92AF1} Dieses Steuerelement enthält die in Microsoft Security Bulletin MS99-037 beschriebene Sicherheitsanfälligkeit. Um eine weitere Ausführung zu verhindern wurde im Rahmen dieses Security Bulletins ein Kill-Bit gesetzt, so dass selbst bei einer Freigabe dieses Controls eine Ausführung blockiert wird. Weitere Informationen:•

<http://www.microsoft.com/technet/security/bulletin/ms99-037.mspx>
(<http://www.microsoft.com/technet/security/bulletin/ms99-037.mspx>)

<http://www.microsoft.com/technet/security/bulletin/fq99-037.mspx>
(<http://www.microsoft.com/technet/security/bulletin/fq99-037.mspx>)



240797 (<http://support.microsoft.com/kb/240797/DE/>) - So verhindern Sie die Ausführung von ActiveX-Steuerelementen in Internet Explorer

Media Player / Active Movie Runtime {A4001DE0-7075-11d0-89AB-00A0C9054129} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Runtime Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / ActiveMovie-Steuerelement{05589FA1-C356-11CE-BF01-00AA0055595A} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Microsoft NetShow Player {2179C5D3-EBFF-11CF-B6FD-00AA00B4E220} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das NetShow Player Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Windows Media Player {22D6F312-B0F6-11D0-94AB-0080C74C7E95} Dies ist das Steuerelement für Windows Media Player version 6.4 und war Installationsbestandteil bis einschließlich Windows Media Player Version 8. Ab Windows Media Player 9 wurde diese ClassID durch die neue ClassID {6BF52A52-394A-11D3-B153-00C04F79FAA6} abgelöst, deren Verwendung stattdessen empfohlen wird. Ab Windows Media Player Version 9 wird ferner die alte ClassID anhand eines Wrappers automatisch auf die neue ClassID umgeleitet. Die ClassID für Windows Media Player Version 9 ist jedoch nicht in der Liste der vom Administrator genehmigten Steuerelemente enthalten, und muss bei Bedarf manuell hinzugefügt werden.

Animierte Schaltflächen{0482B100-739C-11CF-A3A9-00A0C9034920} Dieses Steuerelement erlaubte in frühen Versionen des Internet Explorer die Verwendung animierter Schaltflächen auf Webseiten. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von der Freigabe des Steuerelements wird daher abgeraten.

IE Label-Steuerelement

{99B42120-6EC7-11CF-A6C7-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 auch kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

IE Menu-Steuerelement {74701400-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ermöglicht die Handhabung von Menüstrukturen in Webseiten, wird jedoch nicht mehr unterstützt und dürfte nur noch selten Verwendung finden. Von einer Freigabe des Steuerelements wird daher abgeraten.

IE Preloader-Steuerelement {16E349E0-702C-11CF-A3A9-00A0C9034920} Dieses Steuerelement ermöglichte das Vorladen von Webseiten, ist jedoch inzwischen nicht mehr aktuell, wird nicht mehr unterstützt und dürfte nicht mehr im Einsatz sein. Aufgrund einer potentiellen Sicherheitsanfälligkeit in diesem Steuerelement wird von einer Freigabe abgeraten. Weitere Informationen: • 231452 (<http://support.microsoft.com/kb/231452/DE/>) - Update Available for "Legacy ActiveX Control" Issue



IE Timer-Steuerelement {59CCB4A0-727D-11CF-AC36-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

MCSiMenü {275E2FE0-7486-11D0-89D6-00A0C90C9B67} Dieses Steuerelement dient der Anpassung von Popupmenüs, ist jedoch nicht mehr aktuell und wurde nach Windows 98 nicht mehr ausgeliefert. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten.

Popupmenüobjekt {7823A620-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

Microsoft Agent Control - Version 1.5 {F5BE8BD2-7DE6-11D0-91FE-00C04FD701A5} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Diese Version des Steuerelements ist jedoch nicht mehr aktuell und wird nicht mehr unterstützt. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • <http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm> (<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden. Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Wenn der IE 6.x aktive Inhalte blockiert, wird NOSCRIPT-Tag aktiviert, Ausnahme: Frameset

FRAMESET ist ein aktiver Inhalt:

Da der Frameset anstelle <BODY> kodiert sein muss, gilt:

Alle Tags, die für BODY zulässig sind, werden ignoriert, auch NOSCRIPT.

Wird neben Frameset noch BODY kodiert, so wird Frameset ignoriert.

Die Freigabe der Scriptblockierung erzeugt Ausführung aller Script-Teile inklusive der Eventauslösungen

Bsp.: Folgendes funktioniert vom Dokument, das window.open() hat im geöffneten Dokument (Quelltext im Dokument das window.open() verwendet):

```
function Y_unload(X00){X85[X00].close();}

var X85=new Array();var X86=new Array();

X85[0]=window.open(...);

var X87='parent.Y_unload(0)'; X86[0]=new Function("X87);

X85[0].document.body.onunload=X86[0];
```

Wird die Scriptblockierung im geöffneten Fenster abgeschaltet, so wird das Fenster geschlossen, weil onunload ausgelöst wird.

Achtung: document.body.onunload funktioniert ev. nicht mehr wenn z.B. mit attachEvent() aktiviert wurde

Folgende Metatags sind für den IE 6.x aktiver Inhalt:



```
<META HTTP-EQUIV="imagetoolbar" CONTENT="no">
    unterdrückt NICHT IE-Kontextmenü rechte Maus auf Bild
```

```
<META HTTP-EQUIV="site-enter" CONTENT="revealtrans(duration=0.3, transition=12) ">
<META HTTP-EQUIV="site-exit" CONTENT="revealtrans(duration=0.3, transition=12) ">
```

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.

Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein, innerhalb dessen dann die neuen HTML-Elemente erzeugt werden.

3.11.6. PopUp-Fenster ab IE 5.5 per createPopup()

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    // HTML-Inhalt des PopUp-Fensters
    var PopUpFenster_HTMLString =
        'Das ist eine Sprechblase!<BR>Bitte ausserhalb der Sprechblase klicken,<BR>um sie zu
        schliessen !';

    // PopUp-Fenster-Start-Pixel-Koordinaten gegenüber den Bildschirm-Dimensionen
    //      (nicht Browserfenster etc.)
    var PopUpFenster_LinkeObereEcke_X =0;           // Spalte
    var PopUpFenster_LinkeObereEcke_Y =0;           // Zeile

    // PopUp-Fenster-Dimension in Pixel
    var PopUpFenster_Breite      =250;              // Spalte
    var PopUpFenster_Hoehe      =60;                // Zeile

    var PopUpFenster_HintergrundFarbe="yellow";

    // Es werden keine Events verarbeitet !!
    //      also bleibt z.B. <INPUT TYPE=button VALUE="PopUpFenster verstecken" onclick=".....">
    //      wirkungslos

    function PopUpInstanzErzeugen_PopUpAnzeigen_PoUpInstanzLoeschen()
    {
        //      Instanz erzeugen
        var PopUpFenster = window.createPopup();

        //      den Inhalt zuweisen
        PopUpFenster.document.body.innerHTML=PopUpFenster_HTMLString;

        //      Das Aussehen gestalten mit der Eigenschaft .style
        //      z.B. Hintergrundfarbe
        PopUpFenster.document.body.style.backgroundColor=PopUpFenster_HintergrundFarbe;

        //      anzeigen, aber nicht schliessen
        PopUpFenster.show(PopUpFenster_LinkeObereEcke_X,
                          PopUpFenster_LinkeObereEcke_Y,
                          PopUpFenster_Breite,
                          PopUpFenster_Hoehe,
                          document.body);

        //      Instanz löschen
        PopUpFenster=null;
    }
    PopUpInstanzErzeugen_PopUpAnzeigen_PoUpInstanzLoeschen();

    // Jeder Mausklick außerhalb der PopUp-Fensters schliesst es; Tastaturereignisse jedoch nicht
    // Jedes andere Fenster, das danach geöffnet wird (auch per alert();), schliesst das PopUp-Fenster
    //      automatisch
    // -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
```



</HTML>

3.11.7. Stackoperationen ab IE 5.5 für das Array-Objekt

Push-Methode Browsern <> ab IE5.5

```
function push()
{
    var sub = this.length;
    for (var i = 0; i < push.arguments.length; ++i)
    {
        this[sub] = push.arguments[i];
        sub++;
    }
}

var names = new Array("Tom", "Mark", "Bart", "John");
names.push = push; oder Array.prototype.push = push;
```

Push-Operation ab IE 5.5:

```
var names = new Array("Tom", "Mark", "Bart", "John");
names.push("Jim", "Richard", "Tim");
alert(names);
```

Pop-Methode Browsern <> ab IE5.5

```
function pop()
{
    var lastElement = this[this.length - 1];
    this.length--;
    return lastElement;
}

Array.prototype.pop = pop;
```

Pop-Methode ab IE5.5

```
var names = new Array("Tom", "Mark", "Bart", "Jonn");
var last = names.pop();
alert("Last = " + last + " Other = " + names);
```

3.11.8. Auf ActiveX prüfen beim IE

```
<OBJECT ID="ActiveXObjekt"
    WIDTH="1"
    HEIGHT="1"
    CLASSID="CLSID:978C9E23-D4B0-11CE-BF2D-00AA003F40D0"
>
    <PARAM NAME="Caption"
        VALUE="ActiveX">
</OBJECT>
```

```
if (document. ActiveXObjekt
{ //ActiveX ist aktiv}
```

3.11.9. Linkrahmen unterdrücken beim IE

Immer, wenn Sie auf einen Link oder einen Button klicken, zieht der Internet Explorer einen kleinen, gestrichelten Rahmen darum, der erst beim Klick auf ein anderes Objekt wieder verschwindet. Dieser Rahmen ist vor allem bei Imagemaps störend. Der HTML-Code für das Objekt (z.B einen Link) ist onclick="this.blur()" zu erweitern.

Bsp: Link

3.11.10. Tooltip beim IE

durch Kodierung des Attributes TITLE ="freier Text "

Bsp:<hr align="center" width="300" size="5" title="Hier kann ein beliebiger Text stehen">

Der Text darf bis zu 1024 Zeichen lang sein

	 für Texteinzug

 für Zeilenumbruch



3.11.11. Webseite mit eigenem Icon ab IE 5.x

in die Webseite einfügen

```
<LINK REL="SHORTCUT ICON"
      HREF="name.ico"
>
```

name.ico: name ist beliebig, auch mit Pfad

ICO-Datei: 32x32 Pixel mit 16 Farben
oder 16x16 Pixel mit 256 Farben
ist BMP-Datei, die zu ICO-Datei konvertiert werden muss
(kann nicht jedes Grafik-Programm, aber
Microsoft bietet dafür IconPro an)

3.11.12. Hintergrundbild beim IE als Wasserzeichen

```
if (document.all||document.getElementById)
{document.body.style.background="url('notebook.jpg') white center no-repeat fixed";}
```

3.11.13. Uhr mit IE-Filter

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
//          24-Stunden Digitaluhr mit Spiegelbild nuf für IE
//
// *****
//
//          vom Programmierer zu belegende Variablen
//
// *****
var Uhr_Frequenz=960;                                // Bitte an die echte Genauigkeit anpassen auf Sekundentakt

var Uhr_Breite                                     = 10;                                // in Pixel
var Uhr_Hoehe                                     = 10;                                // in Pixel
var Uhr_AbstandVomFensterRandOben                = 10;                                // in Pixel
var Uhr_AbstandVomFensterRandLinks               = 50;                                // in Pixel

var Uhr_UngespiegeltFontFamily                    = "Comic Sans Ms";
var Uhr_UngespiegeltFontSize                      = 14;                                // in pt
var Uhr_UngespiegeltFontStyle                    = "italic";
var Uhr_UngespiegeltFontWeight                   = "bold";
var Uhr_UngespiegeltFontFarbe                    = "#00008D";

var Uhr_GespiegeltFontFamily                      = "Comic Sans Ms";
var Uhr_GespiegeltFontSize                        = 14;                                // in pt
var Uhr_GespiegeltFontStyle                      = "italic";
var Uhr_GespiegeltFontWeight                     = "bold";
var Uhr_GespiegeltFontFarbe                      = "#00008D";
var Uhr_GespiegeltDeckungskraft                  = 11;                                // je höher um so stärker die Farbe denkend
                                                    // je kleiner um so blasser die Farbe

var Uhr_GespiegeltAbstandVonUngespiegelt         = Uhr_UngespiegeltFontSize + 6;

var LeeresBildUrl="1x1tran.gif"                    // muss 1x1 Pixel sein, ist transparent

// *****
//
//          nachfolgenden Code nicht verändern
//
// *****
// Dokument mit leeren Bild instanzieren
//          Instanzierung erfolgt normalerweise ERST mit Lesen des BOD>-Tags, aber
//          das Dokument muss spätestens zur Ermittlung der Fensterdimensionen
//          per Javascript instanziiert sein, also auch in Javascript.
//          ACHTUNG: Der GESAMTE BODY-Teil des Dokumentes wird damit ignoriert und MUSS leer bleiben !!!

document.write('<IMG SRC="'+ LeeresBildUrl + '" HEIGHT=1 WIDTH=1>');

//          Prüfung auf Browsertyp
//          Dieser Quellcode muss VOR allen anderen Routinen codiert sein, damit zuerst abgearbeitet
```



```

var ns = document.layers ? true : false;
var ie = document.all ? true : false;

var TimeoutID;

function UhrAnzeigen()
{
    var DatumObjekt= new Date();
    var Stunde = DatumObjekt.getHours();
    var Minute = DatumObjekt.getMinutes();
    var Sekunde = DatumObjekt.getSeconds();

    if (Stunde < 10 )
    {Stunde="0" + Stunde;}

    if (Minute <10 )
    {Minute="0" + Minute;}

    if (Sekunde <10 )
    {Sekunde="0" + Sekunde;}

    DIV_ID_Uhr_ungespiegelt.innerHTML= Stunde + ":" + Minute + ":" +Sekunde
    DIV_ID_Uhr_gespiegelt.innerHTML = Stunde + ":" + Minute + ":" +Sekunde

    TimeoutID=setTimeout('UhrAnzeigen()',Uhr_Frequenz);
}

if (ie)
{
    document.write(
        '<DIV ID="DIV_ID_Uhr_ungespiegelt"'
        + ' STYLE="position:absolute;'
        + 'width:' + Uhr_Breite + 'px;'
        + 'height:' + Uhr_Hoehe + 'px;'
        + 'top:' + Uhr_AbstandVomFensterRandOben + 'px;'
        + 'left:' + Uhr_AbstandVomFensterRandLinks + 'px;'
        + 'font-family:' + Uhr_UngespiegeltFontFamily + ';'
        + 'font-size:' + Uhr_UngespiegeltFontSize + 'pt;'
        + 'font-style:' + Uhr_UngespiegeltFontStyle + ';'
        + 'font-weight:' + Uhr_UngespiegeltFontWeigth + ';'
        + 'color:' + Uhr_UngespiegeltFontFarbe + ';'
        + '""
        + '>'
        + '</DIV>'
    );

    document.write(
        '<DIV ID="DIV_ID_Uhr_gespiegelt"'
        + ' STYLE="position:absolute;'
        + 'width:' + Uhr_Breite + 'px;'
        + 'height:' + Uhr_Hoehe + 'px;'
        + 'top:'
        + (Uhr_AbstandVomFensterRandOben + Uhr_GespiegeltAbstandVonUngespiegelt)
        + 'px;'
        + 'left:' + Uhr_AbstandVomFensterRandLinks + 'px;'
        + 'font-family:' + Uhr_GespiegeltFontFamily + ';'
        + 'font-size:' + Uhr_GespiegeltFontSize + 'pt;'
        + 'font-style:' + Uhr_GespiegeltFontStyle + ';'
        + 'font-weight:' + Uhr_GespiegeltFontWeigth + ';'
        + 'color:' + Uhr_GespiegeltFontFarbe + ';'
        + 'filter:flipv() alpha(opacity=' + Uhr_GespiegeltDeckungskraft + ');'
        + '""
        + '>'
        + '</DIV>'
    );

    UhrAnzeigen();
}
//-->
</SCRIPT>
</HEAD>

```



```
<BODY>
</BODY>
</HTML>
```

3.12. Verweilzeit des Users in Sekunden ermitteln

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
      <!--
        var start_zeit;          // muss global sein

        function start()
        {start_zeit = new Date();}

        function ende()
        {
          var ende_zeit = new Date();
          var wert=ende_zeit.getTime() - start_zeit.getTime();
          wert = wert /1000;      // in Sekunden umrechnen
          alert("Verweilzeit in Sekunden: " + Math.floor(wert).toString());
        }
      <!-->
    </SCRIPT>
  </HEAD>

  <BODY ... onLoad="start()" onUnload="ende()">
</BODY>
</HTML>
```

3.13. History anwenden

3.13.1. um 1 Seite zurück gehen

```
<A HREF="javascript:window.history.back();">vorherige Seite</A>
<A HREF="javascript:history.go(-1)">zur&uuml;ck...</A>
```

3.13.2. um 1 Seite vorwärts gehen

```
<A HREF="javascript:window.history.forward();">naechste Seite</A>
<A HREF="javascript:history.go(+1)">zur&uuml;ck...</A>
```

3.13.3. um mehrere Seiten wahlweise vorwärts bzw. rückwärts gehen

```
</HEAD>
  <BODY>
    <FORM name="formular">
      <P>Geben Sie die Anzahl der Schritte ein (>0 fuer vorwaerts und <0 fuer rueckwaerts):
      <INPUT TYPE="text"
        NAME="schritt_anzahl"
        SIZE="5">

      </P>
      <INPUT TYPE="button"
        VALUE="gehen"
        NAME="button"
        onClick="javascript:window.history.go(formular.schritt_anzahl.value);">

      </P>
    </FORM>
  </BODY>
</HTML>
```

3.13.4. History-Eintrag ersetzen

Bsp: in test.html kodieren window.location.replace("test.html");

3.14. Alle Textmarken (Anker) eines Dokumentes für deren Anwahl in einem Extra-Fenster anzeigen

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
      <!--
        function textmarken_fenster_anzeigen()
        {
          var textmarken_fenster = open(
```



```

        "",
        "Textmarken-Fenster",
        "toolbar=0,location=0,scrollbars=0,menu=0,"
        + "dependent,resizable,status,width=150,height=300");

with(textmarken_fenster.document)
{
    // HTML-Fenster formatieren
    window.name = "logischer_fenster_name"
    open("text/html")
    writeln("<HTML><HEAD>")
    writeln(    "<TITLE>Steuerung f&uuml;r &quot;"
        + document.title
        + "&quot;</TITLE>"
    )
    writeln(    "<BASE HREF=\"\"
        + location.href
        + "\" TARGET=\" logischer_fenster_name \">"
    )
    writeln("</HEAD><BODY>")

    // Textmarken (Anker im Fenster anzeigen)
    for(var i = 0; i < document.anchors.length; i++)
    {
        writeln(    "<P><A HREF=\"#"
            + document.anchors[i].name
            + "\">"
            + document.anchors[i].name
            + "</A></P>"
        );

        writeln("</BODY></HTML>");
        close();
    }
    textmarken_fenster.document.focus();
}

// -->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
<!--
    document.write(    "<P>"
        + "<A HREF=\"javascript: textmarken_fenster_anzeigen ()\">"
        + "Textmarken-Fenster anzeigen</A></P>"
    );

// -->
</BODY>
</HTML>

```

Hinweis: Um \" zu vermeiden, kann eine Zeichenkette auch in '...' gesetzt werden.

Bsp: ''

3.15. *Inhalt des aktuellen Dokumentes als zusätzlichen Frame im Fenster darstellen*

Achtung: Das aktuelle Fenster kann nicht kleiner 100x100 Pixel sein !

```

<HTML>
</HEAD>
<SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
<!--
    function frame_erzeugen()
    {
        var breite = 0;
        var hoehe = 0;

        if(self.document.body)
        {
            breite = self.document.body.clientWidth;

```



```

        hoehe = self.document.body.clientHeight;
    }
    else
    {
        breite = self.innerWidth;
        hoehe = self.innerHeight;
    }

    if((breite >= 100) || (hoehe >= 100))
    {
        with(document)
        {
            var titel = title;
            open("text/html");
            writeln("<HTML><HEAD><TITLE>" + titel
                + "</TITLE></HEAD>");
            writeln("<FRAMESET " + ((w > h) ? "COLS" : "ROWS")
                + "=\\"*,*\>");
            writeln("<FRAME SRC=\" + location.href + "\">");
            writeln("<FRAME SRC=\" + location.href + "\">");
            writeln("</FRAMESET></HTML>");
            close();
        }
    }

    function frame_aufheben()
    { parent.location.href = self.location.href;}

// -->
</SCRIPT>
</HEAD>
<BODY>
    <A HREF="javascript: frame_erzeugen()">Frame erzeugen</A>
    <A HREF="javascript: frame_aufheben()">Frame aufheben</A>
</BODY>
</HTML>

```

3.16. *Dokument nach dem Laden als oberstes Fenster anzeigen, also sichtbar*

notwendig, wenn Dokument andere Fenster aufruft, die schneller geladen werden als das Dokument
 <BODY ... onLoad="if(window.focus) window.focus()">

3.17. *Dokument drucken*

Hinweis: Seitenumbruch an Absatzende erzeugen lassen beim Druck:

```

<STYLE TYPE="text/css">
@media print {css_klasse_fuer_seitenumbruch { page-break-after:always;}}
</STYLE>

...

<P CLASS="css_klasse_fuer_seitenumbruch">
....
</P>

function printit()
{
    if (NS)
    {
        document.write("<P><A HREF=\"javascript:self.print()\">
        Drucke Dokument</A></P>");
    }
    else
    {
        // IE
        var WebBrowser = '<OBJECT
                                ID="ObjektID"
                                WIDTH=0
                                HEIGHT=0

```




```

        CLASSID="CLSID:8856F961-340A-11D0-A96B-00C04FD705A2"
    >
</OBJECT>;

document.body.insertAdjacentHTML('beforeEnd', ObjektID);

ObjektID.ExecWB(6, 2);

}

}

var NS = (navigator.appName == "Netscape");
var VERSION = parseInt(navigator.appVersion);

if (VERSION > 3)
{
    document.write(
        '<FORM>'
        + '<INPUT TYPE =button'
        + ' VALUE="Drucke diese Seite"'
        + ' NAME="Print"'
        + ' onclick="printit();"
        + '>'
        + '</form>'
    );
}

```

Seitenelemente vom Druck ausschließen:

ab IE 4.x und NS 6.x

Seitenteile, die nicht gedruckt werden sollen, in `<DIV CLASS="keindruck">` und `</DIV>` oder in `` und `` einschliessen.

zwischen `<HEAD>` und `</HEAD>` einfügen:

```
<LINK REL="stylesheet" MEDIA="print" HREF="print.css">
```

print.css enthält nur

```
.keindruck {display:none;}
```

Seitenelemente nur beim Druck anzeigen:

ab IE 4.x

Seitenteile, die nur auf Ausdrucken sichtbar sein sollen, in `<DIV CLASS="nurdruck">` und `</DIV>` oder in `` und `` einschliessen

zwischen `<HEAD>` und `</HEAD>` einfügen:

```
<LINK REL="stylesheet" MEDIA="screen" HREF="screen.css">
```

screen.css enthält nur

```
.nurdruck {display:none;}
```

3.18. **Dokument automatisch scrollen**

Nachfolgendes Beispiel scrollt um einen Festwert an Pixeln.

Soll das Scrollen aber unabhängig vom Festwert sein, dann muss beachtet werden:

- Scrollen sollte dann aktiv werden, wenn die Maus eine bestimmte Stelle vom Fenster erreicht (onmouseover) - sinnvoller Weise obere Fensterkante für nach oben scrollen untere Fensterkante für nach unten scrollen.

Die Scrollschrittweite sollte variabel sein - sinnvoller Weise

- wachsend je näher Richtung o.g. Kanten
- sinkend je mehr in der entgegengesetzten Kantenrichtung
- kein scrollen wenn mouseover-Bereich verlassen

Das Scrollen muss durch Rekursion erfolgen, die allerdings CPU-Ressourcen benötigt.

Der Mauszeiger sollte an die Scroll-Situation angepasst werden.

Wenn Scrollen per onmouseover eingerichtet, so können die Scrollleisten im Body dauerhaft abgeschaltet werden.

- Es sollte aber dem User eine Möglichkeit gegeben werden, Scrollleisten zu aktivieren und damit automatisches Scrollen abzuschalten, denn es gibt Menschen, die mögen es, Mäusräder zu drehen.

Beispiel: Scrollen um Festwert an Pixeln

<HTML>



```

<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var pixel_anzahl = 2000;           // 0,0 ist links oben
      function scrollen()
      {
        for (i=1; i<pixel_anzahl; i++)
        { scrollTo(0,i); }
      }
    //-->
  </SCRIPT>
</HEAD>

<BODY>
  <FORM>
    <INPUT TYPE=button
      VALUE="Start scrollen!"
      onClick="scrollen()">
  </FORM>
</BODY>
</HTML>

```

3.19. *Reload des Dokumentes nach Resize des Browserfensters*

```
<BODY onResize="window.location.href = window.location.href;"></BODY>
```

3.20. *Festplattenverzeichnis unter Windows auslesen*

Es kann sein, dass die Firewall aktiv wird.

```

<HTML>
  <HEAD>
  </HEAD>

  <BODY>
    <FORM ACTION="file:///C:/"
      METHOD="GET">
      <INPUT TYPE="submit"
        VALUE="Root von LW C anzeigen!">
    </FORM>
  </BODY>
</HTML>

```

3.21. *Dokumenten-Hintergrund*

3.21.1. *Dokumenten-Hintergrund ein- bzw. ausblenden*

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript1.2">
  <!--
    //      Dokument-Hintergrundfarbe ein- und ausblenden
    //      einblenden beim Dokument laden
    //      ausblenden beim Wechsel zu anderem Dokument, also entladen des aktuellen
    //      Dokumentes
    //      mit Farbveränderung rückwärts gegenüber Laden
    //
    //*****
    //
    //      Nachfolgende Variablen muss der Programmierer anpassen
    //
    //*****
    //      Farbe wird aus Rot- und Gruen- und Blauanteil gebildet
    //      Alle Farbanteilwerte gelten von 0 bis 255, alles ganzzahlig, wird nicht geprüft
    //      Startwerte für Startfarbe des Ein- bzw. Ausblenden, müssen <= Endwerte sein, wird nicht geprüft
    //      Endwerte für Endfarbe nach Ein- bzw. Ausblenden, müssen >= Startwerte sein, wird nicht geprüft
    //      Farbschritte = Stufen für Ein- bzw. Ausblenden
    //
    //      je mehr um so langsamer das Ein- bzw. Ausblenden
    //      nur ganzzahlig und maximal die kleinste paarige Differenz der obigen Werte
    //      (keine Prüfung)
    var RotAnteil_Start=0;
    var GruenAnteil_Start=0;
    var BlauAnteil_Start=0;

```



```

var RotAnteil_Ende=255;
var GruenAnteil_Ende=255;
var BlauAnteil_Ende=255;
var FarbSchritte=64;
//

// *****
//
//      Nachfolgenden Code nicht verändern
//
// *****
function DezimalZuHexaString(DezimalerWert)
{
    var HexaZeichenFeld = "0123456789ABCDEF";
    var HexaKette="";

    if (DezimalerWert < 0)
    { HexaKette="00";}
    else
    {
        if (DezimalerWert > 255)
        { HexaKette="FF";}
        else
        {
            //      Indexe im HexaZeichenFeld ermitteln
            var HexaKette_Zeichen1_PositionImHexaZeichenFeld =
                Math.floor(DezimalerWert / 16);
            var HexaKette_Zeichen2_PositionImHexaZeichenFeld = DezimalerWert % 16;

            //      HexaKette bilden
            HexaKette= HexaZeichenFeld.charAt(HexaKette_Zeichen1_PositionImHexaZeichenFeld)
                +
                HexaZeichenFeld.charAt(HexaKette_Zeichen2_PositionImHexaZeichenFeld);
        }
    }
    return HexaKette;
}

function BackgroundEinAusBlenden(RotAnteil_Start,GruenAnteil_Start,BlauAnteil_Start,
    RotAnteil_Ende,GruenAnteil_Ende,BlauAnteil_Ende,
    FarbSchritte
)
{
    //      Farbe wird aus Rot- und Gruen- und Blauanteil gebildet
    //      Alle Farbanteilwerte gelten von 0 bis 255, alles ganzzahlig, wird nicht geprüft
    //      Startwerte für Startfarbe des Ein- bzw. Ausblenden, müssen <= Endwerte sein, wird nicht geprüft
    //      Endwerte für Endfarbe nach Ein- bzw. Ausblenden, müssen >= Startwerte sein, wird nicht geprüft
    //      Farbschritte = Stufen für Ein- bzw. Ausblenden
    //                        je mehr um so langsamer das Ein- bzw. Ausblenden
    //                        nur ganzzahlig und maximal die kleinste paarige Differenz der obigen Werte
    //                        (keine Prüfung)

    for(var i = 0; i <= FarbSchritte; ++i)
    {
        var RotAnteil = Math.floor( (RotAnteil_Start * ((FarbSchritte - i) / FarbSchritte))
            + (RotAnteil_Ende * (i / FarbSchritte))
        );
        var GruenAnteil = Math.floor( (GruenAnteil_Start * ((FarbSchritte - i) / FarbSchritte))
            + (GruenAnteil_Ende * (i / FarbSchritte))
        );
        var BlauAnteil = Math.floor( (BlauAnteil_Start * ((FarbSchritte - i) / FarbSchritte))
            + (BlauAnteil_Ende * (i / FarbSchritte))
        );

        document.bgColor =      "##"
            + DezimalZuHexaString(RotAnteil)
            + DezimalZuHexaString(GruenAnteil)
            + DezimalZuHexaString(BlauAnteil);
    }
}

```



```

}

//      Nachfolgende Funktionen existieren NUR, um die globalen Variablen für BODY verfügbar zu machen
//      da in der HTML-Body-Anweisung keine Javascript-Variablen akzeptiert werden
function EinAusBlenden_DokumentLaden()
{
    BackgroundEinAusBlenden(RotAnteil_Start,GruenAnteil_Start,BlauAnteil_Start,
                            RotAnteil_End,GruenAnteil_End,BlauAnteil_End,
                            FarbSchritte);
}

function EinAusBlenden_DokumentEntladen()
{
    //      ist das Laden rückwärts
    BackgroundEinAusBlenden(RotAnteil_End,GruenAnteil_End,BlauAnteil_End,
                            RotAnteil_Start,GruenAnteil_Start,BlauAnteil_Start,
                            FarbSchritte);
}
// -->
</SCRIPT>
</HEAD>
<BODY BGCOLOR=#ffffff      onload="EinAusBlenden_DokumentLaden()"
                        onunload="EinAusBlenden_DokumentEntladen()">

</BODY>
</HTML>

```

3.21.2. Dokumenten-Hintergrund auswählen

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript1.2">
<!--
//      Neues Fenster öffnen mit selektierten Hintergrund
//
function HintergrundSetzen(bgname)
{
    var Fenster=window.open("", "", "width=380,height=380");
    Fenster.focus();
    Fenster.document.open();
    Fenster.document.write( "<HTML><HEAD></HEAD>"
                            + "<BODY BACKGROUND="
                                + ""
                                + bgname
                                + ""
                            + ">"
                            + "</BODY>"
                            + "</HTML>");
    Fenster.document.close();
}
//-->
</SCRIPT>
</HEAD>
<BODY>
    <A HREF="getbackg.htm" onclick="HintergrundSetzen('bg1.jpg');return false">
        <IMG SRC="bg1.jpg" BORDER="0" WIDTH="96" HEIGHT="96">
    </A>
    <A HREF="getbackg.htm" onclick="HintergrundSetzen('bg2.jpg');return false">
        <IMG SRC="bg2.jpg" BORDER="0" WIDTH="96" HEIGHT="96">
    </A>
</BODY>
</HTML>

```

4. Fenster

4.1. Textausgaben im aktuellen Fenster

4.1.1. mit der Methode document.write()

Beispiel: Berechnen und ausgeben

```
<HTML>
```



```

<SCRIPT LANGUAGE="JavaScript">
<!--
    document.write("Hallo Welt! ");
//-->
</SCRIPT>
</HTML>

```

4.1.2. mit der Methode alert() (Alert-Fenster)

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var x = 17;
    alert("Der Wert von x ist " + x);
// -->
</SCRIPT>
</HTML>

```

4.1.3. schwebender Text von rechts nach links bei 640 Pixel Bildschirmbreite

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    //      IE unterstützt kein Layer-Objekt. Dafür wird beim IE das DIV-Tag-Objekt verwendet.
    //      Netscape unterstützt zwar Layer-Objekt <LAYER> ...</LAYER>,
    //      aber hier aus Gründen der Einfachheit auch das DIV-Tag-Objekt verwenden

    function starte_bewegung()
    {
        if (document.layers) // Netscape
        {
            bewegtes_text_objekt=document. DIV_Id; // Typ laut Netscape
            bewegen();
        }
        else
        {
            if (document.all) // Microsoft
            {
                bewegtes_text_objekt = DIV_Id.style; // Typ laut Microsoft
                bewegen();
            }
        }
    }

    function bewegen() // rekursive Funktion mit setTimeout auf 200 Millisekunden Wartezeit
    {
        ganze_zahl=parseInt(objekt.left); // parseInt erzeugt ganze Zahl, also auch negativ
        bewegtes_text_objekt.left=ganze_zahl - 3; // Bewegung um 3 Pixel

        if (ganze_zahl < 0) { bewegtes_text_objekt.left=640 }
        setTimeout('bewegen()',200);
    }
//-->
</SCRIPT>
</HEAD>

<BODY onload="starte_bewegung()">
// Microsoft unterstützt CSS-Layer !
<DIV ID="DIV_Id"
    STYLE="position:absolute;top:100px;left:250px;z-index:2;height:50px;width:150px"
    // Achtung: STYLE alles 1 Zeile da Zeichenkettenparameter
    // keine Blanks wegen Netscape
>
<H1 ALIGN="center">Bewegter Text</H1>
</DIV>
</BODY>
</HTML>

```

4.1.4. Fenster des Browser rausschieben und danach neu anzeigen

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
//

```



```
//
//      Browserfenster nach rechts unten rausscrollen und anschliessend ohne Neuladen wieder normal anzeigen
//
//*****
//
//      vom Programmierer ist nichts zu setzen
//      nachfolgender Code darf nicht verändert werden

var BrowserFenster_AktuelleBreite=2000; // sollte größer als max. übliche Auflösung sein
var BrowserFenster_AktuelleHoehe=2000; // sollte größer als max. übliche Auflösung sein

function moveWin()
{
    for (var i = 1; i < BrowserFenster_AktuelleBreite; i++)
    {window.moveBy(1, 1);}

    window.moveBy((-1)* BrowserFenster_AktuelleBreite,(-1) * BrowserFenster_AktuelleHoehe);
}
//-->
</SCRIPT>
</HEAD>
<BODY onload="moveWin();">
</BODY>
</HTML>
```

4.1.5. Textstreifen senkrecht auf Bildschirm animieren

```
<HEAD>
<script language="JavaScript">

// Matrix-Bildschirmschoner-Animation

//      Warnung: CPU-Auslastung kann sehr hoch sein.

// +++++ vom Programmierer einstellbare Variablen
var MatrixSpalten_Anzahl=33; // 33 bei BS-Fenster 1024x765 und
//                               ZeichenBlock_Font auf Arial
//                               MatrixSpalten_BreiteUndHoehe 8 oder 15
//                               MatrixSpalten_Abstand 30
// Anzahl ab 1 der Spalten auf dem BS
// wenn < 1 so automatisch ermitteln anhand
//                               MatrixSpalten_BreiteUndHoehe
//                               aktuelle Fensterbreite
//                               ZeichenBlock_AnzeigeArt immer 0
// je höher um so höher muss ZeichenBlock_FallSpeedMax sein
// Spaltenanzahl hängt ab von
//                               MatrixSpalten_BreiteUndHoehe Spaltenbreite- und höhe
//                               MatrixSpalten_Abstand Spaltenabstand
//                               ZeichenBlock_Font Schriftart und damit

Zeichenbreite

// Die gerenderte Spaltenbreite wird diesen Werten automatisch angepasst,
// kann aber wegen Spaltenanzahl bewirken,
// dass horizontale Scrolleisten auftauchen,
// ausser man hat Scrolleisten für Body abgeschaltet
// Wenn Scrolleisten für Body abgeschaltet:
// Spalten, die deswegen nicht gerendert werden, können trotzdem
// berechnet werden !
var MatrixSpalten_BreiteUndHoehe=8; // Spaltenbreite und -höhe ab 1, wenn < 1 so 8
// hängt ab von ZeichenBlock_Font
// Schriftart und damit Zeichenbreite
// Die gerenderte Spaltenbreite wird diesen Werten automatisch angepasst.
var MatrixSpalten_Abstand=30; // Spaltenabstand ab 1 in Pixel
var MatrixAnimationTraegheit=50; // Trägheit > 0 der Matrixanimation
// je höher um so Träger
// je kleiner um so mehr CPU-Ressourcen werden benötigt
// kann um so höher je höher ZeichenBlock_FallSpeedMax
// ist der Timeoutwert in Milliskundender Rekursion von MatrixAnim()
var ZeichBlock_FallArt=0; // 0 für Standard für differenziertes Fallen der Zeichenblöcke ab 1. Fallen
// 1 für allmähliches differenziertes Fallen der Zeichenblöcke ab 2. Fallen
// 2 für tropfend von oben
```




```

'T',
',',
'D',
'W',
'#'
);

// +++++ interne Variablen
var ZeichenBlock_FontWechselFarben_FeldLaenge=ZeichenBlock_FontWechselFarben_Feld.length;
var ZeichenBlock_WechselZeichen_FeldLaenge=ZeichenBlock_WechselZeichen_Feld.length;
var X_Container; // Container-DIV
var X_SpaltenFeld; // Feld der Spalten-DIV im Container
var Spalten_PosY_AktuellFeld;
var Spalten_PosY_FallSpeedFeld;
var Spalten_innerText_Aktuell_Feld;
var Spalten_TextColorFeld;
var ZeichenBlock_WechselZeichen_StandardFeld;
var ZeichenBlock_WechselZeichen_StandardFeldLaenge;
var MatrixErzeugungErfolgreich=false; // true so Matrix-Objekte erzeugt
var MatrixSpalten_BreiteUndHoeheNegativ=0 // Spaltenbreite negativ
var MatrixAnimW1=0;
var MatrixAnimW2=0;
var MatrixAnimW3=0;
var MatrixAnimW4=0;
var MatrixAnimW5=0;
var MatrixAnimW6=0;
var MatrixAnimW7=0;
var MatrixAnimK1="";
var MatrixAnimK2="";
var MatrixAnimZ1;
var MatrixAnimF1=false;

// +++++ Zufallszahl erzeugen
function ZufallsZahl(X00) // Zufallszahl von inklusive 0
                        // bis inklusive X00
                        // erzeugen als ganze Zahl >=0
// X00 wenn Gleitkomma, so vor Verwendung nach ganzzahlig konvertiert
// wenn negativ, so Absolutwert verwendet und wenn dann Gleitkomma, dann zu ganzer Zahl
// Bsp.: -22.5 wird zu 22.5 wird zu 22
// jedoch NICHT -22.5 wird zu -23 wird zu 23
// liefert 0 wenn X00 0 ist oder Parameterfehler
{var X01=0; // Returnwert
var X02=0;
var X03=0;
var X04=0;
var X05=0;
var X06=false;
if(X00!=null)
{if(X00!=0)
{X02=Math.abs(X00); // X00 zu positiv
X03=Math.floor(X02); // nächste ganze Zahl kleiner X00
X04=X03;
X04++; // +1 wird durch nachfolgendes floor aufgehoben
X05=Math.random(); // 1. Zufallswert holen und auf > 0 prüfen
X05*=X04;
X01=Math.floor(X05); // nächste ganze Zahl kleiner X05
if(X01>0) // > 0 also 2. Zufallswert holen ...
{if(X03<6) // .... nur für X03 1 bis 5: Zufallswert mit X03*10 ermitteln
{X04=X03;
X04*=10;
X04++; // +1 wird durch nachfolgendes floor aufgehoben
X05=Math.random(); // 2. Zufallswert inklusive 0
X05*=X04;
X01=Math.floor(X05); // nächste ganze Zahl kleiner X05
if(X01>0)
{X00=0; // Annahme: 0 liefern
X02=0; // Schrittzähler init
do // Bsp: X00 ist 17, also X04 ist 170, also X01 ist 0 bis 170
{X06=(X01>X02); // 0 bis 16 in 17 Zehnerschritten laut X02

```




```

//          X01 > 0 also mindestens 1 liefern
//          X01 > 10 also mindestens 2 liefern
//          X01 > 20 also mindestens 3 liefern
//          ....
//          X01 > 160 also          17 liefern
// Returnwert+1 und Schrittzähler +10

if(X06){X00++;X02+=10;}
}
while(X06);
X01=X00;
}
}
if(X01>X03){X01=X03;}
}
}
return X01;
}

// +++++ Matrix animieren
function MatrixAnim()           // interne Routine, rekursiv
{ // +++++ prüfen ob Matrix-Objekte erzeugt wurden
if(MatrixErzeugungErfolgreich)
{ // +++++ Annahme: Fallen der Zeichenblöcke überschreitet nicht die Höhe des BS-Fensters
//                                     also es fällt nichts mehr in allen Spalten,
//                                     also Neustart der Matrix-Animation notwendig
MatrixAnimF1=false;
// +++++ Container-PosY an Scrollposition anpassen
X_Container.style.top=document.body.scrollTop;
// +++++ Spalten abklappern
for(MatrixAnimW1=0;MatrixAnimW1<MatrixSpalten_Anzahl;MatrixAnimW1++)
{
// +++++ PosY verändern
// ----- prüfen auf Fallart der Zeichenblöcke laut ZeichBlock_FallArt
// 0   für Standard für differenziertes Fallen der Zeichenblöcke ab 1. Fallen
// 1   für allmähliches differenziertes Fallen der Zeichenblöcke ab 2. Fallen
// 2   für tropfend von oben
// < 0 für zufällig 0 bis 2
MatrixAnimW6=ZeichBlock_FallArt;
if(MatrixAnimW6<0){MatrixAnimW6=ZufallsZahl(2);}
// - - - wenn 0
if(MatrixAnimW6==0){Spalten_PosY_AktuellFeld[MatrixAnimW1]+=Spalten_PosY_FallSpeedFeld[MatrixAnimW1];}
// - - - wenn 1
if(MatrixAnimW6==1){Spalten_PosY_AktuellFeld[MatrixAnimW1]++;}
// - - - wenn 2
if(MatrixAnimW6==2)
{MatrixAnimW6=ZufallsZahl(Spalten_PosY_AktuellFeld[MatrixAnimW1]);
if(MatrixAnimW6<1){MatrixAnimW6=1;}           // wie 1
Spalten_PosY_AktuellFeld[MatrixAnimW1]+=MatrixAnimW6;
}

// +++++ wenn BS-Höhe dabei überschritten (Überschreitung in Länge des Zeichenblocks der Spalte)
if(Spalten_PosY_AktuellFeld[MatrixAnimW1]>document.body.clientHeight)
{ // +++++ Standardfarbe für Zeichenblock setzen
X_SpaltenFeld[MatrixAnimW1].style.color=ZeichenBlock_FontFarbe_Standard;
Spalten_TextColorFeld[MatrixAnimW1]=ZeichenBlock_FontFarbe_Standard;
// +++++ Für alle Spalten die Zeichenblöcke erneuern
for(MatrixAnimW2=0;MatrixAnimW2<MatrixSpalten_Anzahl;MatrixAnimW2++)
{ // +++++ Zeichenblock erzeugen
MatrixAnimK2="";
// ----- Endewert der Schleife zufällig ermitteln, also Zeichenblocklänge
MatrixAnimW7=ZufallsZahl(ZeichenBlock_LaengeMax);
//                                     // Zufallszahl von inklusive 0
//                                     //                                     bis inklusive X00
//                                     // erzeugen als ganze Zahl >=0

MatrixAnimW7++;           // +1 falls 0 war
// ----- zeichenweise erzeugen
for(MatrixAnimW3=0;MatrixAnimW3<MatrixAnimW7;MatrixAnimW3++)
{
// ----- neues Zeichen zufällig wählen

```



```

if(Zeichenblock_ZeichenLautZeichen_Feld) // Zeichen aus ZeichenBlock_WechselZeichen_Feld holen
{MatrixAnimW6=ZufallsZahl(ZeichenBlock_WechselZeichen_FeldLaenge-1);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0
MatrixAnimK1=ZeichenBlock_WechselZeichen_Feld[MatrixAnimW6];
}
else // Zeichen aus ZeichenBlock_WechselZeichen_StandardFeld holen
{MatrixAnimW6=ZufallsZahl(ZeichenBlock_WechselZeichen_StandardFeldLaenge-1);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0
MatrixAnimK1=ZeichenBlock_WechselZeichen_StandardFeld[MatrixAnimW6];
}

// ----- Zeichenblocklage auswerten laut ZeichenBlock_AnzeigeArt
// 0 so Zeichenblock senkrecht gerendert, also als Spalte
// 1 so Zeichenblock senkrecht gerendert, also als Zeile
// < 0 so zufällig 0 bis 1
MatrixAnimW6=ZeichenBlock_AnzeigeArt;
if(MatrixAnimW6<0){MatrixAnimW6=ZufallsZahl(1);}
if(MatrixAnimW6==0){MatrixAnimK1+=' ';}
// sonst kein Trennzeichen also Zeichblock horizontal gerendert in Spalte
// da Spalte DIV ist, wird der in der Breite automatisch angepasst
// ----- Zeichen kumulieren
MatrixAnimK2+=MatrixAnimK1;
// ----- Zeichenblock kumuliert als innerText der Spalte die die BS-Höhenüberschreitung verursacht hat
Spalten_innerText_Aktuell_Feld[MatrixAnimW1]=MatrixAnimK2;
// +++++ neues PosY ermitteln
MatrixAnimW5=MatrixSpalten_BreiteUndHoeheNegativ;
MatrixAnimW5*=Spalten_innerText_Aktuell_Feld[MatrixAnimW1].length;
Spalten_PosY_AktuellFeld[MatrixAnimW1]=MatrixAnimW5;
}
// +++++ Fallspeed für PosY zufällig ermitteln
MatrixAnimW6=ZufallsZahl(ZeichenBlock_FallSpeedMax);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0
MatrixAnimW6++; // +1 falls 0 war
Spalten_PosY_FallSpeedFeld[MatrixAnimW1]=MatrixAnimW6;
}

// +++++ Spalte rendern
// ----- Zeiger der Spalte
MatrixAnimZ1=X_SpaltenFeld[MatrixAnimW1];
// ----- .top aktualisieren
MatrixAnimZ1.style.top=Spalten_PosY_AktuellFeld[MatrixAnimW1];
// ----- .innerText belegen
MatrixAnimZ1.innerText=Spalten_innerText_Aktuell_Feld[MatrixAnimW1];
// ----- Wechselfarbe zufällig wählen
MatrixAnimF1=ZeichenBlock_FontFarbePermanentWechseln; // Annahme: permanent zufällig wechseln
if(!MatrixAnimF1){MatrixAnimF1=(Spalten_TextColorFeld[MatrixAnimW1]==ZeichenBlock_FontFarbe_Standard);}
// wenn nicht permanent gewechselt werden soll, dann
// nur Wechsel, wenn Standardfarbe vorhanden ist

if(MatrixAnimF1)
{ // ----- prüfen welche Farbquelle
if(ZeichenBlock_FontFarbeLautFarben_Feld) // Farbe laut ZeichenBlock_FontWechselFarben_Feld
{MatrixAnimW6=ZufallsZahl(ZeichenBlock_FontWechselFarben_FeldLaenge-1);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0
MatrixAnimK1=ZeichenBlock_FontWechselFarben_Feld[MatrixAnimW6];
MatrixAnimZ1.style.color=MatrixAnimK1;
Spalten_TextColorFeld[MatrixAnimW1]=MatrixAnimK1;
}
else // alle möglichen Farben
{MatrixAnimW6=ZufallsZahl(16777215); // 16777215 entspricht '#FFFFFF'
MatrixAnimZ1.style.color=MatrixAnimW6;
}
}

```



```

    Spalten_TextColorFeld[MatrixAnimW1]=MatrixAnimW6.toString();    // Ziffernfolge als Pseudo-Farbname nehmen
  }
}
// +++++ Rekursion
window.setTimeout('MatrixAnim()',MatrixAnimationTraegheit);
} // Matrix-Objekt sind erzeugt
}

// ----- Matrix-Elemente erzeugen
function MatrixErzeugen()      // interne Routine
// liefert true, wenn Matrix erzeugt wurde
// setzt MatrixErzeugungErfolgreich auf Returnwert
{ var X00=false;
  var X01;
  var X02=0;
  var X03=false;
  // +++++ Annahme: Erzeugung nicht möglich
  MatrixErzeugungErfolgreich=false;
  // +++++ Container-DIV erzeugen
  X01=document.createElement("DIV");
  X00=(X01!=null);
  if(X00)
  {X_Container=document.body.appendChild(X01);
  // +++++ Spalten-DIV's im Container erzeugen
  X_SpaltenFeld=new Array();
  for(X02=0;X02<MatrixSpalten_Anzahl;X02++)
  { X01=document.createElement("DIV");
    X03=(X01!=null);
    if(X03){X_SpaltenFeld[X02]=X_Container.appendChild(X01);}
    else{X00=false;}
  }
  }
  MatrixErzeugungErfolgreich=X00;
  return X00;
}

// ----- erzeugte Matrix init und start
function MatrixInitUndStart()      // auch von Programmierer aufrufbar, wiederholbar
{ var X00;
  var X01=0;
  var X02=0;
  var X03=0;
  var X04="";
  var X05=0;
  var X06="";
  // +++++ prüfen ob Matrix-Objekte erzeugt wurden
  if(MatrixErzeugungErfolgreich)
  { // +++++ Objekte Layout
    // ----- Container-DIV
    X00=X_Container.style;
    X00.position='absolute';
    X00.top=0;
    X00.left=(-1*MatrixSpalten_Abstand);
    // ----- Spalten-DIV
    X02=MatrixSpalten_Abstand;
    Spalten_TextColorFeld=new Array();
    for(X01=0;X01<MatrixSpalten_Anzahl;X01++)
    { X00=X_SpaltenFeld[X01].style;
      X00.position='absolute';
      X00.top=0;
      X02+=MatrixSpalten_Abstand;X00.left=X02;
      X00.width=MatrixSpalten_BreiteUndHoehe;
      X00.height=0;
      X00.color=ZeichenBlock_FontFarbe_Standard;
      Spalten_TextColorFeld[X01]=ZeichenBlock_FontFarbe_Standard;
      X00.fontFamily=ZeichenBlock_Font;
      X00.fontSize=MatrixSpalten_BreiteUndHoehe;
    }
  }
}

```



```

Spalten_PosY_FallSpeedFeld=new Array();
Spalten_PosY_AktuellFeld=new Array();
Spalten_innerText_Aktuell_Feld=new Array();

for(X01=0;X01<MatrixSpalten_Anzahl;X01++)
{
// +++++ PosY der Spalte auf 0 (also .top)
Spalten_PosY_AktuellFeld[X01]=0;

// +++++ Fallspeed für PosY zufällig ermitteln
X03=ZufallsZahl(ZeichenBlock_FallSpeedMax);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0

X03++; // +1 falls 0 war
Spalten_PosY_FallSpeedFeld[X01]=X03;

// +++++ Zeichenblock erzeugen: genau 1
X04="";

// ----- Endwert der Schleife zufällig ermitteln, also Zeichenblocklänge
X05=ZufallsZahl(ZeichenBlock_LaengeMax);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0

X05++; // +1 falls 0 war
for(X02=0;X02<X05;X02++)
{
// ----- neues Zeichen zufällig wählen
if(Zeichenblock_ZeichenLautZeichen_Feld) // Zeichen aus ZeichenBlock_WechselZeichen_Feld holen
{X03=ZufallsZahl(ZeichenBlock_WechselZeichen_FeldLaenge-1);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0

X06=ZeichenBlock_WechselZeichen_Feld[X03];
}
else // Zeichen aus
ZeichenBlock_WechselZeichen_StandardFeld holen
{X03=ZufallsZahl(ZeichenBlock_WechselZeichen_StandardFeldLaenge-1);
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0

X06=ZeichenBlock_WechselZeichen_StandardFeld[X03];
}

// ----- Zeichenblocklage auswerten laut ZeichenBlock_AnzeigeArt
// 0 so Zeichenblock senkrecht gerendert, also als Spalte
// 1 so Zeichenblock senkrecht gerendert, also als Zeile
// < 0 so zufällig 0 bis 1
X03=ZeichenBlock_AnzeigeArt;
if(X03<0){X03=ZufallsZahl(1);}
// Zufallszahl von inklusive 0
// bis inklusive X00
// erzeugen als ganze Zahl >=0

if(X03==0){X06+=' ';}
// sonst kein Trennzeichen also Zeichblock horizontal gerendert in Spalte
// da Spalte DIV ist, wird der in der Breite automatisch angepasst

// ----- Zeichen kumulieren
X04+=X06;

// ----- Zeichenblock kumuliert als innerText merken
Spalten_innerText_Aktuell_Feld[X01]=X04;
}
}

// +++++ Matrix animieren
MatrixAnim(); // rekursiv MatrixAnimieren

```



```

} // Matrix-Objekte sind erzeugt
}

// ----- Matrix erzeugen und init und start
function MatrixErzeugenUndInitUndStart() // vom Programmierer aufrufbar, nicht wiederholbar
// liefert true wenn init erfolgreich
{ var X00=false;
  var X01=0;
  var X02="";
  // +++++ Annahme: Init nicht erfolgreich
  MatrixErzeugungErfolgreich=false;           // true so Matrix-Objekte erzeugt

  // +++++ Matrix-Spaltenbreite prüfen
  if(MatrixSpalten_BreiteUndHoehe<1){MatrixSpalten_BreiteUndHoehe=8;}
  MatrixSpalten_BreiteUndHoeheNegativ=-1*MatrixSpalten_BreiteUndHoehe;

  // +++++ Matrix-Spaltenabstand prüfen
  if(MatrixSpalten_Abstand<1){MatrixSpalten_Abstand=1;}

  // +++++ Anzahl der Spalten in der Matrix berechnen
  if(MatrixSpalten_Anzahl<1)
  {ZeichenBlock_AnzeigeArt=0;                  // 0 so Zeichenblock senkrecht gerendert, also als Spalte
                                           // 1 so Zeichenblock senkrecht gerendert, also als Zeile
                                           // < 0 so zufällig 0 bis 1
                                           // immer 0 verwendet wenn MatrixSpalten_Anzahl < 1
    X01=document.body.clientWidth;             // aktuelle Fensterbreite
    X01/=(MatrixSpalten_BreiteUndHoehe+MatrixSpalten_Abstand);
                                           // durch Summe aus Spaltenbreite und Spaltenabstand zu Gleitkomma
    MatrixSpalten_Anzahl=Math.floor(X01);       // und ganze Zahl direkt davor
  }

  // +++++ Trägheit der Matrix-Animation prüfen
  if(MatrixAnimationTraegheit<1){MatrixAnimationTraegheit=20;}

  // +++++ Feld der Standard-Wechselzeichen füllen
  X02="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890(){ }!\"$%&[]=?\"ÄÖÜß*+#+#,-_';
                                           // nicht Zeichen wie ' oder > oder < / oder \ oder ° oder ^ etc.
  ZeichenBlock_WechselZeichen_StandardFeld=X02.split("");
  ZeichenBlock_WechselZeichen_StandardFeldLaenge=ZeichenBlock_WechselZeichen_StandardFeld.length;

  // +++++ ERST alle leer Objekte erzeugen
  X00=MatrixErzeugen();                       // liefert true, wenn Matrix erzeugt wurde
                                           // setzt MatrixErzeugungErfolgreich auf Returnwert

  // +++++ Scrolleleisten des Body abschalten
  document.body.style.overflowX='hidden';      // waagerechte Scrolleleisten
  // Aktionen sobald Objekt-Dimension > Fensterdimension (Breite und oder Höhe):
  // 'visible' Objekt anzeigen und keine Scrollbars aktivieren, Default außer bei document.body
  // 'scroll' Objekt anzeigen und Scrollbar immer aktivieren
  // 'hidden' Objekt nicht anzeigen und damit auch keine Scrollbars möglich (entspricht visibility='hidden')
  // 'auto' Objekt anzeigen und Scrollbar nach Bedarf, Default nur für document.body
  document.body.style.overflowY='hidden';      // senkrechte Scrolleleisten

  // +++++ body-Hintergrundfarbe setzen
  document.body.style.backgroundColor='green';

  // +++++ Init und start
  if(X00){MatrixInitUndStart();}              // liefert true wenn init erfolgreich

  return X00;
}
</script>
</HEAD>
<BODY>
<script language="JavaScript">
alert(MatrixErzeugenUndInitUndStart());
</script>
</BODY>

```



4.2. *Popup-Fenster per open()*

mit alert(); oder nachfolgender Variante

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
    var Fenster;

    function PopupFensterOeffnen()
    {
        Fenster=window.open("", "Neu", "HEIGHT=200, WIDTH=400");
        Fenster.write("Dieses Fenster schliesst sich nach 5 Sekunden !");
        window.setTimeout("PopupFensterSchliessen()", 5000);
    }

    function PopupFensterSchliessen()
    {Fenster.close();}
    // beachte: setTimeout lässt leider keine weiteren Parameter zu !
-->
</SCRIPT>
</HEAD>
<BODY>
    <FORM>
        <INPUT TYPE="button"
            VALUE="Fenster oeffnen"
            onClick=" PopupFensterOeffnen()">
    </FORM>
</BODY>
</HTML>
```

4.3. *Nicht-Popup-Fenster per open()*

4.3.1. Fenster öffnen

4.3.1.1. mit Standardgröße öffnen

Beispiel Fenster öffnen und fremde Homepage laden

```
<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JavaScript">
            <!--
                function FensterOeffnen()
                { neuesFenster = open("http://www.test.de"); }
            -->
        </SCRIPT>
    </HEAD>

    <BODY>
        <FORM>
            <INPUT TYPE="button"
                VALUE="Oeffnen"
                onClick="FensterOeffnen()">
        </FORM>
    </BODY>
</HTML>
```

4.3.1.2. mit vordefinierter Größe öffnen

Fenster öffnen mit vordefinierter Größe und fremde Homepage laden

```
<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JavaScript">
            <!--
                function FensterOeffnen()
                {
                    neuesFenster = open("http://www.test.de",
                        "anzeigeFenster",
                        "height=300, width=500");
                }
            -->
        </SCRIPT>
    </HEAD>

    <BODY>
        <FORM>
            <INPUT TYPE="button"
                VALUE="Oeffnen"
                onClick="FensterOeffnen()">
        </FORM>
    </BODY>
</HTML>
```



```

        // -->
    </SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <INPUT TYPE="button"
            VALUE="Oeffnen"
            onClick="FensterOeffnen()">

    </FORM>
</BODY>
</HTML>

```

4.3.1.3. ohne Eigenschaften eines Browserfensters öffnen

```

<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function open_window(datei)
        {
            url=dateiname + ".htm";
            var fenster=windows.open(
                url,
                dateiname,
                "toolbar=0,location=0,directories=0,status=0,
                menubar=0,scrollbars=1,resizeable=1,
                screenX=100,screenY=20,width=600,height=140"
            );
            // ACHTUNG: "toolbar=0.....height=140" alles in EINE Zeile und OHNE Blanks
            // zwischendurch !!!

            if(navigator.appName.subString(0,8) == "Netscape" { fenster.location=url; }

        }
    <!-->
    </SCRIPT>
</HEAD>

```

Aufruf per `open_window("name_der_html_datei_ohne_suffix");`

4.3.1.4. Fenster, das andere Fenster aufruft, als oberstes also sichtbar anzeigen

notwendig, wenn Dokument andere Fenster aufruft, die schneller geladen werden als das Dokument

```
<BODY ... onLoad="if(window.focus) window.focus()">
```

4.3.2. Fenster schliessen

4.3.2.1. Fenster schließen, das nicht FRAME ist

Beispiel 1

```

<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JavaScript">
        <!--
            function FensterOeffnen()
            { neuesFenster = open("http://www.test.de"); }

            function FensterSchliessen()
            { neuesFenster.close(); }

        <!-->
        </SCRIPT>
    </HEAD>

    <BODY>
        <FORM>
            <INPUT TYPE="button"
                VALUE="Oeffnen"
                onClick="FensterOeffnen()">

            <INPUT TYPE="button"
                VALUE="Schliessen"
                onClick="FensterSchliessen()">

        </FORM>
    </BODY>
</HTML>

```



```

</BODY>
</HTML>

```

Beispiel 2

```

<A HREF="javascript:self.close()">Schlie&szlig;en</A>

```

4.3.2.2. Fenster, das Frame ist: Oberstes Fenster schliessen

```

<A HREF="javascript:top.close()">Schlie&szlig;en</A>

```

4.3.2.3. Fenster schliesst sich selbst nach Wartezeit (Popup per open())

```

</HTML>
</HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function fenster_offnen_und_schliessen()
    {
        var fenster;
        fenster=window.open("", "Fenster", "width=180,height=100");
        fenster.document.write("<H1>Ich schlie&szlig;e mich nach 4 Sekunden</H1>");
        fenster.setTimeout('window.close()',4000);
    }
    //-->
</SCRIPT>
</HEAD>
<BODY onload=" fenster_offnen_und_schliessen()">
</BODY>
</HTML>

```

4.3.3. Fensterauflösung ändern

```

javascript:window.resizeTo(640,480); javascript:window.resizeTo(800,600); javascript:window.resizeTo(1024,768)

```

Beispiel: sich aufblasendes Fenster von 100x100 bis auf 640x480

Hinweis: Fenster mit Dimensionen kleiner 100x100 Pixel sind NUR mit signiertem Script erzeugbar !!

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
    var start_hoehe=100;
    var start_breite=100;
    var max_hoehe=480;
    var max_breite=640;
    var aktuelle_hoehe=start_hoehe;
    var aktuelle_breite=start_breite;
    var y=5;
    var TimerID=null;
    var fenster;

    function start()
    {
        fenster=window.open("", "", "scrollbars");

        if ( document.layers || document.all)
        {
            fenster.resizeTo(start_breite,start_hoehe);
            fenster.moveTo(0,0);
            blasen();
        }
        else
        {alert("Weder Netscape noch Microsoft erkannt !")};
    }
    function blasen()
    {
        if (aktuelle_hoehe>=max_hoehe)
        {x=0;}

        fenster.resizeBy(5,y);
        aktuelle_hoehe+=5;

```




```

        aktuelle_breite+=5;

        if (aktuelle_breite>=max_breite)
        {
            alert("Maximal aufgeblasen !");
            fenster.close();
            x=5;
            TimerID=null;
        }
        else
        {TimerID=setTimeout("blasen()",50);}
    }
//-->
</SCRIPT>
</HEAD>
<BODY>
<A      HREF="javascript:start()"
        onMouseOver="javascript:window.status='Oeffne Fenster';return true;" // Text nach Statuszeile
        onMouseout="javascript:window.status=";" // Statuszeile löschen
>Oeffne NEUES Fenster mit 100x100 Pixel und blase es auf 640x480 Pixel !
</A>
</BODY>
</HTML>

```

4.3.4. Fenstertitel nachträglich ändern

Beispiel: permanenter Fenstertitel-Wechsel

funktioniert nicht mit jedem Browser --> ausprobieren !

```

<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
<!--
    var titel_texte_feld = [

                                "Titel 1",
                                "Titel 2",
                                "Titel 3",
                                "Titel 4"

                                ];

    var wechsel_tempo = 2500; // Zeit in ms zwischen zwei Schritten
    var zahler = 0;

    function start()
    {
        titel_texte_feld[titel_texte_feld.length] = document.title;
        // beim ersten Aufruf wird der Original-Titel mit gespeichert
        // Länge ab 1, aber Index ab 0
        // Länge == Index +1, an desssen Position der Original-Titel
        // abgelegt wird
        window.setTimeout("wechselln()", wechsel_tempo);
    }

    function wechselln()
    {
        document.title = titel_texte_feld [zahler];
        zahler ++;
        if(zahler >= texte.length)
        {zahler = 0}

        id = window.setTimeout("wechselln()", wechsel_tempo);
    }

//-->
</SCRIPT>

```

4.4. Fenstertitelzeile mit scrollendem Text

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
//

```



```
//      In der aktuellen Fenster-Titelzeile einen freien Text durch endloses Scrollen von links nach rechts vorsetzen
//
//*****
//
//      nachfolgende Variablen sind vom Programmierer zu setzen
//
//*****
var VorsetzText_Wert="Freien Text vorsetzen durch scrollen und warten ";
var ScrollGeschwindigkeit="150";      // in Millisekunden, immer > 0, wird nicht geprüft
var DummyLeerzeichenAnzahl=10;      // immer > 0, wird nicht geprüft
// Die Leerzeichen werden nicht sichtbar angezeigt, aber gescrollt.
//      So entsteht eine Wartezeit (pro Leerzeichen laut
//      ScrollGeschwindigkeit)
//      für die vollständige Anzeige von VorsetzText NACH dem
//      vollständigem Vorsetzen per Scrollen von links nach
//      rechts, ehe der Scroller wieder von vorn beginnt

//
//*****
//
//      nachfolgender Code darf nicht verändert werden
//      wird anstelle von onLoad innerhalb BODY verwendet
//
//*****
//
//      rechtsbündige Auffüllung mit DummyLeerzeichen
for (i=0; i <=DummyLeerzeichenAnzahl; i++)
{ VorsetzText_Wert+=" ";}
//
//      Länge NACH Auffüllung ermitteln
var VorsetzText_Laenge=VorsetzText_Wert.length;
//
//      globale Variablen für EndlosScrollen(), die dort laufend manipuliert werden,
//      also hier initialisiert werden müssen
var Zahler="0";
var VorsetzText="";
//-->
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript1.2">
//*****
//
//      nachfolgender Code darf nicht verändert werden
//
//*****
function EndlosScrollen()
{
    // endloses Scrollen

    if (
        (document.all)
        || (document.getElementById)
    )
    {
        // VorsetzText um nächsten Buchstaben aus dem VorsetzText_Wert erweitern
        // (einschliesslich DummyLeerzeichen)
        VorsetzText+=VorsetzText_Wert.charAt(Zahler); // charAt ab Null

        // Fenster-Titelzeile neu belegen
        document.title=VorsetzText;

        Zahler++;
        if(Zahler==VorsetzText_Laenge)
        {
            Zahler="0";
            VorsetzText="";
        }
        setTimeout("EndlosScrollen()",ScrollGeschwindigkeit);
    }
}
```



```

window.onload=EndlosScrollen;
    // Achtung ohne () kodieren, damit nicht sofort sondern beim Laden ausgeführt wird
//-->
</SCRIPT>
</BODY>
</HTML>

```

5. Statuszeile

5.1. Standardtext in der Statuszeile

```
<BODY onLoad="window.defaultStatus='standard_text';">
```

5.2. Text Buchstabe für Buchstabe in Statuszeile anzeigen

Der aktuelle Statuszeilentext wird als Teilkette von Position 0 bis zeichen_nr angezeigt, wobei zeichen_nr pro Anzeige um 1 erhöht wird

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        var statuszeile = new Array();
        statuszeile[0] = "Text0";
        statuszeile[1] = "Text1";
        statuszeile[2] = "Text2";

        var statuszeile_nr = 0; var zeichen_nr = 0;

        function textanzeigen ()
        {
            if (position < statuszeile[statuszeile_nr].length) // Länge ab 1
            {
                // nächste Teilkette des aktuellen Statuszeilentextes anzeigen
                window.status = statuszeile[statuszeile_nr].substring(0, zeichen_nr);

                // nächste Position
                position++;
                setTimeout("textanzeigen()", 100);
            }
            else
            {
                // aktuelle Statuszeilentext komplett anzeigen
                window.status = statuszeile[statuszeile_nr];
                // nächste Statuszeile adressieren
                statuszeile_nr++;
                if (statuszeile_nr >= statuszeile.length) { statuszeile_nr = 0; }
                // Position 0 einstellen
                zeichen_nr = 0;
                // Start der buchstabenweise Anzeige
                setTimeout("textanzeigen()", 1000);
            }
        }
      // -->
    </SCRIPT>
  </HEAD>
  <BODY onLoad="textanzeigen()">
  </BODY>
</HTML>

```

5.3. Automatisch wechselnder Text in der Statuszeile

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        var statuszeile = new Array();
        statuszeile[0] = "Text0";
        statuszeile[1] = "Text1";
        statuszeile[2] = "Text2";

        var statuszeile_nr = 0;

        function textanzeigen()
        {

```



```

        window.status = statuszeile[statuszeile_nr];
        statuszeile_nr++;
        if (statuszeile_nr >= statuszeile.length) {statuszeile_nr = 0;}
        setTimeout("textanzeigen()", 1000);
    }
    // -->
</SCRIPT>
</HEAD>

<BODY onLoad="textanzeigen()">
</BODY>
</HTML>

```

5.4. Scroller in der Statuszeile

Beispiel 1 Dauerhaftes Scrollen eines Textes

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var scrolltext_gesamt = "Dieser Text scrollt!";
    var scrolltext = "";
    var zahler = scrolltext_gesamt.length;

    function scrollen()
    {
        if (zahler == scrolltext_gesamt.length)
        {
            zahler = 0;
            scrolltext = scrolltext_gesamt;
        }
        else
        {
            zahler++; // 1 bis scrolltext_gesamt.length
            // Blank vorsetzen, also Kette um 1 Zeichen verlängern
            scrolltext = " "+scrolltext;
            // rausgerutschtes Zeichen abschneiden
            scrolltext = scrolltext.substring(0, scrolltext_gesamt.length -1);
            // Angaben ab 0
        }
    }

    window.status = scrolltext;
    setTimeout("scrollen()", 100);
}
// -->
</SCRIPT>
</HEAD>

<BODY onLoad="scrollen()">
</BODY>
</HTML>

```

Beispiel 2 Scroller, der angehalten wird, wenn sich der Mauszeiger über einen Link bewegt

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var scrolltext_gesamt = "Dieser Text scrollt!";
    var scrolltext = "";
    var zahler = scrolltext_gesamt.length;

    function scrollen()
    {
        if (zahler >= scrolltext_gesamt.length)
        {
            zahler = 0;
            scrolltext = scrolltext_gesamt;
        }
        else
        {
            zahler++; // 1 bis scrolltext_gesamt.length
            // Blank vorsetzen, also Kette um 1 Zeichen verlängern

```



```

        scrolltext = " "+scrolltext;
        // rausgerutschtes Zeichen abschneiden
        scrolltext = scrolltext.substring(0, scrolltext_gesamt.length -1);
        // Angaben ab 0
    }

    window.status = scrolltext;
    id=setTimeout("scrollen()", 100);
}

function scrollen_stoppen()
{clearTimeout(id)}

// -->
</SCRIPT>
</HEAD>

<BODY onLoad="scrollen()">
    Bewegen Sie den Mauspfel &uuml;ber diesen
    <A   HREF="http://www.test.de"
        onMouseOver="scrollen_stoppen()"
        onMouseOut="scrollen()">
        Link</A>
</BODY>
</HTML>

```

5.5. Text zu einem Hyperlink (HREF) für eine feste Zeitspanne in der Statuszeile anzeigen

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var id;
    var anzeige_aktiv = false;
    var anzeige_zeit = 3000;           // 3000 Millisekunden = 3 Sekunden

    function anzeige_starten(href_text)
    {
        if(anzeige_aktiv)
        {clearTimeout(id);}

        window.status = href_text;

        id = setTimeout("anzeige_stoppen()", anzeige_zeit);

        anzeige_aktiv = true;

        return true;                // Wichtig !!!
    }

    function anzeige_stoppen ()
    {
        anzeige_aktiv = false;

        window.status = "";
    }
//-->
</SCRIPT>
</HEAD>

<BODY>
    <A HREF= ... onMouseOver="javascript:return anzeige_starten('Hinweistext...')">
    ...
    </A>
</BODY>
</HTML>

```



5.6. *Text zu einem Hyperlink (HREF) für Überfahren mit Maus in der Statuszeile anzeigen bzw. mit fester Zeitspanne*

```
<HTML>
  </HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        var id;
        var anzeige_aktiv = false;
        var anzeige_zeit = 3000;      // 3000 Millisekunden = 3 Sekunden

        function anzeige_starten(href_text)
        {
          if(anzeige_aktiv)
            {clearTimeout(id);}

          window.status = href_text;

          id = setTimeout("anzeige_stoppen()",anzeige_zeit);

          anzeige_aktiv = true;

          return true;      // Wichtig !!!
        }

        function anzeige_stoppen ()
        {
          if(anzeige_aktiv)
          {
            clearTimeout(id);
            anzeige_aktiv = false;
          }

          window.status = "";
        }
      //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    <A      HREF= .....
      onMouseOver="javascript:return anzeige_starten('Hinweistext...');"
      onMouseOut="anzeige_stoppen();">
      .....
    </A>
  </BODY>
</HTML>
```

5.7. *Hinweise zu Formularfelder in Statuszeile anzeigen*

```
<HTML>
  </HEAD>
    <SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
      <!--
        function hinweis_anzeigen(hinweis_text)
        {window.status = hinweis_text;}

        function hinweis_loeschen()
        {window.status = "";}
      // -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT  TYPE=TEXT
        onFocus="hinweis_anzeigen('Hinweis');"
        onBlur="hinweis_loeschen();">
    </FORM>
  </BODY>
```



</HTML>

5.8. Text in Statuszeile in fester Zeitspanne blinkend anzeigen

<HTML>

</HEAD>

```

<SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
<!--      var zeitspanne = 6000;           // 6 Sekunden blinken lassen, danach
                                                //      Statuszeile löschen
      var anzeigezeit  = 500;              // 0,5 Sekunden warten nach Setzen
                                                //      bzw. Löschen der Statuszeile
      var id_blinken  = null; // muss auf null initialisiert sein !!

      function blinken_timer_loeschen
      {
          if (id_blinken != null)
          {
              clearTimeout(id_blinken); // blinken stoppen falls aktiv
              id_blinken = null;
          }
      }

      function blinken_stop()
      {
          blinken_timer_loeschen;
          window.status="";
      }

      function blinken_start(status_text)
      {
          blinken_timer_loeschen;
          blinken(true, status_text);      // Blinken anstossen für
                                                // paralleles Arbeiten per Timer

          setTimeout("blinken_stop()",zeitspanne); // warten und danach blinken stoppen
      }

      function blinken(ein_aus, text)
      {
          if (ein_aus)
          {
              window.status = text;
              ein_aus=false;      // im nächsten Aufruf die Statuszeile löschen
          }
          else
          {
              window.status = "";
              ein_aus=true;      // im nächsten Aufruf die Stautuszeile setzen
          }
          id_blinken = setTimeout("blinken(" + ein_aus + ")", anzeigezeit)
          // nächsten Aufruf nach Ablauf der anzeigezeit starten
      }
      //-->
</SCRIPT>

```

</HEAD>

<BODY ... onLoad="blinken_start('Ich blinke !')">

</BODY>

</HTML>

6. FRAMES**6.1. FRAMES füllen**

Beispiel Zeigt die Verwendung von FRAMES. In die drei erzeugten FRAMES werden die Dateien <FRAME1.htm>, <FRAME2.htm> und <FRAME3.htm> laden

<HTML>

```

<FRAMESET ROWS="50%,50%">
  < FRAMESET COLS="50%,50%">
    <FRAME SRC="FRAME1.htm"
      NAME="FRAME1">
    < FRAME
      SRC="FRAME2.htm"
      NAME="FRAME2">
  </ FRAMESET >
  < FRAME
    SRC="FRAME3.htm"
    NAME="FRAME3">
</ FRAMESET >

```

</HTML>



6.2. Reload eines Dokumentes mit allen seinen FRAMES

```
function frame_neu_laden()
{
    for (var i=0; i<windows.FRAMES.length; i++)
    { windows.frames[i].location.reload(false); }
}
```

<FRAMESET onResize="frame_neu_laden()">

6.3. Datei ohne FRAMESET in eine Dateimit FRAMESET laden

```
if (self.location == top.location)
{ location.HREF="/FRAMESET.html?" + escape(location.pathname); }
```

6.4. Werbefenster als FRAME alle 3 Sekunden einblenden

werbung1.htm bis Werbung3.htm sind die Werbedateien,
die als FRAME innerhalb von **WERBUNG.HTM** angezeigt werden
wobei 1 bis 3 als Index zur Adressierung des Werbungsdocumentes verwendet werden
WERBUNG.HTM ist die Startseite der Werbung und erscheint im werbe_frame
werbung1.htm bis Werbung3.htm wechseln alle 3 Sekunden

FRAME.htm ist das Fenster,
das mit Start von **INDEX.HTML** geöffnet wird
Größe: 200x200 Pixel, in Größe nicht änderbar
angezeigt an Position 50,50 im Browserfenster
in dem die FRAMEs gebildet werden

auswahl1.htm bis auswahl3.htm sind die in **MENU.HTM** auswählbaren Dokumente
MENU.HTM wird im Menu FRAME angezeigt
auswahl1.htm wird als normaler Link kodiert
auswahl2.htm bis auswahl3.htm werden im Auswahl_frame angezeigt

Haupt-Datei INDEX.HTM

```
<HTML>
<HEAD>
    <TITLE> Index.html </TITLE>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        index_der_werbedatei=1;

        function oeffne_werbefenster_als_frame()
        {
            start_werbefenster=window.open(
                "WERBUNG.HTM",
                "StartWerbeFenster",
                "resizeable=no, WIDTH=200, HEIGHT=200"
            );
            start_werbefenster.moveTo(50,50);
        }

        function wechsele_werbung() // rekursive Funktion !!
        {
            // index_der_werbedatei beginnt ab 1
            aktuelle_werbung='werbung' + index_der_werbedatei + '.htm';
            self.auesserer_frame.location.HREF=aktuelle_werbung;
            // und im aeußeren FRAME anzeigen
            index_der_werbedatei++; // Index der nächsten Werbedatei
            if (index_der_werbedatei > 3) { index_der_werbedatei = 1; }
            // wenn 4 so wieder 1
            setTimeout("wechsele_werbung()", 3000); // alle 3 sec die Werbedatei wechseln
        }
    <!-->
    </SCRIPT>
</HEAD>

<FRAMESET ROWS="100,*"
    onLoad="setTimeout('wechsele_werbung()',3000; oeffne_werbefenster_als_frame())">
    <FRAME NAME="werbe_frame"
        SRC=" WERBUNG.HTM "
        TITLE="Werbung.htm">
    <FRAMESET COLS="30,*">
```




```

        <FRAME NAME="menu_frame"
            SRC="MENU.HTM"
            TITLE="Auswahl">
        <FRAME NAME="auswahl_frame"
            SRC="auswahl1.htm"
            TITLE="Auswahl1">
    </FRAMESET>
</NOFRAME>
    <BODY>
        <H1>Link fuer framelose Darstellung</H1>
    </BODY>
</NOFRAME>
</FRAMESET>
</HTML>

```

Datei MENU.HTM

dient nur der Auswahl von auswahl1.htm bist auswahl3.htm

```

<HTML>
  <HEAD>
    <TITLE>.....</TITLE>
    <BASE TARGET="auswahl_frame"> // jede Anzeige erfolgt in diesem FRAME
  </HEAD>

  <BODY>
    <H2>Auswahl</H2>
    <A HREF="auswahl1.htm">Auswahl1</A></BR>
    <A HREF="auswahl2.htm">Auswahl2</A></BR>
    <A HREF="auswahl3.htm">Auswahl3</A></BR>
    <HR>
    <FORM ACTION="">
      <INPUT TYPE="button"
        VALUE="Auswahl 2 im Elternframe anzeigen"
        onClick="parent.auswahl_frame.location.HREF='auswahl2.htm'"
      >
      <INPUT TYPE="button"
        VALUE="Auswahl 1 mit Frameauflösung anzeigen"
        onClick="top.location.HREF='auswahl1.htm'"
      >
    </FORM>
  </BODY>
</HTML>

```

Datei WERBUNG.HTM

dient nur dem Start der Werbung

```

<HTML>
  <HEAD>
    <TITLE>.....</TITLE>
  </HEAD>

  <BODY>
    Werbung
    <FORM ACTION="">
      Klicken fuer Werbung starten
      <BR>
      <INPUT TYPE="button"
        VALUE="Starte durch Klick !"
        onClick="opener.top.location.HREF='werbung0.htm'; self.close()"
      >
    </FORM>
  </BODY>
</HTML>

```

6.5. HTML-Inhalte von FRAMES tauschen**Beispiel 1 Inhalte zweier Frames tauschen**

Kodierung im Dokument, dass die beiden Frames definiert !

```
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
```



```
<!--  
    function rahmentausch(logischer_name_rahmen1,html_datei1,  
                          logischer_name_rahmen2,html_datei2)  
{  
    frames[logischer_name_rahmen1].location.href = html_datei1;  
    frames[logischer_name_rahmen2].location.href = html_datei2;  
}  
// -->  
</SCRIPT>
```

Anwendung:

```
<A HREF="javascript:parent.rahmentausch('r1', 'a.html', 'r2',b.html)">tauschen</A>
```



Beispiel 2 *Inhalte beliebig vieler Frames als Ring tauschen*

Die logischen Framennamen werden als variable Argumenteliste übergeben und dienen der Indizierung der Frame im Dokument. Argumententrenner sind Doppelpunkte.

Tausch:

```

    erster Frame retten und dann mit zweiten Frame überschreiben
    zweiten Frame mit drittem Frame überschreiben
    ....
    vorletzten Frame mit letztem Frame überschreiben
    letzten Frame mit geretteten ersten Frame überschreiben
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
<!--
function rahmentausch()
{
    if (arguments.length>1)        // Länge ab 1, mindestens 2 Argumente
    {
        var pos_doppelpunkt = arguments[0].indexOf(":");
        if(pos_doppelpunkt != -1)  // nächstes Argument ist vorhanden
        {
            var rette_logischer_framename= arguments[0].substring(0, pos_doppelpunkt);

            for(var i = 0; i < arguments.length; i++)    // Index ab 0
            {
                pos_doppelpunkt = arguments[i].indexOf(":");
                if(pos_doppelpunkt != -1)  // nächstes Argument ist vorhanden
                {
                    // ersten zu ersetzenden Framennamen retten
                    if (i==0)
                    {var rette_logischer_framename=
                        arguments[i].substring(0, pos_doppelpunkt)
                    }

                    // tauschen der logischen Framennamen
                    logischer_framename_zu_ersetzender_frame=
                        arguments[i].substring(0, pos_doppelpunkt);

                    logischer_framename_ersetzender_frame=
                        arguments[i].substring(0, pos_doppelpunkt + 1);

                    frames[logischer_framename_zu_ersetzender_frame].location.href =
                        logischer_framename_ersetzender_frame;
                }
            }

            frames[logischer_framename_ersetzender_frame].location.href=
                rette_logischer_framename;
        }
    }
}
// -->
</SCRIPT>

```

Anwendung:

```
<A HREF="javascript:parent.rahmentausch('r1:a.html', 'r2:b.html', 'r3:c.html')">tauschen</A>
```

6.6. Ein FRAME für mehrere HTML-Dokumente vom gemeinsamen Server

```

<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
<!--
    function lade_dokument_in_rahmen(html_datei)
    {
        top.location.href = html_datei;
    }

    if(top.location.href == self.location.href)    // wenn Server identisch
    { lade_dokument_in_rahmen ("test.htm");}
// -->
</SCRIPT>

```



7. **Formulare**

7.1. **Werte auslesen**

7.1.1. **Auslesen eines Eingabefeldes**

Beispiel 1

```
<HTML>
  <HEAD>
    <TITLE>Formulare</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function gruss()
        {alert("Hallo " + document.meinFormular.Eingabe.value + "!!");}
      // -->
    </SCRIPT>
  </HEAD>

  <BODY>
    Bitte Namen eingeben und danach den Knopf drücken;
    <FORM NAME="meinFormular">
      <INPUT TYPE="text"
        NAME="Eingabe"
        VALUE=""><BR>
      <INPUT TYPE="button"
        NAME="Knopf"
        VALUE="Bitte drücken"
        onClick="gruss()">
    </FORM>
  </BODY>
</HTML>
```

Beispiel 2

```
<HTML>
  <HEAD>
    <TITLE>Formulare</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function gruss(Text)
        { alert("Hallo " + Text + "!!");}
      // -->
    </SCRIPT>
  </HEAD>

  <BODY>
    Bitte Namen eingeben und danach den Knopf drücken;
    <FORM NAME="meinFormular">
      INPUT TYPE="text"
        NAME="Eingabe"
        VALUE=""><BR>
      <INPUT TYPE="button"
        NAME="Knopf"
        VALUE="Bitte drücken"
        onClick="gruss(this.form.Eingabe.value)">
    </FORM>
  </BODY>
</HTML>
```

7.1.2. **Auslesen einer Checkbox**

```
<HTML>
  <HEAD>
    <TITLE>Formulare</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function test()
        {
          if (document.meinFormular.Checkbox.checked)
            {alert("Das Feld ist markiert!!")}
```



```

        else
        {alert("Das Feld ist nicht markiert!");}
    }
    // -->
</SCRIPT>
</HEAD>

<BODY>
    <FORM NAME="meinFormular">
        <INPUT TYPE="checkbox"
            NAME="Checkbox"><BR>
        <INPUT TYPE="button"
            NAME="Knopf"
            VALUE="Bitte dr&uuml;cken"
            onClick="test()">

    </FORM>
</BODY>
</HTML>

```

7.1.3. Auslesen einer Auswahlliste

7.1.3.1. Liste mit einer Auswahlmöglichkeit

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function ausgabe(auswahl)
        {
            var text = "Index: ";
            text = text + auswahl.selectedIndex;
            text = text + "\n";           // Zeilenumbruch
            text = text + "Text: ";
            text = text + auswahl[auswahl.selectedIndex].text;
            alert(text);
        }
    // -->
    </SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <SELECT NAME="liste">
            <OPTION>element 0
            <OPTION>element 1
            <OPTION>element 2
            <OPTION>element 3
            <OPTION>element 4
        </SELECT>
        <INPUT TYPE="button"
            VALUE="Bitte dr&uuml;cken"
            onClick="ausgabe(this.form.liste)">

    </FORM>
</BODY>
</HTML>

```

7.1.3.2. Liste mit mehreren Auswahlmöglichkeiten

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function ausgabe(auswahl)
        {
            var text = "";

            for (var i = 0; i < auswahl.length; i++)
            {
                if (auswahl[i].selected)
                {
                    text += "Index: "; // identisch mit text = text + "Index :";
                    text += i;

```



```

        text += "\n";           // Zeilenumbruch
        text += "Text: ";
        text += auswahl[i].text;
        text += "\n";
    }
    alert(text);
}
// -->
</SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <SELECT NAME="liste" MULTIPLE>
            <OPTION>element 0
            <OPTION>element 1
            <OPTION>element 2
            <OPTION>element 3
            <OPTION>element 4
        </SELECT>
        <INPUT TYPE="button"
            VALUE="Bitte dr&uuml;cken"
            onClick="ausgabe(this.form.liste)">
    </FORM>
</BODY>
</HTML>

```

7.2. Werte zuweisen

Beispiel: Wert einem Eingabefeld zuweisen

```

<HTML>
<HEAD>
    <TITLE>Formulare</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function umrechnung(celsius)
        {
            var fahrenheit;
            fahrenheit = 9 / 5 * celsius + 32;
            document.Formular.Ausgabe.value = fahrenheit;
        }
    // -->
    </SCRIPT>
</HEAD>

<BODY>
    Bitte geben Sie einen Temperaturwert in Celsius ein:
    <FORM NAME="Formular">
        <INPUT TYPE="text"
            NAME="Eingabe"
            VALUE=""><BR>
        <INPUT TYPE="button"
            NAME="Knopf"
            VALUE="Berechnung"
            onClick="umrechnung(this.form.Eingabe.value)"><BR>
        Entpricht
        <INPUT TYPE="text"
            NAME="Ausgabe"
            VALUE=""> Fahrenheit<BR>
    </FORM>
</BODY>
</HTML>

```

7.3. Werte überprüfen

Hinweis: false entspricht 0, alles andere ist true

7.3.1. prüfen auf Leerkette

function AufLeerkettePruefen(zeichenkette)



```

{
    if zeichenkette.length == 0)
    { return false}
    else {return true}
}

```

7.3.2. prüfen auf Leerzeichen

```

function aufLeerzeichenPruefen(zeichenkette)
{
    for (var i=0; i<zeichenkette.length; i++)
    {
        if (zeichenkette.charAt(i) == " ")
        {return true}
    }
    return false
}

```

7.3.3. auf vorgegebene Zeichenanzahl prüfen

```

function AufZeicheanzahlPruefen(zeichenkette,anzahl)
{
    if(zeichenkette.lengt == anzahl)
    {return true}
    else {return false}
}

```

7.3.4. auf Ziffern prüfen

```

function AufZiffernPruefen(zeichenkette)
{
    for (var i=0; i<zeichenkette.length; i++)
    {
        aktZeichen= zeichenkette.charAt(i);
        if ( (aktZeichen < "0") || (aktZeichen > "9"))
        {return false}
    }
    return true
}

```

7.3.5. auf Integerwert aus einem Gültigkeitsbereich prüfen

```

function AufIntwertPruefen(Zeichenkette,int_bereich_beginn,int_bereich_ende)
{
    var zahl=parseInt(zeichenkette);
    return ((zahl >= int_bereich_beginn) && (zahl <= int_bereich_ende));
}

```

7.4. Werte konvertieren

7.4.1. Numerischen Wert als Zeichenkette liefern und dabei Dezimalpunkt zu Dezimalkomma umwandeln

Es wird angenommen, dass genau ein Dezimalpunkt vorkommt.

```

function punkt_zu_komma (numerischer_wert)
{
    var zeichenkette = numerischer_wert.toString();
    var funktionswert=zeichenkette;
    var pos_punkt = zeichenkette.indexOf(".");
    if(pos_punkt >=0)
    {
        funktionswert =      zeichenkette.substring(0, pos_punkt)
                           + ","
                           + zeichenkette.substring(pos_punkt + 1, zeichenkette.length);
    }
    return zeichenkette;
}

```

7.4.2. Ziffern-Zeichenkettenwert nach numerisch und dabei Dezimalkomma zu Dezimalpunkt umwandeln

Es wird angenommen, dass genau ein Komma vorkommt.

```

function punkt_zu_komma(zeichenkette)
{
    var pos_komma = zeichenkette.indexOf(",");
    var funktionswert;

    if(pos_komma == -1)
    {
        if (zeichenkette.indexOf(".") == -1)

```



```

        { funktionswert = parseInt(zeichenkette)}
        else
        { funktionswert = parseFloat(zeichenkette)}
    }
    else
    {
        funktionswert = parseFloat(   zeichenkette.substring(0, pos_komma)
                                     + ","
                                     + zeichenkette.substring(pos_komma + 1, zeichenkette.length)
                                     );
    }

    return funktionswert;
}

```

7.4.3. Vornull bei Ziffern-Zeichenkette, die numerisch <1 bzw. > -1 ist, vor dem Dezimalkomma bzw. Dezimalpunkt oder dem Minuszeichen erzeugen

```

function vornull(zeichenkette)
{
    var funktionswert=zeichenkette;

    if(      (zeichenkette.substring(0, 1) == ";")
        ||   (zeichenkette.substring(0, 1) == ".")
        )
    {funktionswert = "0" + zeichenkette }

    if(      (zeichenkette.substring(0, 2) == "-.")
        ||   (zeichenkette.substring(0, 2) == "-.")
        )
    { funktionswert = "-0" + zeichenkette.substring(1)}

    return funktionswert;
}

```

7.5. Formularprüfung für Submit

Es wird eine Email-Adresse über Tastatur eingegeben und danach geprüft,
ob der Klammeraffe überhaupt auftaucht
und wenn ja, ob nur genau einmal

```

<FORM
NAME="logischer_form_name"
onSubmit="return submit_pruefen();" // () nicht vergessen, damit Funktion ausführbar
>
E-Mail eingeben
<INPUT
    TYPE=TEXT
    NAME="logischer_text_name"
>
</FORM>

<SCRIPT ...>
<!--
    function submit_pruefen()
    {
        var email=document.logischer_form_name.logischer_text_name.value;
        // value enthält die Eingabe
        var return_wert=false; //Annahme: Klammeraffe      nicht vorhanden
                                                oder mehrfach vorhanden

        if (email != "")
        {
            // Eingabe erfolgt
            var klammer_affen_position=email.indexOf("@");
                                                // Klammeraffe suchen
                                                // erstes Auftreten

            if (klammer_affen_position != -1)
            { klammer_affen_position=email.lastIndexOf("@");
                                                // Klammeraffe 1. Mal gefunden, also
                                                // nach letztem Auftreten suchen
            }
        }
    }
}

```




```

        if (klammer_affen_position===-1)
        {return_wert=true;}
    }
    return return_wert;
}
// -->
</SCRIPT>

```

7.6. Laufschrift in einem Formular

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        text="Laufschrift";

        function anzeigen()
        {
            teilkette=text.substring(0, 1);
            text=text.substring(1, text.length);
            text= text + teilkette;

            window.document.formular.schrift.value=text;
            setTimeout('anzeigen()',200);
        }
    <!-->
    </SCRIPT>
</HEAD>

<BODY>
    <FORM NAME="formular">
        <INPUT TYPE=button
            VALUE="Laufschrift im Formularfeld"
            onClick="anzeigen()">
        <INPUT TYPE="text"
            NAME="schrift"
            SIZE="20"
            READONLY>
    </FORM>
</BODY>
</HTML>

```

8. Menü

8.1. Menüleiste

8.1.1. Realisierung einer Menüleiste mittels FRAMES

In die beiden FRAMES werden die Dateien <ind.htm> und <menu1.htm> geladen.

Menu-Dokument:

```

<HTML>
<HEAD>
    <TITLE>Menue</TITLE>
</HEAD>
<FRAMESET ROWS="90%,10%">
    <FRAME SRC="ind.htm"
        NAME="haupt">
    <FRAME SRC="menu1.htm"
        NAME="menu">
</FRAMESET>
</HTML>

```

8.1.2. Realisierung einer Menüleiste in einem separaten Fenster

In das neu erzeugte Fenster wird die Datei <panel.htm> geladen.

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">

```



```

        <!--
            function fensterErzeugen()
            {neu = open("panel.htm", "navigation", "WIDTH=160,HEIGHT=180");}
        // -->
    </SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <INPUT TYPE="button"
            VALUE="Bitte drücken"
            onClick="fensterErzeugen()">
    </FORM>
</BODY>
</HTML>

```

8.2. Formular-Text-Pulldown-Menü mit Button ein und ausschalten

```

<HTML>
    <HEAD>
        <TITLE>Menu test</TITLE>
        <SCRIPT type="text/jscript" language="jscript">
            <!--
                var isVisible = false;

                function menu_verwalten()
                {
                    if(isVisible)
                    {
                        sichtbarkeit = "hidden";
                        isVisible = false;
                    }
                    else
                    {
                        sichtbarkeit = "visible";
                        isVisible = true;
                    }
                    document.all["test_menu"].style.overflow = sichtbarkeit;
                }
            //-->
        </SCRIPT>
    </HEAD>

    <BODY>
        <FORM>
            <INPUT TYPE="button"
                VALUE="Menu verwalten"
                onClick="menu_verwalten()">
            <DIV
                ID="test_menu"
                STYLE="height: 0; overflow: hidden;">
                Menu Eintrag #1<BR>
                Menu Eintrag #2<BR>
                Menu Eintrag #3<BR>
            </DIV>
        </FORM>
    </BODY>
</HTML>

```

8.3. Formular-Text-Pulldown-Menü mit Aktivierung der Menüpunkte

Die Menüpunkte müssen HTML-Elemente oder Urls sein.

```

<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JavaScript">
            <!--
                function menuepunkt_aktivieren(logischer_formular_name)
                {
                    var index= logischer_formular_name. auswahl.selectedIndex
                    if (logischer_formular_name. auswahl.options[index] != 0)
                    {

```



```

        logischer_formular_name. auswahl.selectedIndex = 0;
        window.location=
            logischer_formular_name. auswahl.options[index].value;
        document.forms[0].reset();
        document.forms[0].elements[0].blur();
    }
    }
    </SCRIPT>
</HEAD>

<BODY>
    <FORM NAME="formular">
        <SELECT NAME="auswahl"
            onChange="menuepunkt_aktivieren(formular)"
            SIZE=1>
            <OPTION SELECTED VALUE=0> --- Waehle aus ! ---
            <OPTION VALUE="test1.html">HTML-Dokument Test1
            <OPTION VALUE="test2.html">HTML-Dokument Test2
            <OPTION VALUE="http://www.test.de /index.html">www.test.de
        </SELECT>
    </FORM>
</BODY>
</HTML>

```

8.4. Formular-Option-Menü aus Hyperlinks

```

<HEAD>
    <SCRIPT ...>
    <!-- // globale Größen festlegen
    function waehle_hyper_link(feld)
    {
        var ausgewaehlte_url=feld.options[feld.selectedIndex].value;
        window.top.location=ausgewaehlte_url;
    }
    // -->
    </SCRIPT>
</HEAD>

....
<BODY>
    <FORM NAME="formular">
        <SELECT NAME="logischer_select_name"
            onChange="waehle_hyper_link(this);"
            >
            <OPTION VALUE=http://www.test1.de"test1</OPTION>
            <OPTION VALUE=http://www.test2.de"test2</OPTION>
        </SELECT>
    </FORM>
</BODY>

```

9. Buttons

9.1. Aussehen eine Button mit Mausklick verändern

Zwei Mauszustände: gedrückt und ungedrückt, also zwei verschiedene GIF-Dateien für das Button nötig

Annahmen: Button ist die erste Grafik auf der Seite, also Index=0 (Index immer ab 0 zählen!)
Standard ist die GIF-Datei für das ungedrückte Button

Variante 1

```

<HTML>
    <HEAD>
        <TITLE> ..... <TITLE>
        <SCRIPT LANGUAGE="JavaScript">
        <!--
            button_ungedruickt=new Image();
            button_ungedruickt.SRC="ungedr.gif";
            button_gedruickt=new Image();
            button_gedruickt.SRC="gedr.gif";

```



```

        function ungedrueckt()
        { document.images[1].SRC=button_ungedrueckt.src;}

        function gedrueckt()
        { document.images[1].SRC=button_gedrueckt.src;}

    //-->
    </SCRIPT>
</HEAD>

<BODY .....>
    .....
    <A      HREF="#"
        onMouseDown="gedrueckt();"
        onMouseUp="ungedrueckt();"

        <IMG      SRC="ungedr.gif"
            BOLDER=0
            HSPACE=abstand_zum_alternativen_text
        >
        alternativer_text
    </A>
</BODY>
</HTML>

```

Variante 2

```

<HTML>
    <HEAD>
        <TITLE> ..... <TITLE>
        <SCRIPT LANGUAGE="JavaScript">
            <!--
                button_ungedrueckt=new Image();
                button_ungedrueckt.SRC="ungedr.gif";
                button_gedrueckt=new Image();
                button_gedrueckt.SRC="gedr.gif";

            //-->
        </SCRIPT>
    </HEAD>

    <BODY .....>
        .....
        <A      HREF="#"
            onMouseDown=" document.images[1].SRC=button_gedrueckt.src;"
            onMouseUp=" document.images[1].SRC=button_ungedrueckt.src;"

            <IMG      SRC="ungedr.gif"
                BOLDER=0
                HSPACE=abstand_zum_alternativen_text
            >
            alternativer_text
        </A>
    </BODY>
</HTML>

```

9.1. Aussehen eine Button-Objektes verändern

Text eines Button gestalten und Bild einfügen:

statischer Button per HTML-Code:

```

<BUTTON ID="ID_AbhakButton" VALUE=">
<IMG SRC="haekchen_bild.jpg">
<SPAN>Ich bin einverstanden.</SPAN>
</BUTTON>

<BUTTON ID="ID_KreuzButton" VALUE=">
<DIV>
<IMG SRC="kreuz_bild.jpg">
<SPAN>Ich lehne ab.</SPAN>
</BUTTON>

```



dynamischer Button per JScript:

```

var ButtonErzeugen_ObjektZeiger_Button; // interner Arbeitszeiger
var ButtonErzeugen_ObjektZeiger_IMG; // interner Arbeitszeiger
var ButtonErzeugen_ObjektZeiger_SPAN; // interner Arbeitszeiger

function ButtonErzeugen(X00,X01) // Button in document.body erzeugen
// X00 Pfad und Dateiname des Bildes
// wenn Leerkette so kein Bild erzeugt
// X01 Text des Buttons (auch HTML-Kodierungen wie Umlaute etc.)
// wenn Leerkette so kein Button erzeugt
// liefert true wenn Button erzeugt wurde
// füllt ButtonErzeugen_ObjektZeiger_Button
// aber nur gültig, wenn Returnwert true
{ var X02=false;
  var X03;
  // +++++ Parameter prüfen
  X02=(X00!=null);
  if(X02){ X02=(X01!=null); }
  if(X02){ X02=(X01!=""); }
  // +++++ BUTTON in document.body erzeugen
  if(X02)
  { X03=document.createElement('BUTTON');
    X02=(X03!=null);
    if(X02)
    { ButtonErzeugen_ObjektZeiger_Button=document.body.appendChild(X03);
      X02=(ButtonErzeugen_ObjektZeiger_Button!=null);
      if(X02){ X02=(ButtonErzeugen_ObjektZeiger_Button==X03); }
    }
  }
  // +++++ erst IMG im Button optional erzeugen
  if(X02)
  { if(X00!="")
    { X03=document.createElement('IMG');
      X02=(X03!=null);
      if(X02)
      { ButtonErzeugen_ObjektZeiger_IMG=ButtonErzeugen_ObjektZeiger_Button.appendChild(X03);
        X02=(ButtonErzeugen_ObjektZeiger_IMG!=null);
        if(X02){ X02=(ButtonErzeugen_ObjektZeiger_IMG==X03); }
        if(X02){ ButtonErzeugen_ObjektZeiger_IMG.src=X00; }
      }
    }
  }
  // +++++ dann immer SPAN erzeugen
  if(X02)
  { X03=document.createElement('SPAN');
    X02=(X03!=null);
    if(X02)
    { ButtonErzeugen_ObjektZeiger_SPAN=ButtonErzeugen_ObjektZeiger_Button.appendChild(X03);
      X02=(var ButtonErzeugen_ObjektZeiger_SPAN!=null);
      if(X02){ X02=(ButtonErzeugen_ObjektZeiger_SPAN==X03); }
      if(X02){ ButtonErzeugen_ObjektZeiger_SPAN.innerHTML=X01; }
    }
  }
  return X02;
}

...

var ID_AbhakButton;
if(ButtonErzeugen("haekchen_bild.jpg","Ich bin &uuml;berzeugt.)) // Button in document.body erzeugen
  // X00 Pfad und Dateiname des Bildes
  // wenn Leerkette so kein Bild erzeugt
  // X01 Text des Buttons (auch HTML-Kodierungen wie Umlaute etc.)
  // wenn Leerkette so kein Button erzeugt
  // liefert true wenn Button erzeugt wurde

```



```

        //          füllt ButtonErzeugen_ObjektZeiger_Button
        //          aber nur gültig, wenn Returnwert true
{ID_AbhakButton=ButtonErzeugen_ObjektZeiger_Button;
  ID_AbhakButton.style.position='absolute';
  ID_AbhakButton.style.top=...;
  ID_AbhakButton.style.left=...;
  ID_AbhakButton.style.border=...;
  ....
}

var ID_KreuzButton;
if(ButtonErzeugen("kreuz_bild.jpg","Ich lehne <B>ab</B>.")) // Button in document.body erzeugen
  // X00 Pfad und Dateiname des Bildes
  //          wenn Leerkette so kein Bild erzeugt
  // X01 Text des Buttons (auch HTML-Kodierungen wie Umlaute etc.)
  //          wenn Leerkette so kein Button erzeugt
  // liefert true wenn Button erzeugt wurde
  //          füllt ButtonErzeugen_ObjektZeiger_Button
  //          aber nur gültig, wenn Returnwert true
{ID_KreuzButton=ButtonErzeugen_ObjektZeiger_Button;
  ID_KreuzButton.style.position='absolute';
  ID_KreuzButton.style.top=...;
  ID_KreuzButton.style.left=...;
  ID_KreuzButton.style.border=...;
  ....
}

```

10. Mathematische Berechnungen (math-Objekt)

10.1. Gibt den Wert von Math.PI aus

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    document.write("Der Wert von Pi ist ");
    document.write(Math.PI); // Der Wert von Pi ist 3.141592653589793
// -->
</SCRIPT>
</HTML>

```

10.2. Runden auf n Nachkommastellen

```

function runden(numerischer_wert, nachkommastellen)
{
    var faktor = Math.pow(10, nachkommastellen)
    return Math.round(numerischer_wert * faktor) / faktor
}

```

10.3. Zufallszahl

Beispiel 1

```

<SCRIPT LANGUAGE="JavaScript">
<!--
    var textfeld = new Array(3);
    textfeld[0] = "TEXT0";
    textfeld[1] = "TEXT1";
    textfeld[2] = "TEXT2";
    textfeld_lang= textfeld.length;

    var zufall_index= Math.floor(    Math.random()
                                   *    textfeld_lang
                                   );
    document.writeln(textfeld[zufall_index]);
//-->
</SCRIPT>

```

Beispiel 2:

```

function ZufallsWertGanzzahligErzeugen(MaxWert)
// Zufallszahl von inklusive 0
//          bis inklusive MaxWert

```



```
// erzeugen als ganze Zahl
{
    var ZufallsZahl=Math.random(); // ab eventuell einschliesslich 0 bis unter 1, Gleitkomma
                                    // eventuell: falls nicht inklusive 0, so doch 0
                                    // letztendlich erzeugbar durch nachfolgendes
                                    // Runden
    if (ZufallsZahl > 0)              // Bps. MaxWert 3 Zufallszahl 0,412345
                                    // 0,612345
                                    // 0,0015545
                                    // 0,29456
    {
        // Kommaverschiebung
        while ((ZufallsZahl * 10) <= MaxWert)
        {ZufallsZahl = ZufallsZahl * 10;} // 0,412345 0,612345 1,5545 2,9456

        ZufallsZahl= Math.round(ZufallsZahl); // Kommastellen abschneiden durch runden
                                              // ab inklusive 0,5 aufwärts, sonst abwärts
                                              // 0 1 2 3

        // falls Runden einen Wert > MaxWert ergab, so den Wert begrenzen auf MaxWert
        if (ZufallsZahl > MaxWert)
        {ZufallsZahl = MaxWert;}
    }

    return ZufallsZahl;
}
}
```

10.4. **Größten gemeinsamen Teiler zweier ganzer Zahlen ab Javascript 1.2 per Rekursion ermitteln**

```
function ermittle_ggt_rekursiv(ganze_zahl_1, ganze_zahl_2) // ganze_zahl_1 >= ganze_zahl_2
{
    if (ganze_zahl_2 ==0)
    {return ganze_zahl_1;}
    else
    {return ermittle_ggt_rekursiv((ganze_zahl_2,(ganze_zahl_1 % ganze_zahl_2));}
}
}
```

10.5. **konvertieren zu ganzzahlig**

```
function KonvertiereZu_GanzeZahl(Wert)
{
    var ReturnWert = 0;

    if (Wert != 0)
    {
        var Faktor = 1;

        if (Wert < 0)
        {
            Faktor = -1; // Ergebnis ist negativ
            Wert = -1 * Wert; // nur positiven Werten wegen Math.floor()
                            // Bsp: 3.1456 wird zu 3
                            // also Kommaabschneidung
                            // -22,4567 wird zu -23 da -23 < -22
                            // also keine Kommaabschneidung
                            // aber 22,4567 wird zu 22
                            // also Kommaabschneidung
        }

        ReturnWert = Faktor * Math.floor(Wert); // Bsp.: 0,01 wird zu 0
                                              // Bsp.: 1,01 wird zu 1
    }

    return ReturnWert;
}
}
```

10.6. **konvertieren zu ganzzahlig teilbar**

```
function KonvertiereZu_GanzzahligTeilbar(Wert)
// wenn Wert > 0, so liefert die größte ganze Zahl <= Wert, die durch 2 teilbar ist Bsp: 5, so 4 geliefert
// wenn Wert < 0, so liefert die größte ganze Zahl >= Wert, die durch 2 teilbar ist Bsp: -5, so -4 geliefert
```



```

{
    // zu ganze Zahl konvertieren wegen Modulo: Der Rest ist immer ganzzahlig, sonst 0 !!
    //      Bsp.:    5 % 2   ergibt Rest 1
    //      4,3 % 2   ergibt Rest 0
    var ReturnWert = KonvertiereZu_GanzeZahl(Wert);

    if (ReturnWert != 0)           // Division von 0 durch ReturnWert ungleich Null ergibt immer 0
    {
        if (ReturnWert >= 2) // 1 ist nicht ganzzahlig teilbar
        {
            while ((ReturnWert % 2) != 0)
            {ReturnWert--;}          // Bsp.: 5 % 2 Rest 1, dann 4 % 2 Rest 0, also 4 geliefert
        }
        else
        {
            if (ReturnWert <= -2) // -1 ist nicht ganzzahlig teilbar
            {
                while ((ReturnWert % 2) != 0)
                {ReturnWert++;}      // Bsp.: -5 % 2 Rest 1, dann -4 % 2 Rest 0, also -4 geliefert
            }
        }
    }

    return ReturnWert;
}

```

10.7. **ganzzahlige Division**

```

function GanzZahlDivision(Dividend, Divisor)    // wenn Divisor == 0, so 0 geliefert
{
    var ReturnWert = 0;

    // Division durch Null ausschliessen
    if (Divisor != 0)
    {
        // Vorzeichen des Ergebnisses ermitteln UND nur positiven Werten wegen Math.floor()
        //      Bsp:      3,1456  wird zu 3      also Kommaabschneidung
        //      -22,4567 wird zu -23 da -23 < -22 also keine Kommaabschneidung
        //      aber 22,4567 wird zu 22 also Kommaabschneidung

        var Faktor = 1;

        if (Dividend < 0)
        {
            Faktor = -1 * Faktor;          // Ergebnis ist negativ
            Dividend = -1 * Dividend;      // und positiv setzen
        }

        if (Divisor < 0)
        {
            Faktor = -1 * Faktor; // falls Dividend < 0, so Ergebnis positiv; sonst negativ
            Divisor = -1 * Divisor;      // und positiv setzen
        }

        ReturnWert = Faktor * Math.floor(Dividend / Divisor);
    }

    return ReturnWert;
}

```

11. **Datenstrukturen**

11.1. **HTML-Datenstrukturen anlegen und verwerten**

11.1.1. **ausführliche Kodierung**

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function datenstruktur_ausgeben()
    {

```




```

        with (document)
        {
            write(person.vorname + " " + person.nachname + "<BR>");
            write(person.strasse + " " + person.nummer + "<BR>");
            write(person.plz + " " + person.ort + "<BR>");
            //      Erika Mustermann
            //      Musterstrasse 1
            //      .....
        }

        for (i in person)
        {
            document.write(i + ": " + person[i] + "<BR>");
            //      vorname: Erika
            //      nachname: Mustermann
            //      .....
        }
    }

    function datenstruktur_erzeugen(vorname, nachname, strasse, nummer, plz, ort)
    {
        this.vorname = vorname;
        this.nachname = nachname;
        this.strasse = strasse;
        this.nummer = nummer;
        this.plz = plz;
        this.ort = ort;
    }

    person = new datenstruktur_erzeugen
        (
            "Erika",           // Vorname
            "Mustermann",     // Nachname
            "Musterstrasse",   // Strasse
            "1",               // Nummer
            "10000",           // PLZ
            "Musterstadt");    // Ort
    );
// -->
</SCRIPT>
</HTML>

```

11.1.2. verkürzte Kodierung und Prototyping

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function datenstruktur_ausgeben()
    {
        with (document)
        {
            write(person.vorname + " " + person.nachname + "<BR>");
            write(person.strasse + " " + person.nummer + "<BR>");
            write(person.plz + " " + person.ort + "<BR>");
            //      Erika Mustermann
            //      Musterstrasse 1
            //      .....
        }

        for (i in person)
        {
            document.write(i + ": " + person[i] + "<BR>");
            //      vorname: Erika
            //      nachname: Mustermann
            //      .....
        }
    }

    var person = {
        vorname:"Erika",           // Vorname
        nachname:"Mustermann",     // Nachname
        strasse:"Musterstrasse",   // Strasse
        nummer:"1",               // Nummer
        plz:"10000",               // PLZ
        ort:"Musterstadt";         // Ort
        methode:datenstruktur_ausgeben // Ausgabemethode
    }

```



```

    };

    person.methode(); // Prototyping, also person um eine Methode erweitern
// -->
</SCRIPT>
</HTML>

```

11.2. **Datensätze als Teil einer Javascript Datei (*.js)**

11.2.1. ***.js-Datei erzeugen**

Die Javascript-Datei darf NUR Javascriptanweisungen enthalten !

Beispiel daten.js

Pro Satz sind 3 Felder vorhanden, die mit Zeichenkettenwerten belegt werden.
Die Anzahl der Sätze sind 3.
Die Sätze werden in eine Array-Variablen abgelegt → Array-Indizierung ab 0 !!

Aufbau von daten.js:

```

var anzahl_saetze=2;                                /* Anzahl stets ab 0, sonst beliebig */

function erzeuge_satz(satz_teil0, satz_teil1, satz_teil2)
                                                    /* Konstruktor zur satzweisen Feldbelegung */
{
    this.satz_teil0=wert0;
    this.satz_teil1=wert1;
    this.satz_teil2=wert3;
}

datensaetze_feld=new ARRAY(anzahl_saetze);
datensaetze_feld[0]=new erzeuge_satz("0_1","0_2","0_3");
datensaetze_feld[1]=new erzeuge_satz("1_1","1_2","1_3");
datensaetze_feld[2]=new erzeuge_satz("2_1","2_2","2_3");

```

11.2.2. ***.js-Datei in das HTML-Dokument einbinden und auf die Daten zugreifen**

Beispiel zu daten.js

```

<HTML>
  <HEAD>
    <TITLE> .... </TITLE>
    <SCRIPT LANGUAGE="JavaScript" SRC="daten.js"> </SCRIPT>

    <SCRIPT LANGUAGE="JavaScript" >
      <!--
          // Zugriff per datensaetze_feld[index].satz_teil0 etc.
      //-->
    </SCRIPT>
  </HEAD>
</HTML>

```

Hinweis: Mit Einbinden der *.js-Datei wird der Javascript-Code SOFORT interpretiert. Damit werden alle Anweisungen außerhalb von Funktionen sofort ausgeführt.



11.2.3. Beispiel Stichwortliste in obiger daten.js mit freier Suche nach Stichworten

Beispiel daten.js

Es wird im satz_teil0 eines jeden Satzes gesucht.

Das Stichwort ist der Suchbegriff.

Im Satz und im Suchbegriff können Gross -und kleinbuchstaben auftauchen.

Der Suchbegriff kann im satz_teil0 als Teilkette auftauchen, aber auch das gesamte Feld umfassen.

```
<HTML>
</HEAD>
<TITLE> .... </TITLE>

<SCRIPT LANGUAGE="JavaScript" SRC=daten.js"> </SCRIPT>           // js-Datei einbinden
<SCRIPT LANGUAGE="JavaScript">
<!--

function vergleich(satznr,suchwort)
{
    suchwort_laenge=suchwort.length;           /* Suchwort akt. Länge */
    datensatz_laenge=datensaetze[satznr].satz_teil0.length;
                                                    /* Feld 0 im Satz: akt. Länge */

    if (datensatz_laenge >= suchwort_laenge)
    {
        anzahl_pos= datensatz_laenge - suchwort_laenge;

        for (pos=0; pos <= anzahl_pos; pos++)
        {
            teilkette=
                datensaetze[satznr].satz_teil0.substring(pos,suchwort_laenge).toLowerCase();

            if (teilkette == suchwort.toLowerCase()) {return true;}
                                                    // gefunden im satz_teil0 des Datensatzes
        }
    }
    else {return false;}           // nicht gefunden im satz_teil0 des Datensatzes
}

function suche(suchwort)
{
    var gefunden=false;

    if (anzahl_saetze > 0)           // anzahl_saetze laut js-Datei !!
    {
        var satzzahler=-1;
        while (
            (!gefunden)
            && (satznr <= anzahl_saetze)
        )
        {
            satzzahler++;           // um 1 erhöhen
            gefunden= vergleich(satzzahler,suchwort);
        }
    }

    return gefunden;
}

function starten()
{
    suchwort=document.form_name.input_name.value;
    if (suchwort == "")
    {
        alert("Bitte Suchwort eingeben !");
    }
    else
    {
        if (suche(suchwort))
        {alert(" gefunden !");}
        else
        {alert("nicht gefunden!");}
    }
}
}
```



```

        </-->
    </SCRIPT>
</HEAD>

<BODY>
    <FORM NAME="form_name" onSubmit="starten()">
        Bitte Suchbegriff eingeben:
        <INPUT NAME="input_name" TYPE=TEXT>
        <INPUT TYPE=submit VALUE="Suche starten"> </INPUT>
    </FORM>
</BODY>
</HTML>

```

11.3. Programmierung eines Java-Stacks zum temporären Zwischenspeichern von Daten in den RAM

Die Reihenfolge der Daten auf dem Stack ist **UNBEDINGT** zu beachten:
 Der zuletzt auf dem Stack abgelegte Wert ist zuerst auszulesen.
 (Last in First Out = LiFo)

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        // Neues Stack-Objekt erzeugen
        java_objekt_stack = new java.util.Stack();

        function in_stack_ablegen(wert)
        {java_objekt_stack.push(wert);}

        function aus_stack_holen()
        {return java_objekt_stack.pop();}

    // -->
    </SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <INPUT TYPE="text"
            NAME="wert">
        <INPUT TYPE="button"
            VALUE="In Stack ablegen"
            onClick="in_stack_ablegen (this.form.wert.value)">
        <INPUT TYPE="button"
            VALUE="aus Stack holen"
            onClick="alert(aus_stack_holen ())">
    </FORM>
</BODY>
</HTML>

```

12. Ereignisse (Events)

12.1. Events allgemein

12.1.1. Der Event-Handler onAbort

```

<HTML>
    <IMG SRC="bild.jpg"
        WIDTH="100"
        HEIGHT="100"
        onLoad="alert('Bild geladen!')"
        onError="alert('Fehler beim Laden!')"
        onAbort="alert('Ladevorgang abgebrochen!')">
</HTML>

```

12.1.2. Der Event-Handler onBlur

Beispiel 1

```

<HTML>
<HEAD>

```



```

<SCRIPT LANGUAGE="JavaScript">
<!--
    function neuesFenster()
    {
        fenster = open("", "Anzeige", "width=300, height=300");
        fenster.document.open();
        with (fenster.document)
        {
            write("<HTML><BODY onBlur='window.close()'>");
            write("Dieses Fenster schlie&szlig;t sich von selbst,");
            write("wenn es nicht mehr aktiv ist.");
            write("</BODY></HTML>");
        }
        fenster.document.close();
    }
// -->
</SCRIPT>
</HEAD>
<BODY>
    <FORM>
        <INPUT TYPE="button"
            VALUE="Bitte dr&uuml;cken"
            onClick="neuesFenster()">
    </FORM>
</BODY>
</HTML>

```

Beispiel 2

```

<HTML>
<BODY bgcolor="0000ff"
    onBlur="document.bgColor='ff0000'"
    onFocus="document.bgColor='0000ff'">
    Dieses Fenster &auml;ndert die Hintergrundfarbe, wenn es nicht mehr aktiv ist.
</BODY>
</HTML>

```

12.1.3. Der Event-Handler onChange

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function aenderung(wert)
    {
        alert("Die Eingabe hat sich geaendert!\n" + "Der neue Wert ist " + wert);
    }
// -->
</SCRIPT>
</HEAD>
<BODY>
    <FORM>
        <INPUT TYPE="text"
            onChange="aenderung(this.value)">
    </FORM>
</BODY>
</HTML>

```

12.1.4. Der Event-Handler onClick**Beispiel 1**

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function meineFunktion()
    {
        alert("Hallo!");
    }
// -->
</SCRIPT>

```



```

</HEAD>
<BODY>
  <FORM>
    <INPUT TYPE="button"
           VALUE="Bitte drücken"
           onClick="meineFunktion()">
  </FORM>
</BODY>
</HTML>

```

Beispiel 2

```

<HTML>
  <FORM>
    <INPUT TYPE="checkbox"
           NAME="check"
           onClick="alert('OK!')">
    <INPUT TYPE="button"
           NAME="knopf"
           VALUE="Wert ändern"
           onClick="this.form.check.click()">
  </FORM>
</HTML>

```

Beispiel 3

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function neuesFenster()
        {
          fenster = open("", "Anzeige", "width=300, height=100");
          fenster.document.write("<HTML><BODY>");
          fenster.document.write("Mein Popup Fenster");
          fenster.document.write("</BODY></HTML><BR>");
        }
      // -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT TYPE="button"
            VALUE="Öffnen"
            onClick="neuesFenster()">
      <INPUT TYPE="button"
            VALUE="Schliessen"
            onClick="fenster.close()">
      <INPUT TYPE="button"
            VALUE="Nach vorne holen"
            onClick="fenster.focus()">
    </FORM>
  </BODY>
</HTML>

```

Beispiel 4

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function hole_taste(ereignis)
        {
          if (navigator.appName == "Netscape")
          { alert("Netscape erkannt. Maustaste gedrückt: "+ereignis.which);}
          else
          {
            if (navigator.appName == "Microsoft")

```



```

        {
            if (window.event.srcElement.type == "textarea")
            {
                alert("Explorer erkannt. Maustaste gedrückt:" +
                    window.event.keyCode);
            }
            else
            {
                alert("Weder Netscape noch Explorer erkannt ! " +
                    "Maustaste nicht ermittelbar !");
            }
        }
    }
}

//-->
</SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <P>
            <TEXTAREA ROWS=2 COLS=30
                <INPUT TYPE="button"
                    VALUE="Klicke "
                    onClick="hole_taste(event)>
            </TEXTAREA>
        </P>
    </FORM>
</BODY>
</HTML>

```

12.1.5. Der Event-Handler onError (Fehlermeldungen)

12.1.5.1. Fehlerbehandlung durch browsereigene Routinen abschalten

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    // Fehlermeldungen abschalten
    window.onError = null;

    // Damit wird bei den folgenden Fehlern wird keine Fehlermeldung ausgegeben
    // dafür mit Fehlerkennung das Script abgebrochen !!!
    alert(1 / "xyz");      // Division ungültig

    // Dieser Befehl wird nicht mehr abgearbeitet
    alert("Wird nicht aufgerufen!");

// -->
</SCRIPT>
</HTML>
```

12.1.5.2. Fehler umleiten auf private Behandlungsroutine anstelle der browsereigenen Routinen

12.1.5.2.1. Fehlermeldung per alert()

```
<HTML>
  <IMG SRC="bild.jpg"
    WIDTH="100"
    HEIGHT="100"
    onLoad="alert('Bild geladen!')"
    onError="alert('Fehler beim Laden!')">
</HTML>
```

12.1.5.2.2. Fehlermeldung in einem neuen Fenster

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        // Array fuer Fehlermeldungen
        fehlerText = new Array();
        fehlerURL = new Array();
        fehlerZeile = new Array();

        var fehlerNummer = 0;
```



```

// Fehlerbehandlungsroutine
function fehler(txt, URL, zeile)
{
    fehlerText[fehlerNummer] = txt;
    fehlerURL[fehlerNummer] = URL;
    fehlerZeile[fehlerNummer] = zeile;

    fehlerNummer++;
    return true;
}

function ausgabe()
{
    neuesFenster= window.open("",
                                "Fehler Fenster",
                                "scrollbars=yes,
                                WIDTH=500,
                                HEIGHT=400")

    with (neuesFenster.document)
    {
        open();
        write("<HTML><HEAD><TITLE>Fehlermeldungen");
        write("</TITLE></HEAD>");
        write("<H1>Folgende Fehler sind aufgetreten:</H1>");
        for (i = 0; i < fehlerNummer; i++)
        {
            write("<B>Datei:</B> <I>" + fehlerURL[i] + "</I><BR>");
            write("<B>Zeile:</B> <I>" + fehlerZeile[i] + "</I><BR>");
            write("<B>Fehlermeldung:</B> <I>" + fehlerText[i] +
                "</I><BR><BR>");
        }
        write("</BODY></HTML>");
        close();
    }
}

// Event-Handler onError neu setzen
window.onerror = fehler;
// -->
</SCRIPT>
</HEAD>
// Nachfolgend absichtlich falsche Scriptanweisungen
<BODY onLoad="nichtDefinierteFkt()" // Funktion nicht deklariert

<SCRIPT LANGUAGE="JavaScript">
<!--
    var x = "xyz";
    var y = 1/ x; // ungültige Division
// -->
</SCRIPT>

<FORM>
    <INPUT TYPE="button"
        VALUE="onBlur: Funktion nicht vorhanden"
        onBlur="FktGibtEsNicht()">
    <INPUT TYPE="button"
        VALUE="onClick: Anführungszeichen falsch!"
        onClick="alert('Anführungszeichen falsch!')">
    <P><BR>
    <INPUT TYPE="button"
        VALUE="letzten Fehler anzeigen"
        onClick="ausgabe()">
</FORM>
</BODY>
</HTML>

```

12.1.6. Der Event-Handler onFocus

Beispiel 1

```
<HTML>
```




```

<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      function ausgabe(text)
      { window.status = text;}

      function loeschen()
      { window.status = "";}
    // -->
  </SCRIPT>
</HEAD>

<BODY>
  <FORM>
    Text1:
    <INPUT TYPE="text"
           NAME="Text1"
           VALUE=""
           onFocus="ausgabe('Focus auf Text1') onBlur="loeschen()">

    <BR>
    Text2:
    <INPUT TYPE="text"
           NAME="Text2"
           VALUE=""
           onFocus="ausgabe('Focus auf Text2') onBlur="loeschen()">
  </FORM>
</BODY>
</HTML>

```

Beispiel 2

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function setzeFokus()
        {
          document.meinFormular.eingabe.focus();
          document.meinFormular.eingabe.select();
        }
      // -->
    </SCRIPT>
  </HEAD>

  <BODY>
    <FORM NAME="meinFormular">
      <INPUT TYPE="text"
            NAME="eingabe"
            VALUE="Eingabe...">
      <INPUT TYPE="button"
            VALUE="Bitte drücken"
            onClick="setzeFokus()">
    </FORM>
  </BODY>
</HTML>

```

12.1.7. Der Event-Handler onKeyPress

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function hole_taste(ereignis)
        {
          if (navigator.appName == "Netscape")
          { alert("Netscape erkannt. Taste "+ereignis.which+" gedrückt !");}
          else
          {
            if (navigator.appName == "Microsoft")
            {
              if (window.event.srcElement.type == "textarea")

```



```

        { alert("Explorer erkannt. Taste "+window.event.keyCode
              +" gedrückt !");
        }
    }
    else {alert("Weder Netscape noch Explorer erkannt ! "+
              "Taste nicht ermittelbar !");}
    }
}
//-->
</SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <P><TEXTAREA    ROWS="2"
                                COLS="30" onKeyPress="hole_taste(event)></TEXTAREA></P>
    </FORM>
</BODY>
</HTML>

```

12.1.8. Der Event-Handler onLoad

Beispiel 1

```

<HTML>
    <BODY    onLoad="alert('Herzlich Willkommen!')"
            onUnload="alert('Auf Wiedersehen!')">
        Dies ist meine HTML-Seite.
        <A HREF="http://www.test.de">Dies ist die Seite eines anderen.</A>
    </BODY>
</HTML>

```

Beispiel 2

```

<HTML>
    <IMG      SRC="bild.jpg"
            WIDTH="100"
            HEIGHT="100"
            onLoad="alert('Fertig!');">
</HTML>

```

12.1.9. Der Event-Handler onMouseout

```

<HTML>
    Bewegen Sie Ihren Mauszeiger &uuml;ber diesen
    <A      HREF="http://www.test.de"
            onMouseOver="javascript:window.status='Die test.de-Homepage';return true;"
            onMouseOut="javascript:window.status='';">
        Link</A>!
</HTML>

```

12.1.10. Der Event-Handler onMouseover

Beispiel 1

```

<HTML>
    Bewegen Sie den Mauszeiger über diesen
    <A      HREF="#"
            onMouseOver="alert('Hallo!')">Link</A>
</HTML>

```

Beispiel 2

```

<HTML>
    Bewegen Sie Ihren Mauszeiger &uuml;ber diesen
    <A      HREF="http://www.test.de"
            onMouseOver="javascript:window.status='Die test.de-Homepage'; return true;">
        Link</A>!
</HTML>

```

12.1.11. Der Event-Handler onReset

```

<HTML>
    <FORM onReset="javascript:return confirm('Alles loeschen?')">

```



```

        <INPUT TYPE="text"
              SIZE="40"
              VALUE="Bitte einen Text eingeben"><BR>
        <INPUT TYPE="checkbox">Test<BR>
        <INPUT TYPE="reset">
    </FORM>
</HTML>

```

12.1.12. Der Event-Handler onSelect

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function setzeFokus()
        {
            document.meinFormular.eingabe.focus();
            document.meinFormular.eingabe.select();
        }
      // -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM NAME="meinFormular">
      <INPUT TYPE="text"
            NAME="eingabe"
            VALUE="Eingabe...">
      <INPUT TYPE="button"
            VALUE="Bitte dr&uuml;cken"
            onClick="setzeFokus()">
    </FORM>
  </BODY>
</HTML>

```

12.2. Events im Netscape Communicator

12.2.1. Event onClick

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function test(e)
        { alert('pageX: ' + e.pageX + ' pageY: ' + e.pageY); }
      // -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <A HREF="#"
      onClick="test(event); return false;">
      <IMG SRC="obj.gif"
          WIDTH="278"
          HEIGHT="216"
          BORDER="0">
    </A>
  </BODY>
</HTML>

```

12.2.2. Event onLoad

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function init()
        { document.links[0].onclick= test; }

        function test(e)
        { alert('pageX: ' + e.pageX + ' pageY: ' + e.pageY); }
      // -->
    </SCRIPT>

```



```

</HEAD>

<BODY onLoad="init()">
  <A HREF="#">
    <IMG SRC="obj.gif"
          WIDTH="278"
          HEIGHT="216"
          BORDER="0"></A>

</BODY>
</HTML>

```

12.2.3. Das event-Objekt

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      function taste(e)
      {
        if (e.which == 97)
          {alert("Es wurde die Taste a gedrueckt");}
      }

      window.captureEvents(Event.KEYPRESS);
      window.onkeypress = taste;
    //-->
  </SCRIPT>

</HEAD>

<BODY>
  Druecken Sie die Taste 'a'.
</BODY>
</HTML>

```

12.2.4. Methode captureEvents()

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      function koordAusg(e)
      {status = "x: " + e.pageX + " y: " + e.pageY;}

      window.captureEvents(Event.MOUSEMOVE);
      window.onmousemove = koordAusg;
    //-->
  </SCRIPT>

</HEAD>

<BODY>
  Bewegen Sie den Mauszeiger ueber diese Seite.
</BODY>
</HTML>

```

12.2.5. Methode releaseEvents()

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      var anzeigen = false;

      function koordAusg(e)
      {status = "x: " + e.pageX + " y: " + e.pageY;}

      function koordAnz()
      {
        anzeigen = !anzeigen;
        if (anzeigen)
        {
          window.captureEvents(Event.MOUSEMOVE);
          window.onmousemove = koordAusg;
        }
      }
    -->
  </SCRIPT>

```



```

        else
        {
            window.releaseEvents(Event.MOUSEMOVE);
            window.onmousemove = null;
        }
    }
    <!-->
</SCRIPT>
</HEAD>
<BODY>
    <FORM>
        <INPUT TYPE="button"
            VALUE="Anzeige ein-/ausschalten"
            onClick="koordAnz()">
    </FORM>
</BODY>
</HTML>

```

12.3. Events im Microsoft Internet Explorer

12.3.1. Das event-Objekt

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function anzeigen()
        {
            var x = window.event.offsetX;
            var y = window.event.offsetY;

            status = "x: " + x + " y: " + y;
        }
    <!-->
</SCRIPT>
</HEAD>
<BODY>
    <IMG SRC="obj.gif"
        WIDTH="278"
        HEIGHT="216"
        onMouseMove="anzeigen();">
</BODY>
</HTML>

```

12.3.2. Zusätzliche Event-Handler

```

<HTML>
<BODY>
    <FORM>
        <INPUT TYPE="button"
            VALUE="Knopf"
            onMouseOver="window.status='Ein Knopf';"
            onMouseOut="window.status='';"
        >
        <BR>
        <INPUT TYPE="text"
            onMouseOver="window.status='Ein Eingabefeld';"
            onMouseOut="window.status='';"
        >
        <BR>
        <SELECT
            onMouseOver="window.status='Eine Auswahlliste';"
            onMouseOut="window.status='';"
        >
            <OPTION>Auswahl 1
            <OPTION>Auswahl 2
            <OPTION>Auswahl 3
        </SELECT>
    </FORM>

```



```
</BODY>
</HTML>
```

12.3.3. Das Bubble-up Event

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function over(e)
        {e.srcElement.style.color = "red";}

        function out(e)
        {e.srcElement.style.color = "blue";}
      //-->
    </SCRIPT>
  </HEAD>

  <BODY onmouseover="over(event)"
        onmouseout="out(event)">
    <A HREF="link1.htm">Link 1</A><BR>
    <A HREF="link2.htm">Link 2</A><BR>
    <A HREF="link3.htm">Link 3</A><BR>
  </BODY>
</HTML>
```

12.3.4. Das cancelBubble Event

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function reaktion()
        {
          alert('Button-Objekt');
          window.event.cancelBubble = true;
        }
      //-->
    </SCRIPT>
  </HEAD>

  <BODY>
    <FORM onClick="alert('Form-Objekt');">
      <INPUT TYPE="button"
            VALUE="Bitte druecken"
            onClick="reaktion();">
    </FORM>
  </BODY>
</HTML>
```

13. Reguläre Ausdrücke (reg-Methode-Suchalgorithmen)

13.1. Überprüfen einer E-Mail-Adresse

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function test(eing)
        {
          var reg= /@/;
          if (reg.exec(eing))
          {alert("Gueltige e-Mail Adresse");}
          else
          {else alert("Ungueltige e-Mail Adresse");}
        }
      //-->
    </SCRIPT>
  </HEAD>

  <BODY>
    Bitte geben Sie eine e-Mail Adresse ein:<BR>
```



```

        <FORM onSubmit="test(this.eingabe.value);return false;">
            <INPUT TYPE="text"
                NAME="eingabe"
                SIZE="30">
            <INPUT TYPE="button"
                VALUE="Test"
                onClick="test(this.form.eingabe.value)">
        </FORM>
    </BODY>
</HTML>

```

13.2. *Genauere Überprüfung einer E-Mail-Adresse*

```

<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JavaScript">
            <!--
                function test(eing)
                {
                    var reg= /.+@.+/;
                    if (reg.test(eing))
                    {alert("Gueltige e-Mail Adresse");}
                    else
                    {alert("Ungueltige e-Mail Adresse");}
                }
            <!-->
        </SCRIPT>
    </HEAD>
    <BODY>
        Bitte geben Sie eine e-Mail Adresse ein:<BR>
        <FORM onSubmit="test(this.eingabe.value);return false;">
            <INPUT TYPE="text"
                NAME="eingabe"
                SIZE="30">
            <INPUT TYPE="button"
                VALUE="Test"
                onClick="test(this.form.eingabe.value)">
        </FORM>
    </BODY>
</HTML>

```

13.3. *Besondere Zeichen suchen*

Beispiel 1

```

<HTML>
    <SCRIPT LANGUAGE="JavaScript">
        <!--
            var str= "x";
            var reg= /. /;

            if (reg.exec(str))
            { document.write("gefunden");}
            else
            { document.write("nicht gefunden");}
        <!-->
    </SCRIPT>
</HTML>

```

Beispiel 2

```

<HTML>
    <SCRIPT LANGUAGE="JavaScript">
        <!--
            var str= "Dieser String erstreckt sich\n" + "ueber mehrere Zeilen";
            var reg= /\n/;
            if (reg.exec(str))
            { document.write("Mehrere Zeilen.");}
            else
            { document.write("Nur eine Zeile.");}
        <!--

```



```

    //-->
</SCRIPT>
</HTML>

```

Beispiel 3

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var str= "Dies ist ein String,\n" + "der sich ueber mehrere\n" + "Zeilen erstreckt";
    var erg= str.split(/\s+/);

    document.write(erg.join(" - ")); // Ausgabe des Arrays
    //-->
</SCRIPT>
</HTML>

```

13.4. Teilketten suchen

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var str= ["abc", "aabc", "aaabc", "aaaabc", "aaabcbc", "aaaaaxyzaaabc",
             "bc", "aaaxbc", "aaabxyzc"];

    var reg= /a{1,3}bc/;

    for (i in str)
    {
        var found= reg.exec(str[i]);
        if (found)
        {document.write(str[i] + " - gefunden: " + found + "<BR>");}
        else
        {document.write(str[i] + " - nicht gefunden<BR>");}
    }
    //-->
</SCRIPT>
</HTML>

```

13.5. Überprüfung einer Formular-Text-Eingabe

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    function test(eing)
    {
        var reg= /stefan/i;
        if (reg.exec(eing))
        {location.HREF= "stefan.htm";}
        else
        {location.HREF= "default.htm";}
    }
    //-->
</SCRIPT>
</HEAD>

<BODY>
    Bitte geben Sie Ihren Namen ein:<BR>
    <FORM onSubmit="test(this.eingabe.value);return false;">
        <INPUT TYPE="text"
                NAME="eingabe"
                SIZE="30">
        <INPUT TYPE="button"
                VALUE="Log in"
                onClick="test(this.form.eingabe.value)">
    </FORM>
</BODY>
</HTML>

```



13.6. Anker suchen

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function test(eing)
        {
          reg= /\bstefan\b/i;
          (reg.exec(eing))
          {location.HREF= "stefan.htm";}
          else
          {location.HREF= "default.htm";}
        }
      //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    Bitte geben Sie Ihren Namen ein:<BR>
    <FORM onSubmit="test(this.eingabe.value);return false;">
      <INPUT TYPE="text"
        NAME="eingabe"
        SIZE="30">
      <INPUT TYPE="button"
        VALUE="Log in"
        onClick="test(this.form.eingabe.value)">
    </FORM>
  </BODY>
</HTML>
```

14. Timer, Datum, Uhrzeit

14.1. Aktion nach einer Wartezeit starten

14.1.1. Erzeugt ein Popup-Fenster nach 10 Sekunden

```
<HTML>
Bitte warten...
  <SCRIPT LANGUAGE="JavaScript">
    <!--
      setTimeout("alert('Fertig!')", 10000);
    // -->
  </SCRIPT>
</HTML>
```

14.1.2. Lädt eine WWW-Seite nach 10 Sekunden.

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function wechseln()
        {location.HREF="http://www.test.de";}
      // -->
    </SCRIPT>
  </HEAD>
  <BODY onLoad="setTimeout('wechseln()', 10000)">
    Nach 10 Sekunden wird automatisch die Homepage von test geladen.
  </BODY>
</HTML>
```

14.2. Tageberechnungen

14.2.1. Aktuellen Tag prüfen z.B. auf den 1. April

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  function pruefen()
  {
    jetzt=new Date();
```



```

        if (jetzt.getMonth() == 4)
        {
            if (jetzt.getDate() == 1)
            {alert("1. April !");}
            else {alert("kein 1. April!")}}
        }
    else {alert("kein 1. April!")}}
}
//-->
</SCRIPT>

```

14.2.2. Wieviel Tage sind es noch bis Weihnachten?

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    // Date-Objekte erzeugen
    var heute = new Date();
    var weihnachten = new Date(1900 + heute.getYear(), 11, 24);

    // Ist Weinachten dieses Jahr schon vorbei?
    if (heute.getTime() > weihnachten.getTime())
    { weihnachten.setYear(weihnachten.getYear() + 1);}

    // Berechnung der Differenz
    var differenz = weihnachten.getTime() - heute.getTime();
    var tage = Math.floor( differenz / (1000 * 60 * 60 * 24));

    document.write("Es sind noch ");
    document.write(tage);
    document.write(" Tage bis zum n&auml;chsten" +
        " Weihnachten.");

    // -->
</SCRIPT>
</HTML>

```

14.3. Zeigt die Uhrzeit und das Datum an

Beispiel 1

```

<HTML>
<SCRIPT LANGUAGE="JavaScript">
<!--
    heute = new Date();
    document.write("Jetzt ist es: ", heute.getHours(), ":",
    heute.getMinutes(), " Uhr.<BR>");
    document.write("Datum: ", heute.getDate(), ".",
    heute.getMonth()+1, ".", heute.getYear());
    // -->
</SCRIPT>
</HTML>

```

Beispiel 2

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var ZeitString, DatumsString;
    function ZeitDatum ()
    {
        Jetzt = new Date();

        // aktuelle Uhrzeit holen
        Stunden = Jetzt.getHours();
        Minuten = Jetzt.getMinutes();
        Sekunden = Jetzt.getSeconds();
        ZeitString = "" + Stunden;
        ZeitString += ((Minuten < 10) ? ":0" : ":") + Minuten;
        ZeitString += ((Sekunden < 10) ? ":0" : ":") + Sekunden;
        document.Uhr.Zeit.value = ZeitString;
    }

```



```

// aktuelles Datum holen
Tag = Jetzt.getDate();
Monat = Jetzt.getMonth()+1;
Jahr = Jetzt.getYear();
DatumsString = "" + Tag;
DatumsString += ((Monat<10) ? ".0" : ".") + Monat;
DatumsString += "." + Jahr;
document.Uhr.Datum.value = DatumsString;

Timer = setTimeout("ZeitDatum()", 1000);
    }
// -->
</SCRIPT>
</HEAD>

<BODY onLoad="ZeitDatum()">
    <CENTER>
        <FORM NAME="Uhr">
            Jetzt ist es
            <INPUT TYPE="text"
                NAME="Zeit"
                SIZE="8"
                VALUE="">

            Uhr.<BR>
            Datum:
            <INPUT TYPE="text"
                NAME="Datum"
                SIZE="8"
                VALUE="">

        </FORM>
    </CENTER>
</BODY>
</HTML>

```

Beispiel 3

```

HTML>
<HEAD>
<SCRIPT LANGUAGE=JavaScript1.2>
<!--
//      Datumsanzeige in deutscher Norm

//*****
//
//      Vom Programmierer ist nichts zu verändern
//
//*****
//      Felder der deutschen Datumsangaben
var WochenTag=new Array( "Sonntag",
    "Montag",
    "Dienstag",
    "Mittwoch",
    "Donnerstag",
    "Freitag",
    "Sonntag");

var Monate=new Array("Januar",
    "Februar",
    "März",
    "April",
    "Mai",
    "Juni",
    "Juli",
    "August",
    "September",
    "Oktober",
    "November",
    "Dezember");

```



```
//      Daten des aktuellen Datums ermitteln
var Datum_Aktuell=new Date();
var Datum_Aktuell_Jahr=Datum_Aktuell.getYear();
var Datum_Aktuell_WochenTag=Datum_Aktuell.getDay();
var Datum_Aktuell_Monat=Datum_Aktuell.getMonth();
var Datum_Aktuell_Tag=Datum_Aktuell.getDate();

//      Anpassungen der Daten des aktuellen Datums
//      wenn Jahresangabe < 1000 so 1900 dazuzaddieren
if (Datum_Aktuell_Jahr < 1000) {Datum_Aktuell_Jahr+=1900;}
//      Vornull einbasteln
if (Datum_Aktuell_Tag<10)   {Datum_Aktuell_Tag="0"+Datum_Aktuell_Tag;}

//      Formatierte Anzeige des aktuellen Datums in deutscher Norm
document.write( WochenTag[Datum_Aktuell_WochenTag]
               + ", "
               + Datum_Aktuell_Tag
               + ". "
               + Monate[Datum_Aktuell_Monat]
               + " "
               +Datum_Aktuell_Jahr);

//-->
</SCRIPT>
<BODY>
</BODY>
</HTML>
```

14.4. Uhren anzeigen

14.4.1. Minuten-Eieruhr

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    var restzeit_in_sekunden;
    var uhr gestartet = false;
    var id;

    function restzeit_pruefen()
    {
        if (      (uhr gestartet)
        {
            if (restzeit_in_sekunden > 0)
            {
                restzeit_in_sekunden --;

                var minuten = Math.floor(restzeit_in_sekunden / 60);
                // Integerdivison
                var sekunden = restzeit_in_sekunden - (minuten *60);

                document.Eieruhr.Wartezeit.value = "" + minuten + ":"
                + sekunden;

                id = setTimeout("restzeit_pruefen()", 1000)
                // jede Sekunden aktivieren = 1000 Millisekunden
            }
            else
            {
                alert("Eieruhr ist abgelaufen !");
                clearTimeout(id);
                uhr gestartet = false;
            }
        }
        else
        (alert("Eieruhr tickt nicht !");)
    }

    function uhrstart(warteminuten)
    {
        if (uhr gestartet)

```



```

        {
            if (confirm("Eieruhr tickt bereits !\nSoll " +
                        "die Uhr neu gestellt werden?"))
            {
                var sekunden = 0;
                restzeit_in_sekunden = warteminuten * 60;

                document.Eieruhr.Wartezeit.value = "" + warteminuten + ":" +
                                                    + sekunden;
                id= setTimeout("restzeit_pruefen()", 1000);
                    // jede Sekunden aktivieren = 1000 Millisekunden

                uhrgestartet = true;
            }
            else
            (alert("Eieruhr nicht neu gestartet !");)
        }
        else
        {
            var sekunden = 0;
            restzeit_in_sekunden = warteminuten * 60;

            document.Eieruhr.Wartezeit.value = "" + warteminuten + ":" +
                                                + sekunden;
            id= setTimeout("restzeit_pruefen()", 1000);
                // jede Sekunden aktivieren = 1000 Millisekunden

            uhrgestartet = true;
            alert("Eieruhr tickt bereits !");
        }
    }
    // -->
</SCRIPT>
</HEAD>
<BODY>
    <FORM NAME="Eieruhr">
        Bitte Zeit in Minuten eingeben:
        <INPUT TYPE="text"
                NAME="Dauer_In_Minuten"
                VALUE="">
        <INPUT TYPE="button"
                VALUE="Eieruhr starten"
                onClick="uhrstart(this.form.Dauer_In_Minuten.value)"><BR>
        verbleibende Zeit in Sekunden:
        <INPUT TYPE="text"
                NAME="Wartezeit"
                VALUE="">
    </FORM>
</BODY>
</HTML>

```

14.4.2. Uhr im Formular anzeigen

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
        <!--
            var id = null;
            var ticken = false;

            function anzeigen()
            {
                var zeit = new Date();

                document.formular.uhrzeit.value =
                    zeit.getHours()
                    + ":"
                    + zeit.getMinutes()
                    + ":"
                    + zeit.getSeconds();

                ticken = true;
                id = setTimeout("anzeigen()", 1000);
            }
        </SCRIPT>
    </HEAD>
</HTML>

```



```

    }

    function stop()
    {
        if(ticken)
        {
            clearTimeout(id);
            ticken= false;
        }
    }

    function start()
    {
        stop();
        anzeigen ();
    }
}
//-->
</SCRIPT>
</HEAD>

<BODY ... onLoad="start()">
    <FORM NAME="formular">
        <INPUT TYPE=text
            NAME="uhrzeit"
            READONLY SIZE=8 VALUE="">
    </FORM>
</BODY>
</HTML>

```

14.4.3. Uhr in der Statuszeile

```

<HEAD>
<SCRIPT ...>
<!-- // globale Größen festlegen
function uhr ()
{
    uhr_instanz=new Date();
    stunde= uhr_instanz.getHours();
    stunde=(stunde<10 ) ? "0"+stunde:stunde;
    minute= uhr_instanz.getMinutes();
    minute=(minute<10 ) ? "0"+minute:minute
    sekunde=uhr_instanz.getSeconds();
    sekunde=(sekunde<10 ) ? "0"+sekunde:sekunde;
    uhr_wert=stunde + ":" + minute+ ":" + sekunde+ " Uhr";
    time=setTimeout("uhr()",1000);
}
// -->
</SCRIPT>
</HEAD>

<BODY onLoad="uhr()">

```

15. Bilder und Animationen

Wird innerhalb von <DIV> ... </DIV> ein Bild per kodiert, so ist zu beachten:

Der DIV ist das übergeordnete Objekt zum IMG. Style-Angaben im IMG-Tag per STYLE="..." zu Positionen sind relativ zur den Dimensionen des DIV.

15.1. Bild preloaden (vorladen) in den Browser-Cache (ohne .readyState)

15.1.1. Pixelgröße 1x1 als Platzhalter

Soll eine Grafik subjektiv für den User schnell geladen werden, so ist das Laden in den Browser-Cache bereits VOR Aufruf der Seite, die die Grafik anzeigt, auszuführen. Oder man verwendet 1x1-Pixelgrafik GIF transparent als Platzhalter.

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2">
<!--
//          Bild vorladen
//          bei animierten Gif dauert es etwas, bis Animation aktiv wird
//
//*****
//
//          vom Programmierer zu belegende Variablen
//

```



```

var BildBreite=30;
var BildHoehe=49;
var BildUrl="picture.gif";
var LeeresBildUrl="1x1pixel.gif"// muss 1x1 Pixel sein
/*****
//
//      nachfolgender Code darf nicht verändert werden
//
//      Bildobjekt allokiieren
var BildObjekt = new Image();
//      Bildobjekt-Url zuweisen
BildObjekt.src = BildUrl;
//      Leeres Bild aber in den Dimensionen vom vorzuladenen Bild als IMG-Tag erzeugen
//      und damit anzeigen
document.write( ' <IMG '
                + 'NAME="LeeresBild"'
                + 'SRC=' + LeeresBildUrl
                + 'WIDTH=' + BildBreite
                + 'HEIGHT=' + BildHoehe
                + '>'
            );
//      neue Url zu weisen und vorladen
document['LeeresBild'].src = BildObjekt.src;
//-->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>

```

15.1.2. Mehrere Bilder mit Wartezeit preloaden

Die Wartezeit ist allerdings ein Schätzwert. Ausserdem lädt der Browser unvorhersehbar parallel.

Für exakte Wartezeit muss dateiweise vorgeladen werden, wobei pro Datei .readyState ausgewertet werden muss (Vorladen rekursiv).

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
<!--
    var bild_feld =      [      "b1.gif",
                                // hier alle Bilder eintragen.
                                ];

    var wartezeit = 500; // Zeit in ms zwischen zwei Ladevorgaengen
    var feld_index = 0;

    function preloaden()
    {
        var bild = new Image();
        bild.src = bild_feld[feld_index];

        feld_index ++;

        if(feld_index < bild_feld.length) // Länge ab 1, Index ab 0
        {setTimeout('preloaden()', load_next)}
    }

    function start()
    {setTimeout('preloaden()', wartezeit)}

    // -->
</SCRIPT>
</HEAD>
<BODY ... onLoad="start()">
</BODY>
</HTML>

```

15.2. Zeigt zwei verschiedene Bilder abwechselnd an

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">

```



```

<!--
    var aktuelles_bild = 1;

    function wechseln()
    {
        if (aktuelles_bild == 1)
        {
            document.werbung.src = "werbung2.gif";
            aktuelles_bild = 2;
        }
        else
        {
            document.werbung.src = "werbung1.gif";
            aktuelles_bild = 1;
        }
    }
// -->
</SCRIPT>
</HEAD>

<BODY>
    <IMG SRC="werbung1.gif" // Startbild
        NAME="werbung"
        WIDTH="250"
        HEIGHT="180"
        onLoad="setTimeout('wechseln()', 5000)">

</BODY>
</HTML>

```

15.3. Zeigt eine Bildfolge

Beispiel 1:

10 Bilder mit Namen b1.gif bis b10.gif. Wichtig ist der numerische Namenteil 1 bis 10, da als Zähler verwendet

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--

        verzoegerung = 120; // in Millisekunden
        bildNummer = 2;

        bilder= new Array(); //nimmt die Bilder der Animation auf

        // hier werden die Bilder im Hintergrund geladen
        for (i = 1; i <= 10; i++)
        {
            bilder[i] = new Image();
            bilder[i].src = "b" + i + ".gif";
        }

        function naechstesBild()
        {
            document.animation.src = bilder[bildNummer].src;
            bildNummer++;
            if (bildNummer > 10) bildNummer = 1;
        }
    // -->
    </SCRIPT>
</HEAD>

<BODY>
    <IMG SRC="dp1.gif"
        NAME="animation"
        WIDTH="165"
        HEIGHT="185"
        onLoad="setTimeout('naechstesBild()', verzoegerung)">

</BODY>
</HTML>

```



Beispiel 2:

es können **geladene Bilder** bild1.gif bis bild5.gif an fester Position im Wechsel ausgegeben werden
 geladenen Bilder erzeugen per Image-Objekt
 erstes Bild wird angezeigt per
 alle Bilder müssen gemeinsame Dimension haben, sonst wird gestreckt oder gestaucht

durch Aktualisierung vom Attribut SRC aus mit der jeweiligen Url des Bildes wird
 der sichtbare Bildwechsel vollzogen

```
<HEAD>
<SCRIPT ...>
<!--      // alle Bilder vorladen

           // globale Größen festlegen

           var bild_anzahl=5;           // ab 1

           var feld=new Array(bild_anzahl);// Inhalt des Feldes ist hier noch unbekannt

           feld[1]=new Image();         // Feldelement 1 ist Image-Objekt
                                           //      Bild 1 wird vorgeladen und nicht nicht angezeigt,
                                           //      da der Javascript-Code VOR dem IMG-Tag
                                           //      aus <BODY> .. </BODY> abgearbeitet wird !

           feld[1].url="bild1.gif";      // Prototyping: Eigenschaft url hinzufügen für Index 1
           ....

           feld[5]=new Image();         // Feldelement 5 ist ein Image-Objekt
           feld[5].url="bild5.gif";      // Prototyping: Eigenschaft url hinzufügen für Index 5

           var index=bild_anzahl;        // auch möglich           var index= feld.length;

           function bild_wechsel()
           {
               // Nummer des aktuellen Bild ermitteln
               if (index == bild_anzahl) index=0;
               index+=1;
               // Kopieren der Url nach <IMG>-Attribut SRC
               //      also gleichzeitig anzeigen
               self.document.logischer_img_name.url=feld[index].url;
               // logischer_img_name.src entspricht SRC aus <IMG>
           }

           // -->
           </SCRIPT>
</HEAD>
<BODY>
<IMG      NAME="logischer_img_name"
          SRC="bild1.gif"           // vorgeladenes Bild anzeigen
>
<SCRIPT ...>
<!--
setInterval(bild_wechsel,2000); // periodische Abarbeitung alle 2 Sekunden
// -->
</SCRIPT>
</BODY>
```

15.4. Festes, nicht mitscrollendes Bild

Beispiel 1: feste Grafik in einem Fenster
 für Internet Explorer oder Netscape

```
<HTML>
<HEAD>
<TITLE> .... </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
           posX=20;           // horizontal
           posY=20;           // vertikal
           idName="grafik_name";

           function merkeYPosition()
           {
               if (document.all)           // IE
```



```

        {
            merkeY=document.body.scrollTop;}
        else
            // Netscape
        {
            if (document.layers)
                {merkeY=pageYOffset;}
        }
    }

    funktion setzeYPosition()
    {
        merkeYPosition();
        if (document.all)
        {
            document.all[idName].style.top=merkeY+posY;}
            // idName in eckige Klammern

        else
        {
            if (document.layers)
                {document.layers[idName].top=merkeY+posY;}
            // idName in eckige Klammern

        }
    }

    funktion merkeXPosition()
    {
        if (document.all)
            // IE
        {
            merkeX=document.body.scrollLeft;}
        else
            // Netscape
        {
            if (document.layers)
                {merkeX=pageXOffset;}
        }
    }

    funktion setzeXPosition()
    {
        merkeXPosition();
        if (document.all)
        {
            document.all[idName].style.left=merkeX+posX;}
            // idName in eckige Klammern

        else
        {
            if (document.layers)
                {document.layers[idName].left=merkeX+posX;}
            // idName in eckige Klammern

        }
    }

    funktion korrigiereScrollen()
    {
        setzeYPosition();
        setzeXPosition();
        setTimeout("korrigiereScrollen()",1000); // alle Sekunde korrigieren
    }

    //-->
</SCRIPT>
</HEAD>

<BODY onLoad="korrigiereScrollen()">
    <DIV ID="grafik_name" // identisch mit idName
        STYLE="position:absolute; top:20; left:20; width:32px;">
        // top identisch mit Anfangswert von posY
        // left identisch mit Anfangswert von posX

        <IMG SRC=grafik.gif"
            WIDTH="32" // identisch mit STYLE
            HIGHT="32"
            BORDER="0"
            ALT="grafik.gif"

        >
    </DIV>

    // nachfolgender Code erzeugt einen Scrollfenster als Tabelle

    <TABLE WIDT="800"
        HEIGHT="800"
        BGCOLOR="yellow"

```



```

        >
        <TR>
        <TD> &nbsp;&nbsp;&nbsp;</TD>
        </TR>
    </TABLE>
</BODY>
</HTML>

```

Beispiel 2

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript1.2">
<!--
//      feste Grafik, die nicht mitscrollt
//      Grafik steht immer rechts unten im Fenster
//      Die Grafik-Positionskorrektur-Geschwindigkeit ist leider abhängig von der
//      Scrollgeschwindigkeit,
//      so dass ein Springen der Grafik unvermeidlich ist,
//      wenn die Scrollgeschwindigkeit zu gross ist.
//
//*****
//
//      nachfolgende Variablen müssen vom Programmierer gesetzt werden
//
//*****
//
var GrafikBreiteLautIMGinBODY=30;                // identisch mit IMG-Werten laut BODY
var GrafikHoeheLautIMGinBODY =49;                // identisch mit IMG-Werten laut BODY
var GrafikAbstandZurFensterEckeRechtsUntenX=0;   // in Pixel
var GrafikAbstandZurFensterEckeRechtsUntenY=0;   // in Pixel
var GrafikPositionsKorrekturGeschwindigkeit=1;    // >=1 wobei 1 = schnellste Positionskorrektur
//
//*****
//
//      nachfolgenden Code nicht verändern
//
//*****
//      Variablen hier ohne extra Funktion initialisieren
//      werden automatisch beim Dokumentladen belegt
//
//      BrowserTyp-Ermittlung
//      Annahme Netscape läuft
var BrowserTyp=true;
if (document.all)
{
    //      IE läuft
    BrowserTyp=false;
}

//      Browser-Fenster-Variablen
var BrowserFenster_AktuelleBreite      =0;
var BrowserFenster_AktuelleHoehe      =0;
var BrowserFenster_AktuelleScrollPositionX=0;
var BrowserFenster_AktuelleScrollPositionY=0;

//      Grafik-Positions-Korrektur-Variablen
var GrafikPositionsKorrekturX=0;
var GrafikPositionsKorrekturY=0;

function GrafikPositionKorrigieren()
{
    //      aktuelle Browserfenster-Daten ermitteln
    if (BrowserTyp)
    {
        //      Netscape läuft
        BrowserFenster_AktuelleBreite      =window.innerWidth;
        BrowserFenster_AktuelleHoehe      =window.innerHeight;
        BrowserFenster_AktuelleScrollPositionX=window.pageXOffset;
        BrowserFenster_AktuelleScrollPositionY=window.pageYOffset;
    }
}

```



```

else
{
    //      IE läuft
    BrowserFenster_AktuelleBreite      =document.body.clientWidth;
    BrowserFenster_AktuelleHoehe      =document.body.clientHeight;
    BrowserFenster_AktuelleScrollPositionX=document.body.scrollLeft;
    BrowserFenster_AktuelleScrollPositionY=document.body.scrollTop;
}

//      Korrekturpositionen für Grafik ermitteln
//      Achtung: Die Klammerung ist hier WICHTIG !!!
GrafikPositionsKorrekturX= BrowserFenster_AktuelleBreite
                        - (GrafikBreiteLautIMGinBODY+GrafikAbstandZurFensterEckeRechtsUntenX)
                        +
                        BrowserFenster_AktuelleScrollPositionX;

GrafikPositionsKorrekturY=BrowserFenster_AktuelleHoehe
                        - (GrafikHoeheLautIMGinBODY+GrafikAbstandZurFensterEckeRechtsUntenY)
                        +
                        BrowserFenster_AktuelleScrollPositionY;

//      Grafikposition korrigieren
//      eval () ermittelt String und liefert dessen Inhalt anstelle eval()
//      da der String ein StyleSheed-Kommando ist, wird das somit ausgeführt
if (BrowserTyp)
{
    //      Netscape läuft
    eval("document.StyleSheedID.left=" + GrafikPositionsKorrekturX);
    eval("document.StyleSheedID.top=" + GrafikPositionsKorrekturY);
}
else
{
    //      IE läuft
    eval("StyleSheedID.style.pixelLeft=" + GrafikPositionsKorrekturX);
    eval("StyleSheedID.style.pixelTop=" + GrafikPositionsKorrekturY);
}
setTimeout("GrafikPositionKorrigieren()",GrafikPositionsKorrekturGeschwindigkeit);
}
//-->
</SCRIPT>
</HEAD>
<BODY onload="GrafikPositionKorrigieren();">
<DIV ID="StyleSheedID" STYLE="position:absolute; visibility:show; left:0px; top:0px; z-index:2">
<CENTER>
<IMG SRC="picture.gif" WIDTH="30" HEIGHT="49" BORDER="0">
</CENTER>
</DIV>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>
Zeile<BR>

```



[illegible]

15.5. Bild in einem neuen Fenster zentriert öffnen

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<!--
```

```
// Grafik in einem neuen Fenster bezüglich aktueller Bildschirmauflösung zentriert öffnen
```

```
var user_bildschirm_breite=screen.width;
```



```

var user_bildschirm_hoehe=screen.height;
var bild_breite=640;
var bild_hoehe=444;

var bild_name="images/brehmer_031299_0.jpg"
var bild_alternate_text="Brehmer_031299_0.jpg";

function ermittle_rahmen_links_rechts()
{
    var rahmen_breite=0;
    if (user_bildschirm_breite > bild_breite)
    {
        rahmen_breite = user_bildschirm_breite - bild_breite;
        rahmen_breite = rahmen_breite / 2;
    }
    return rahmen_breite;
}

function ermittle_rahmen_oben_unten()
{
    var rahmen_hoehe=0;
    if (user_bildschirm_hoehe > bild_hoehe)
    {
        rahmen_hoehe = user_bildschirm_hoehe - bild_hoehe;
        rahmen_hoehe = rahmen_hoehe / 2;
    }
    return rahmen_hoehe;
}

function start()
{
    var fenster=window.open();

    // aktuelle Browserdimensionen anstelle der Init-Screen-Objekt-Werte ermitteln
    if (window.innerWidth)
    {
        // Netscape
        user_bildschirm_breite=window.innerWidth;
        user_bildschirm_hoehe =window.innerHeight;
    }
    else
    {
        // IE ab 4.x
        if (document.all)
        {
            user_bildschirm_breite=document.body.clientWidth;
            user_bildschirm_hoehe =document.body.clientHeight;
        }
    }

    // Grafik anzeigen
    with (fenster.document)
    {
        open("text/html","replace");
        write(
            "<TABLE CELLPADDING='0' CELLSPACING='0' BORDER='0'>"
            + "<TR><TD>"
            + "    <IMG SRC='" + bild_name + "'"
            + "    "ALT='" + bild_alternate_text + "'"
            + "    "WIDTH='" + bild_breite + "'"
            + "    "HEIGHT='" + bild_hoehe + "'"
            + "    "HSPACE='" + ermittle_rahmen_links_rechts() + "'"
            + "    "VSPACE='" + ermittle_rahmen_oben_unten() + "'"
            + "</TD></TR>"
            + "</TABLE>");
    }
    return true;
}

//-->
</SCRIPT>

```



```

</HEAD>
<BODY LEFTMARGIN="0" MARGINWIDTH="0" TOPMARGIN="0" MARGINHEIGHT="0"
onLoad="start();">
</BODY>
</HTML>

```

15.6. *Bild verschwinden lassen*

```

<DIV ID="DIV_ID"
    STYLE="position:absolute;left=0px;top=0px;"
>
    <IMG NAME="IMG_ID"
        SCR="alt.jpg"
        HEIGHT=20
        WIDTH=40
        STYLE="...."
    >
</DIV>

```

```

Variante 1 :if (document.layers)
{
    document.layers[DIV_ID].left = (-1 * 40);
    document.layers[DIV_ID].top = (-1 * 20);
}
else
{
    if (ie)
    {
        document.all.DIV_ID.style.left = (-1 * 40);
        document.all.DIV_ID_.style.top = (-1 * 20);
    }
}

```

Variante 2:document.IMG_ID.src="neu.gif"; // neu.gif ist transparent und 1x1 Pixel

16. Sound

Sound ist per Script nicht unkompliziert in den Browser-Cache vorladbar.

Beim Internet Explorer

gibt es keine Funktion analog zu new Image().

ist die Verwendung eines Konstruktors sinnlos, wenn der Browser nicht anweisbar ist, die Datei mit .src-Belegung zu laden.

muss ein Test-BGSound-Objekt verwendet werden (mit VOLUME auf -10 also stumm), dem .src zugewiesen wird,

was das Laden der Datei in den Browser-Cache bewirkt. Das Testobjekt muss mit .readyState überwacht

werden, will man feststellen, wann die Datei geladen ist, um dann das Testobjekt aus dem DOM zu löschen.

16.1. *Dokument mit klingendem Sound laden*

(am Beispiel des Live Audio-Plugin (JavaApplet) des NS)

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- vor alten Browsern verstecken
var sound_datei_url="test.wav";

function sound_laden_und_spielen()
{
    var live_audio_plugin=(navigator.plugins["LiveAudio"]);
    if (live_audio_plugin)
    {
        // LiveAudio-Plugin ist installiert
        if (navigator.javaEnabled())
        {
            // Browser hat Java eingeschaltet
            var mime_typ=(soundPlugin["audio/wav"]);

            if (mime_typ)
            {
                // Es existiert ein Plugin, das
                // audio/wav wiedergeben kann

                if (mime_typ.enabledPlugin == live_audio_plugin)

```



```

        {
            // das aktive Plugin ist LiveAudio
            document.write(
                '<EMBED SRC="'
                + sound_datei_url
                + ' HIDDEN=TRUE'
                + ' AUTOSTART=FALSE'
                + ' NAME=logischer_sound_name'
                + ' MASTERSOUND'
                + '>'
            );
            document.logischer_sound_name.play(false);
        }
    }
}

// -->
</SCRIPT>
</HEAD>
<BODY onLoad="sound_laden_und_spielen();>
</BODY>
</HTML>

```

Hinweise: Anstelle von onLoad kann natürlich der Anweisungsteil der Funktion sound_laden_und_spielen außerhalb einer Funktion kodiert werden (Wegfall der Funktion selbst), da immer zuerst der Head-Abschnitt des Dokumentes interpretiert wird.

Die Anweisung document.write gibt das EMBED-Tag in das Dokument aus und instanziert zugleich das Objekt **logischer_sound_name**, so dass die Methode play ausgeführt werden kann.

16.2. Sound für IE und NS

```

var SoundUrl = "56sec.mid";
var SoundDauerInSekunden = 56;

// ++++++ Routinen zur Erzeugung Sound
function SoundObjektErzeugen(SoundFileUrl, SoundFileDauerInSekunden)
{
    this.SoundFileUrl = SoundFileUrl;
    this.SoundFileDauerInSekunden = SoundFileDauerInSekunden;
    this.SoundBeendet=true;
}

// ++++++ Browser-Typ ermitteln
// Dieser Quellcode muss VOR allen anderen Routinen codiert sein, damit zuerst abgearbeitet
var ns = document.layers ? true : false;
var ie = document.all ? true : false;

var SoundAbspielKommando = "";
var SoundTimeoutID=0;
var SoundGestartet=false;

function SoundAbspielen()
{
    if (SoundGestartet == false)
    {
        // Sound aktivieren
        if (ie)
        {
            document.write( '<BGSOUND SRC="' + SoundObjekt.SoundFileUrl + "'
                            + ' LOOP="0"'
                            + '>'
            );
        }
        else
        {
            if (ns)
            {
                document.write( '<EMBED SRC="' + SoundObjekt.SoundFileUrl + "'
                                + ' AUTOSTART=true'

```




```

+ ' LOOP=false'
+ ' HIDDEN=true'
+ ' HEIGHT=0'
+ ' WIDTH=0'
+ '>'
);
}

SoundObjekt.SoundBeendet=false;
SoundGestartet=true;
// und warten auf Soundende
SoundTimeoutID = setTimeout("SoundAbspielen()", ((SoundObjekt.SoundFileDauerInSekunden+1) * 1000));
}
else
{
    // Sound zu Ende
    SoundObjekt.SoundBeendet=true;
}
}

// ++++++ Sound abspielen
var SoundAktiv=false;
var SoundGestartet=false;
SoundAbspielen();

```

17. Password

17.1. Password in einer Variablen als Auswahlkriterium für eine HTML-Seite

17.1.1. Eingabe per Formular

Beispiel 1

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- vor alten Browsern verstecken

    var teil_pfad="/index.htm";
    var voller_pfad=""

    function pfad_bilden
    { voller_pfad=logisches_password.value + teil_pfad;}

    // -->
</SCRIPT>
</HEAD>
<BODY>
<FORM>
    <INPUT TYPE=password
        NAME="logisches_password"
        VALUE=""
        SIZE="20"

    <INPUT TYPE="button"
        VALUE="Pfad bilden"

    onClick="pfad_bilden(this.form.logisches_password.value)">
</FORM>
</BODY>
</HTML>

```

Beispiel 2

```

<HTML>
<HEAD>
<TITLE> .... </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--

```



```

        function checkPassword()
        {
            if (      (document.pwort.pw.value != "")
            &&      (document.pwort.pw.value != null)
            { window.location=document.pwort.pw.value + ".htm";}
        }
    //-->
</SCRIPT>
</HEAD>
<BODY>
    <FORM NAME=pwort>
        Passwort eingeben:
        <INPUT TYPE=password
                    NAME="pw"
                    SIZE="8"
                    MAXLENGTH="8"
                >
        <INPUT TYPE="button"
                    VALUE="OK"
                    onClick="checkPassword()"
                >
    </FORM>
</BODY>
</HTML>

```

17.1.2. Eingabe per Hyperlink (HREF) und prompt

```

<HTML>
<HEAD>
    <TITLE> .....</TITLE>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function checkPassword()
        {
            pw=prompt("Passwort eingeben:", "");
            if (      (pw != "")
            &&      (pw != null)
            )
            { window.location=pw + ".htm";}
        }
    //-->
</SCRIPT>
</HEAD>
<BODY>
    <A HREF="javascript:checkPassword()">Password-Link</A>
</BODY>
</HTML>

```

17.1.3. Zufallspasswort erzeugen

```

<HTML>
<HEAD>
</HEAD>
<BODY>
<script language="JavaScript">
<!--

var laenge=10;

var zeichen="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
var zeichen_laenge=zeichen.length;
var Kette="";
var x=0;
var Wert=0;
for (x=0; x < laenge+1; x++)
{
    Wert=Math.random();
    Wert*=zeichen_laenge;
    Kette+=zeichen.charAt(Wert);
}

```



```

}

alert(Kette);
//-->
</SCRIPT>
</BODY>
</HTML>

```

17.2. CGI-Script eines Providers, das auf dessen Server läuft, für eine geschützte Homepage auf gleichem Server

z.B. www.cgi-service.de

CGI-Script wird vom Provider gestellt

Festlegung von URL der Homepage
 Passwörterinhalte
 URL des CGI-Scriptes

muss dementsprechend als Formular in HTML eingebunden werden

setzt voraus, dass Homepage auf Server grundsätzlich geschützt liegen --> kann nicht jeder Provider !!

Bsp.:

```

<FORM ACTION=http://-->www.cgi-service.de/cgi-cgi-service/999999/password/cgi_script.cgi>
  <INPUT TYPE=password
    NAME=cgi_script
  >
  <INPUT TYPE=hidden
    NAME=function
    VALUE=post
  >
  <INPUT TYPE=submit
    VALUE="Prüf mich "
  >
</FORM>

```

18. Cookies

Achtung: Cookies können durch einen Firewall unterbunden werden !

<cookie1.htm> Setzt ein Cookie, das durch die Seite cookie2.htm ausgelesen wird

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function setzeCookie(name, wert)
        {
          var arg_wert = setzeCookie.arguments;
          var arg_laenge = setzeCookie.arguments.length;
          var expires = (arg_laenge > 2) ? arg_wert[2] : null;
          var path = (arg_laenge > 3) ? arg_wert[3] : null;
          var domain = (arg_laenge > 4) ? arg_wert[4] : null;
          var secure = (arg_laenge > 5) ? arg_wert[5] : false;

          document.cookie =   name
                               + "="
                               + escape (wert)
                               + ( (expires == null) ? "" :
                                   ("; expires=" + expires.toGMTString()
                                   )
                               )
                               + ( (path == null) ? "" : ("; path=" + path)
                               )
                               + ( (domain == null) ? "" : ("; domain=" + domain)
                               )
                               + ( (secure == true) ? "; secure" : "");
        }

        setzeCookie("test", "Hallo!");
      // -->
    </SCRIPT>
  </HEAD>
</HTML>

```



```

        </SCRIPT>
    </HEAD>
</HTML>

```

<cookie2.htm>Liest das Cookie aus, das in cookie1.htm gesetzt wurde:

```

<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function holeCookie(name)
        {
            name += "=";
            var laenge = name.length;
            var cookie_laenge = document.cookie.length;

            var i = 0;
            while (i < cookie_laenge)
            {
                var j = i + laenge;
                if (document.cookie.substring(i, j) == name)
                {return holeCookieWert (j);}

                i = document.cookie.indexOf(" ", i) + 1;

                if (i == 0)
                {break;}
            }
            return null;
        }

        function holeCookieWert(position)
        {
            var ende = document.cookie.indexOf(";", position);

            if (ende == -1)
            {ende = document.cookie.length;}

            return unescape(document.cookie.substring(position, ende));
        }

        alert(holeCookie("test"));
      // -->
    </SCRIPT>
  </HEAD>
</HTML>

```

auf Cookieverwaltung testen

```

<script language="JavaScript">

function getCookieVal(offset)
{
    var endstr=document.cookie.indexOf(";",offset);
    if(endstr==-1)
    endstr=document.cookie.length;
    return unescape(document.cookie.substring(offset,endstr));
}

function GetCookie(name)
{
    var arg=name+"=";
    var alen=arg.length;
    var clen=document.cookie.length;
    var i=0;
    while(i<clen)
    {
        var j=i+alen;
        if(document.cookie.substring(i,j)==arg)
        return getCookieVal(j);
    }
}

```



```

i=document.cookie.indexOf(" ",i)+1;
if(i==0)break;
}
return null;
}

function SetCookie(name,value)
{
var argv=SetCookie.arguments;
var argc=SetCookie.arguments.length;
var expires=(argc>2)?argv[2]:null;
var path=(argc>3)?argv[3]:null;
var domain=(argc>4)?argv[4]:null;
var secure=(argc>5)?argv[5]:false;
document.cookie=name+"="+escape(value)+
((expires==null)?""; expires="+expires.toGMTString())+
((path==null)?""; path="+path))+
((domain==null)?""; domain="+domain))+
((secure==true)?"; secure":""");
}
var enabled="";
var exp=new Date();
exp.setTime(exp.getTime()+(60*1000));
SetCookie('wannasomecookie',1,exp);
enabled=GetCookie('wannasomecookie');
</script>
<script language="JavaScript">
if(enabled==null)document.write("<b><center>Ihr Browser akzeptiert keine Cookies<br>Your Browser does not accept Cookies");
else document.write("<b><center>Ihr Browser akzeptiert Cookies<br>Your Browser does accept Cookies");
</script>

```

19. Tastatur und Maus

19.1. *Browserspezifische Ermittlung EINER eingegebenen Taste in einem Textarea*

window.event.keyCode beim Internet Explorer:

Tastaturkombinationen werden nicht erkannt (also parallel gedrückte Tasten werden nicht erkannt)
 liefert dezimalen Wert, der als Hexadezimalwert dem Unicode entspricht.
 Umwandlung des Keycode nach Zeichen:

```
per String.fromCharCode(window.event.keyCode);
```

wobei Werte > 127 nur als Unicode akzeptiert werden z.B. fromCharCode(0x20AC) ergibt €
 Tastaturwerte ähneln ASCII-Code nur.

Beispiel 1

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
  <!--
    function hole_taste(ereignis)
    {
      if (navigator.appName == "Netscape")
      { alert("Netscape erkannt. Taste "+ereignis.which+" gedrückt !");}
      else
      { if (navigator.appName == "Microsoft")
        { if (window.event.srcElement.type == "textarea")
          { alert("Explorer erkannt. Taste "+window.event.keyCode+" gedrückt !");}
        }
        else
        {alert("Weder Netscape noch Explorer erkannt ! Taste nicht ermittelbar !");}
        }
    }
  //-->
  </SCRIPT>
</HEAD>

```



```

<BODY>
  <FORM>
    <P><TEXTAREA  ROWS="2"
                                COLS="30"
                                onKeyPress="hole_taste(event)>
    </TEXTAREA>
  </P>
</FORM>
</BODY>
</HTML>

```

Beispiel 2

```

function HoleTaste(e)
{
    var Taste;
    if (document.layers)
    { Taste = String.fromCharCode(e.wich);}
    else
    {
        if (document.all)
        { Taste = String.fromCharCode(event.keyCode);}
    }

    Taste = Taste.toLowerCase(); // Kleinbuchstabe
}

if (document.layers)
{ document.captureEvents(Event.KEYPRESS);}
else
{
    if (document.all)
    { document.attachEvent('onkeypress', HoleTaste);
    }
}

document.onkeypress = HoleTaste;

```

19.2. Rechte Maustaste belegen

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
<!--
    function rechte_maus_reaktion()
    {
        // hier die Reaktionen für rechte Maus einbauen
    }

    function rechte_maus_analyse(ereignis)
    {
        if(window.Event)
        {
            // NS
            if(ereignis.which == 3)
            {
                rechte_maus_reaktion()
                return false
            }
        }
        else
        {
            // IE
            if((event.button & 0x2) == 0x2)
            {
                rechte_maus_reaktion()
                return false
            }
        }
        return true
    }
    function init()

```



```

        {
            if(window.Event)    // NS
            {document.captureEvents(Event.CLICK | Event.MOUSEDOWN)}
            else
            {
                // IE
                document.attachEvent('onclick', rechte_maustaste);
                document.attachEvent('onmousedown', rechte_maustaste);
            }

            document.onmousedown = rechte_maustaste;
            document.onclick    = rechte_maustaste;
        }
    }
    </SCRIPT>
</HEAD>

<BODY ... onLoad="init()">
    ..
</BODY>
</HTML>

```

19.3. *Mauscursor permanent um ein mitlaufendes Bild erweitern*

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">

// Mauszeiger um ein Bild permanent erweitern:
//  Bild erscheint rechts neben dem Mauszeiger
//  Bild wandert mit dem Mauszeiger mit

// ##### frei belegbare Variablen
var Bild_Name   = "test.gif";
var Bild_Breite = 34; // Pixel
var Bild_Hoehe  = 34; // Pixel
var Bild_AbstandZurMausSpitze_X = 5; // Pixel >=0
var Bild_AbstandZurMausSpitze_Y = 5; // Pixel >=0

// ##### interne Variablen
var Dokument_ScrollPosition_BezueglichRand_Links;
var Dokument_ScrollPosition_BezueglichRand_Oben;
var Dokument_ScrollBereichHoehe;
var Dokument_Breite;
var Dokument_Hoehe;
var Dokument_Breite_Sichtbar;
var Dokument_Hoehe_Sichtbar;
var MausPositionImDokument_X;
var MausPositionImDokument_Y;
var Bild_Position_LinkeObereEcke_X;
var Bild_Position_LinkeObereEcke_Y;
var Bild_Position_RechteUntereEcke_X;
var Bild_Position_RechteUntereEcke_Y;
var Bild_NichtSichtbar; // boolean
var BildObjekt;

// ##### Funktionen
function OnMouseMoveHandler()
{
    // +++++ aktuelle Angaben holen zum Dokument
    Dokument_ScrollPosition_BezueglichRand_Links = document.body.scrollLeft;
    Dokument_ScrollPosition_BezueglichRand_Oben  = document.body.scrollTop;
    Dokument_ScrollBereichHoehe                  = document.body.scrollHeight;

    Dokument_Breite          = document.body.clientWidth;
    Dokument_Hoehe           = document.body.clientHeight;
    Dokument_Breite_Sichtbar = Dokument_ScrollPosition_BezueglichRand_Links + Dokument_Breite;
    Dokument_Hoehe_Sichtbar  = Math.max(Dokument_ScrollBereichHoehe, Dokument_Hoehe);
    // im leeren Dokument ist Dokument_ScrollBereichHoehe gleich Bild_Hoehe

```



```

//      daher muss Math.max kodiert werden

// +++++ aktuelle Mausposition im Dokument holen
MausPositionImDokument_X = event.clientX;
MausPositionImDokument_Y = event.clientY;

// +++++ neue Bildposition ermitteln
//      Grafiksystem: linke obere Ecke ist der Ursprung (0,0)
//      Bild erscheint RECHTS neben dem Mauszeiger

// ----- linke obere Ecke
Bild_Position_LinkeObereEcke_X = Dokument_ScrollPosition_BezueglichRand_Links
                                + MausPositionImDokument_X
                                + Bild_AbstandZurMausSpitze_X;

Bild_Position_LinkeObereEcke_Y = Dokument_ScrollPosition_BezueglichRand_Oben
                                + MausPositionImDokument_Y
                                + Bild_AbstandZurMausSpitze_Y;

// ----- rechte untere Ecke
Bild_Position_RechteUntereEcke_X = Bild_Position_LinkeObereEcke_X + Bild_Breite + 1;
                                // wenn nicht + 1 kodiert, so könnte Scrollbalken horizontal erzeugt werden

Bild_Position_RechteUntereEcke_Y = Bild_Position_LinkeObereEcke_Y + Bild_Hoehe + 1;
                                // wenn nicht + 1 kodiert, so könnte Scrollbalken vertikal erzeugt werden

// +++++ prüfen ob Bild bzw. Mauszeiger den sichtbaren Bereich des Dokumentes NICHT verlässt,
//      Bild erscheint RECHTS neben dem Mauszeiger
if ( (MausPositionImDokument_X > Dokument_ScrollPosition_BezueglichRand_Links)
    && (MausPositionImDokument_Y > Dokument_ScrollPosition_BezueglichRand_Oben)
    && (Bild_Position_RechteUntereEcke_X < Dokument_Breite_Sichtbar)
    && (Bild_Position_RechteUntereEcke_Y < Dokument_Hoehe_Sichtbar)
    )
{
    // Bild bzw. Mauszeiger verlässt NICHT den sichtbaren Bereich des Dokumentes

    //      DIV des Bildes neu positionieren nur wenn sichtbar:
    //      Auch wenn Bild unsichtbar ist, so können bei Neupositionierung Scrollbalken erzeugt werden,
    //      wenn das Bild die Grenzen des sichtbaren Bereiches überschreitet.
    ID_Div.style.left= Bild_Position_LinkeObereEcke_X;
    ID_Div.style.top = Bild_Position_LinkeObereEcke_Y;

    // Bild einblenden
    if (Bild_NichtSichtbar)
    {
        Bild_NichtSichtbar=false;
        ID_Div.style.visibility="visible";
    }
}
else
{
    // Bild ausblenden, da Bild bzw. Mauszeiger den sichtbaren Bereich des Dokumentes verlässt
    ID_Div.style.visibility="hidden";
    Bild_NichtSichtbar=true;
}
}

// ##### Code der mit Laden des Dokumentes abgearbeitet wird

// +++++ Bild vorladen
BildObjekt = new Image(Bild_Breite,Bild_Hoehe);
BildObjekt.src = Bild_Name;

// +++++ Bild unsichtbar anzeigen
Bild_NichtSichtbar=true;

document.write( '<DIV ID="ID_Div"'
+              ' STYLE="position:absolute;'
+              'left:0px;'

```




```

+           'top:0px;'
+           'width:' + Bild_Breite + 'px;'
+           'height:' + Bild_Hoehe + 'px;'
+           'visibility:hidden' // unsichtbar, da erst im OnMouseMove-Handler
                                // sichtbar gemacht werden soll
+           ""
+       '>'
+       '<IMG ID="ID_Img"'
+           'WIDTH=' + Bild_Breite
+           'HEIGHT=' + Bild_Hoehe
+           'BORDER=0'
+           'GALLERYIMG="no"'
+       '>'
+   '</DIV>'
+   );

```

```

ID_Img.src=BildObjekt.src;
// Eventhandler zuweisen
document.onmousemove=OnMouseMoveHandler;
</SCRIPT>
</HEAD>
<BODY></BODY>
</HTML>

```

19.4. *Mauscursor als Bild anhand cur-Datei erweitern*

cur-Datei: 32x32 Pixel, 256 Farben, 8 Bit pro Pixel

Beispiel 1:

```

<style>
<!--
BODY{
cursor:url("cursor.cur");
}
-->
</style>

```

Beispiel 2:

```
objektzeiger.style.cursor='url(cursor.cur)';
```

Beispiel 3:

```

<HTML>
<BODY>
<SCRIPT>
function SpanMausOver(){ID_span.style.cursor="url(cursor.cur)";}

function SpanMausOut(){ID_span.style.cursor="auto";}
</SCRIPT>
<BUTTON VALUE="">
<IMG SRC="au.gif">
<SPAN ID="ID_span" onmouseover="SpanMausOver();" onmouseout="SpanMausOut();">text_des_buttons</SPAN>
</BUTTON>
</BODY>
</HTML>

```

19.5. *Mauscursor mit Uhr und Datum*

```

<HTML>
<BODY>
<SCRIPT language=JavaScript>

```

```

// +++++ vom Programmierer einstellbare Variablen
var WochentagNamenFeld=new Array
('So.', // immer ab Sonntag
'Mo.',
'Di.',
'Mi.',
'Do.',
'Fr.',
'Sa.')

```



```

);
var MonatsNamenFeld=new Array
('Januar',           // immer ab Januar
'Februar',
'März',
'April',
'Mai',
'Juni',
'Juli',
'August',
'September',
'Oktober',
'November',
'Dezember'
);
var Uhr_HoeheUndBreite=40;           // durch 4 ganz teilbar
var Uhr_Ziffern_String='.-..|.-..||'; // Symbole mit Blanktrennung
// Symbole sollten zentriert-symmetrisch sein
// zB. + aber nicht _
// bei nicht-symmetrischen Symbolen muss ev. angepasst werden
// AlleZeiger_PosY_Korrektur
// Ursprung aller Zeiger um Differenz senkrecht
// korrigieren, austesten
// AlleZeiger_PosX_Korrektur
// Ursprung aller Zeiger um Differenz waagerecht
// korrigieren, austesten

var Uhr_AbstandVonMaus_Y=0;
var Uhr_AbstandVonMaus_X=100;
var StundenZeiger_String='...';      // Stundenzeiger
var StundenZeiger_Farbe='red';
var MinutenZeiger_String='....';     // Minutenzeiger
var MinutenZeiger_Farbe='blue';
var SekundenZeiger_String='.....'; // Sekundenzeiger
var SekundenZeiger_Farbe='green';
var AlleZeiger_PosY_Korrektur=-3;    // Ursprung aller Zeiger um Differenz senkrecht korrigieren,
// austesten, siehe Uhr_Ziffern_String
var AlleZeiger_PosX_Korrektur=-1;    // Ursprung aller Zeiger um Differenz waagerecht korrigieren,
// austesten, siehe Uhr_Ziffern_String

var DatumRingZeichenAnzahl=25;      // > 24
var Position_SchrittWeite_Faktor=0.5;
var DatumRing_DrehSpeed=0.02;       // je höher um so schneller
var DatumRing_DrehRichtung=false;  // true so entgegen Uhrzeigersinn und Schrift scrollt vorwärts
// false so im Uhrzeigersinn und Schrift scrollt rückwärts

// +++++ interne Variablen

var DatumRing_Elemente_StringFeld;
var DatumRing_Elemente_StringFeldLaenge=0;
var StundenZeiger_Elemente_StringFeld;
var StundenZeiger_Elemente_StringFeldLaenge=0;
var MinutenZeiger_Elemente_StringFeld;
var MinutenZeiger_Elemente_StringFeldLaenge=0;
var SekundenZeiger_Elemente_StringFeld;
var SekundenZeiger_Elemente_StringFeldLaenge=0;
var Uhr_Ziffern_StringFeld;
var Uhr_Ziffern_StringFeldLaenge=0;
var UhrSymbol_BogenWinkelInGrad=0;
var DatumRing_Element_BogenWinkelInGrad=0;
var Uhr_ElementImContainer_HoeheUndBreite=Uhr_HoeheUndBreite/4; // Uhr_HoeheUndBreite ist ganz durch 4 teilbar
X_Konstante1=Math.PI;
X_Konstante2=Math.PI/180;
var DatumRing_PosY;                 // Datum-Ring aktuelle PosY
var DatumRing_PosX;                 // Datum-Ring aktuelle PosX
var UhrZiffernRing_PosY;            // Uhr-Ziffern-Ring aktuelle PosY
var UhrZiffernRing_PosX;            // Uhr-Ziffern-Ring aktuelle PosX
var DatumRing_Container;            // Datum-Ring-Container
var DatumRing_ElementFeld;          // Datum-Ring-Container Elementfeld
var UhrZiffernRing_Container;        // Uhr-Ziffern-Ring-Container
var UhrZiffernRing_ElementFeld;      // Uhr-Ziffern-Ring-Container Elementfeld

```



```

var StundenZeiger_Container;           // Stundenzeiger-Container
var StundenZeiger_ElementeFeld;       // Stundenzeiger-Container   Elementefeld
var MinutenZeiger_Container;          // Minutenzeiger-Container
var MinutenZeiger_ElementeFeld;       // Minutenzeiger-Container   Elementefeld
var SekundenZeiger_Container;         // Sekundenzeiger-Container
var SekundenZeiger_ElementeFeld;      // Sekundenzeiger-Container   Elementefeld
var X_EventHandler_OnMousMove_UhrPosX=0;
var X_EventHandler_OnMousMove_UhrPosY=0;
var TickenZ1;
var TickenZ2;
var TickenW1=0;
var TickenW2=0;
var TickenW3=0;
var TickenW4=0;
var TickenW5=0;
var TickenW6=0;                       // aktuelle Drehposition auf Sinus bzw. Cosinuskurve also auf Kreis
var TickenK1=-1.57;
var TickenK2=-1.575;
var TickenK3=-1.0471;
var TickenK4=0;
var TickenK5=0;
var TickenK6=0;
var TickenS1="";
var TickenS2="";
var UhrInitErfolgreich=false;         // true so Uhrinit erfolgreich

// ----- onmousemove-Eventhandler
function EventHandler_OnMousMove(X00)
// X00 ist Platzhalter
{
    X_EventHandler_OnMousMove_UhrPosY=window.event.y+Uhr_AbstandVonMaus_Y;
    X_EventHandler_OnMousMove_UhrPosX=window.event.x+Uhr_AbstandVonMaus_X;
}

// ----- Datum-Ring-String ermitteln (aktuelles Datum mit ermitteln)
function DatumRingStringErmitteln()
{var X00=' ';                          // muss Leerzeichen sein wegen Ringform
var X01;
var X02=0;
var X03="";
var X04=0;

// +++++ Date holen
X01=new Date();
// +++++ Wochentag und Tag nach DatumRingString
// ----- Wochentag
X02=X01.getDay();                     // Tag numerisch
X00+=WochentagNamenFeld[X02];         // Wochentagname
X00+=' ';
// ----- Tag
X03="";
X02=X01.getDate();                   // Tag nach String
if(X02<10){X03='0';}                 // immer mit Vornull
X03+=X02.toString();                 // also 2-stellig
X00+=X03;
X00+=' ';
// +++++ Monat nach DatumRingString
X02=X01.getMonth();
X00+=MonatsNamenFeld[X02];           // Monatsname
X00+=' ';
// +++++ Jahr nach DatumRingString
X02=X01.getYear();                   // Jahr numerisch 4 stellig
if(X02<2000){X02=X02+1900;}          // ab 2000, sonst erst 4-stellig machen
X00+=X02.toString();
// +++++ Länge auffüllen
X01=X00.length;
if(X01<DatumRingZeichenAnzahl)
{for(X04=X01;X04<DatumRingZeichenAnzahl;X04++){X00+=' ';}}
return X00;

```



```

}

// ---- Uhr rekursiv animieren
function Ticken()           // interne Routine, rekursiv
{
    // +++++ prüfen ob Uhr initialisiert ist
    if(UhrInitErfolgreich)
    {
        // +++++ Datumring inhaltlich aktualisieren
        // ---- aktuelles Datum holen
        TickenS2=DatumRingStringErmitteln();
        // ---- wenn das neuer ist als das zuletzt geholte dann Datumring neu füllen
        if(TickenS2!=TickenS1)
        {
            // ---- aktuelles als letztes merken
            TickenS1=TickenS2;
            // ---- Datum-Uhrzeit-Ring füllen
            for(TickenW4=0;TickenW4<DatumRing_Elemente_StringFeldLaenge;TickenW4++) // Anzahl der Elemente ist konstant egal welches Datum
            {
                DatumRing_ElementeFeld[TickenW4].innerText=DatumRing_Elemente_StringFeld[TickenW4];
            }

            // +++++ Datum-Ring Position ermitteln
            // Es muss die Berechnung mit Gleitkommazahlen erfolgen da ansonsten ungenau Animation
            // .top und .left können nur ganzzahlige Pixel gerendert werden
            // wobei der Browser automatisch ganzzahlig setzt
            // ---- Startposition Y des 0. Ringelementes (Index 0)
            TickenW5=X_EventHandler_OnMouseMove_UhrPosY-DatumRing_PosY[0];           // aktuelle Pos - letzte Pos
            TickenW5*=Position_SchrittWeite_Faktor;                               // mal Schrittweite
            DatumRing_PosY[0]+=TickenW5;                                           // als neue Pos merken

            // ---- Startposition X des 0. Ringelementes (Index 0)
            TickenW5=X_EventHandler_OnMouseMove_UhrPosX-DatumRing_PosX[0];
            TickenW5*=Position_SchrittWeite_Faktor;
            DatumRing_PosX[0]+=TickenW5;

            // ---- ab 1. Ringelement in Abhängigkeit von Pos des Vorgänger-Ring-Elementes, also kumulativ
            for(TickenW4=1;TickenW4<DatumRing_Elemente_StringFeldLaenge;TickenW4++)
            {
                // ---- Position Y
                TickenW5=DatumRing_PosY[TickenW4-1]-DatumRing_PosY[TickenW4];       // aktuelle Pos Vorgänger - aktuelle Pos Nachfolger
                TickenW5*=Position_SchrittWeite_Faktor;                             // mal Schrittweite
                DatumRing_PosY[TickenW4]+=TickenW5;                                // als neue Pos merken

                // ---- Position X analog wie Position Y
                TickenW5=DatumRing_PosX[TickenW4-1]-DatumRing_PosX[TickenW4];
                TickenW5*=Position_SchrittWeite_Faktor;
                DatumRing_PosX[TickenW4]+=TickenW5;
            }

            // +++++ Uhrziffern-Ring Position ermitteln
            // Es muss die Berechnung mit Gleitkommazahlen erfolgen da ansonsten ungenau Animation
            // .top und .left können nur ganzzahlige Pixel gerendert werden
            // wobei der Browser automatisch ganzzahlig setzt
            // ---- Startposition Y des 0. Ringelementes (Index 0)
            TickenW5=X_EventHandler_OnMouseMove_UhrPosY-UhrZiffernRing_PosY[0];
            TickenW5*=Position_SchrittWeite_Faktor;
            UhrZiffernRing_PosY[0]+=TickenW5;

            // ---- Startposition X des 0. Ringelementes (Index 0)
            TickenW5=X_EventHandler_OnMouseMove_UhrPosX-UhrZiffernRing_PosX[0];
            TickenW5*=Position_SchrittWeite_Faktor;
            UhrZiffernRing_PosX[0]+=TickenW5;

            // ---- ab 1. Ringelement in Abhängigkeit von Pos des Vorgänger-Ring-Elementes, also kumulativ
            for(TickenW4=1;TickenW4<Uhr_Ziffern_StringFeldLaenge;TickenW4++)
            {
                // ---- Position Y
                TickenW5=UhrZiffernRing_PosY[TickenW4-1]-UhrZiffernRing_PosY[TickenW4];
                TickenW5*=Position_SchrittWeite_Faktor;
                UhrZiffernRing_PosY[TickenW4]+=TickenW5;

                // ---- Position X analog wie Position Y
            }
        }
    }
}

```



```

    TickenW5=UhrZiffernRing_PosX[TickenW4-1]-UhrZiffernRing_PosX[TickenW4];
    TickenW5*=Position_SchrittWeite_Faktor;
    UhrZiffernRing_PosX[TickenW4]+=TickenW5;
}

// +++++ Top von Container-DIV setzen, wobei Uhr-Zeiger korrigiert werden
// ----- Ring-Container
TickenW1=document.body.scrollTop;
DatumRing_Container.style.top=TickenW1;
UhrZiffernRing_Container.style.top=TickenW1;
// ----- Uhrzeiger mit Korrektur
TickenW1+=AlleZeiger_PosY_Korrektur;
StundenZeiger_Container.style.top=TickenW1;
MinutenZeiger_Container.style.top=TickenW1;
SekundenZeiger_Container.style.top=TickenW1;

// +++++ Datum-Ring positionieren
for(TickenW4=0;TickenW4<DatumRing_Elemente_StringFeldLaenge;TickenW4++)
{
    TickenZ2=DatumRing_ElementeFeld[TickenW4].style;

    TickenW1=TickenW4*TickenK4;
    TickenW1+=TickenW6;

    TickenW2=Math.sin(TickenW1)*TickenK6;
    TickenW2+=DatumRing_PosY[TickenW4];
    TickenZ2.top=TickenW2;

    TickenW2=Math.cos(TickenW1)*TickenK6;
    TickenW2+=DatumRing_PosX[TickenW4];
    TickenZ2.left=TickenW2;
}

// +++++ Uhrziffern-Ring positionieren
for(TickenW4=0;TickenW4<Uhr_Ziffern_StringFeldLaenge;TickenW4++)
{
    TickenZ2=UhrZiffernRing_ElementeFeld[TickenW4].style;

    TickenW1=TickenW4*TickenK5;
    TickenW1+=TickenK3;

    TickenW2=Math.sin(TickenW1)*Uhr_HoeheUndBreite;
    TickenW2+=UhrZiffernRing_PosY[TickenW4];
    TickenZ2.top=TickenW2;

    TickenW2=Math.cos(TickenW1)*Uhr_HoeheUndBreite;
    TickenW2+=UhrZiffernRing_PosX[TickenW4];
    TickenZ2.left=TickenW2;
}

// +++++ Date holen
TickenZ1=new Date();

// +++++ Sekundenzeiger aktualisieren
TickenW1=TickenZ1.getSeconds();
TickenW1/=30;
TickenW1*=X_Konstante1;
TickenW1+=TickenK1;
for(TickenW4=0;TickenW4<SekundenZeiger_Elemente_StringFeldLaenge;TickenW4++)
{
    TickenZ2=SekundenZeiger_ElementeFeld[TickenW4].style;
    TickenW5=TickenW4*Uhr_ElementImContainer_HoeheUndBreite;

    TickenW2=TickenW5*Math.sin(TickenW1);
    TickenW2+=UhrZiffernRing_PosY[TickenW4];
    TickenZ2.top=TickenW2;

    TickenW2=TickenW5*Math.cos(TickenW1);
    TickenW2+=UhrZiffernRing_PosX[TickenW4];
    TickenW2+=AlleZeiger_PosX_Korrektur;
    TickenZ2.left=TickenW2;
}

```



```

}

// +++++ Minutenzeiger aktualisieren VOR Stunden !
TickenW1=TickenZ1.getMinutes();
TickenW3=TickenW1;          // und retten für Stunden
TickenW1/=30;
TickenW1*=X_Konstante1;
TickenW1+=TickenK1;
for(TickenW4=0;TickenW4<MinutenZeiger_Elemente_StringFeldLaenge;TickenW4++)
{
    TickenZ2=MinutenZeiger_ElementeFeld[TickenW4].style;
    TickenW5=TickenW4*Uhr_ElementImContainer_HoeheUndBreite;

    TickenW2=TickenW5*Math.sin(TickenW1);
    TickenW2+=UhrZiffernRing_PosY[TickenW4];
    TickenZ2.top=TickenW2;

    TickenW2=TickenW5*Math.cos(TickenW1);
    TickenW2+=UhrZiffernRing_PosX[TickenW4];
    TickenW2+=AlleZeiger_PosX_Korrektur;
    TickenZ2.left=TickenW2;
}

// +++++ Stundenzeiger aktualisieren
TickenW3/=360;
TickenW3*=X_Konstante1;
TickenW1=TickenZ1.getHours();
TickenW1/=6;
TickenW1*=X_Konstante1;
TickenW1+=TickenW3;
TickenW1+=TickenK2;
for(TickenW4=0;TickenW4<StundenZeiger_Elemente_StringFeldLaenge;TickenW4++)
{
    TickenZ2=StundenZeiger_ElementeFeld[TickenW4].style;
    TickenW5=TickenW4*Uhr_ElementImContainer_HoeheUndBreite;

    TickenW2=TickenW5*Math.sin(TickenW1);
    TickenW2+=UhrZiffernRing_PosY[TickenW4];
    TickenZ2.top=TickenW2;

    TickenW2=TickenW5*Math.cos(TickenW1);
    TickenW2+=UhrZiffernRing_PosX[TickenW4];
    TickenW2+=AlleZeiger_PosX_Korrektur;
    TickenZ2.left=TickenW2;
}

// +++++ Drehposition aktualisieren (Wert für Sin und Cosinus verändern also für Drehung per Phase)
if(DatumRing_DrehRichtung)
{
    TickenW6=-DatumRing_DrehSpeed;    // entgegen Uhrzeigersinn und Schrift scrollt vorwärts, negative Werte
    if(TickenW6<-10){TickenW6=0;}      // Wertebereich begrenzen sonst irgendwann Runtime-Fehler
}
else
{
    TickenW6+=DatumRing_DrehSpeed;    // im Uhrzeigersinn und Schrift scrollt rückwärts, positive Werte
    if(TickenW6>10){TickenW6=0;}      // Wertebereich begrenzen sonst irgendwann Runtime-Fehler
}
// +++++ Rekursion
setTimeout("Ticken()",20);
} // Uhr initialisiert
}

// ----- init der Uhrverwaltung und Start der Uhr
function init()          // vom Programmierer aufrufbar
// liefert true wenn Uhr initialisiert und gestartet
// setzt UhrInitErfolgreich auf Returnwert
{
    var X00=false;
    var X01=0;
    var X02;
    var X03=false;
    var X04="";

```



```
// +++++ Annahme: Init nicht erfolgreich +++++
UhrInitErfolgreich=false;

// +++++ ERST globale Daten für bereitstellen +++++
// ----- Anzahl der Zeichen im Datum-Ring prüfen
if(DatumRingZeichenAnzahl<25){DatumRingZeichenAnzahl=25;}
// ----- Werte der Elemente des Stundenzeiger ermitteln
StundenZeiger_Elemente_StringFeld=StundenZeiger_String.split(""); // zeichenweise splitten nach Feld
StundenZeiger_Elemente_StringFeldLaenge=StundenZeiger_Elemente_StringFeld.length;
// ----- Werte der Elemente des Minutenzeiger ermitteln
MinutenZeiger_Elemente_StringFeld=MinutenZeiger_String.split(""); // zeichenweise splitten nach Feld
MinutenZeiger_Elemente_StringFeldLaenge=MinutenZeiger_Elemente_StringFeld.length;
// ----- Werte der Elemente des Sekundenzeiger ermitteln
SekundenZeiger_Elemente_StringFeld=SekundenZeiger_String.split(""); // zeichenweise splitten nach Feld
SekundenZeiger_Elemente_StringFeldLaenge=SekundenZeiger_Elemente_StringFeld.length;
// ----- Werte der Elemente des Datum-Ringes ermitteln
X04=DatumRingStringErmitteln();
DatumRing_Elemente_StringFeld=X04.split(""); // zeichenweise splitten nach Feld
DatumRing_Elemente_StringFeldLaenge=DatumRing_Elemente_StringFeld.length;
DatumRing_Element_BogenWinkelInGrad=360/DatumRing_Elemente_StringFeldLaenge;

// Winkel
des Boges jede Elementes
// ----- Werte der Elemente des Uhrziffern-Ringes ermitteln
Uhr_Ziffern_StringFeld=Uhr_Ziffern_String.split(' ');
Uhr_Ziffern_StringFeldLaenge=Uhr_Ziffern_StringFeld.length;
UhrSymbol_BogenWinkelInGrad=360/Uhr_Ziffern_StringFeldLaenge; // Winkel des Bogens jeder Uhrziffer

// +++++ DANN inits für Ticken() +++++
// ----- Konstanten
TickenK4=DatumRing_Element_BogenWinkelInGrad*X_Konstante2;
TickenK5=UhrSymbol_BogenWinkelInGrad*X_Konstante2;
// ----- aktuelle Drehposition des Datum- und des Uhrziffern-Ringes
TickenW6=0;
TickenK6=1.5*Uhr_HoeheUndBreite;

// ----- Felder der Positinsberechnungen init
// - - - Uhrziffern-Ring
UhrZiffernRing_PosY=new Array();
UhrZiffernRing_PosX=new Array();
for(X01=0;X01<Uhr_Ziffern_StringFeldLaenge;X01++)
{UhrZiffernRing_PosY[X01]=0;
UhrZiffernRing_PosX[X01]=0;
}
// - - - Datum-Ring
DatumRing_PosY=new Array();
DatumRing_PosX=new Array();
for(X01=0;X01<DatumRing_Elemente_StringFeldLaenge;X01++)
{DatumRing_PosY[X01]=0;
DatumRing_PosX[X01]=0;
}

// +++++ DANN: Uhr leer erzeugen +++++
// Uhr-Container gibt es nicht, da alle Uhr-Elemente eigenen Container haben, die
// selbständig animiert werden

// ----- Datum-Uhrzeit-Ring erzeugen, wird eigenständig animiert
// - - - Container-DIV erzeugen (Ring)
X02=document.createElement("DIV");
X00=(X02!=null);
if(X00)
{DatumRing_Container=document.body.appendChild(X02);
X02=DatumRing_Container.style;
X02.position='absolute';
X02.top='0px';
X02.left='0px';
// - - - Ringelemente je als DIV erzeugen
DatumRing_ElementFeld=new Array();
for(X01=0;X01<DatumRing_Elemente_StringFeldLaenge;X01++)
```



```

{X02=document.createElement('DIV');
X03=(X02!=null);
if(X03)
{DatumRing_ElementFeld[X01]=DatumRing_Container.appendChild(X02);
X02=DatumRing_ElementFeld[X01].style;
X02.position='absolute';
X02.top='0px';
X02.left='0px';
X02.height=Uhr_ElementImContainer_HoeheUndBreite;
X02.width=Uhr_ElementImContainer_HoeheUndBreite;
X02.textAlign='center';
}
else{X00=false;}
}
}

// ----- Uhr-Ziffern-Ring erzeugen, wird eigenständig animiert
if(X00)
{
// - - - Container-DIV erzeugen (Ring)
X02=document.createElement('DIV');
X00=(X02!=null);
if(X00)
{UhrZiffernRing_Container=document.body.appendChild(X02);
X02=UhrZiffernRing_Container.style;
X02.position='absolute';
X02.top='0px';
X02.left='0px';
// - - - Ringelemente je als DIV erzeugen
UhrZiffernRing_ElementFeld=new Array();
for(X01=0;X01<Uhr_Ziffern_StringFeldLaenge;X01++)
{X02=document.createElement('DIV');
X03=(X02!=null);
if(X03)
{UhrZiffernRing_ElementFeld[X01]=UhrZiffernRing_Container.appendChild(X02);
X02=UhrZiffernRing_ElementFeld[X01].style;
X02.position='absolute';
X02.top='0px';
X02.left='0px';
X02.height=Uhr_ElementImContainer_HoeheUndBreite;
X02.width=Uhr_ElementImContainer_HoeheUndBreite;
X02.textAlign='center';
X02.fontWeight='bold';
}
else{X00=false;}
}
}
}

// ----- Stundezeiger erzeugen, wird eigenständig animiert
if(X00)
{
// - - - Container-DIV erzeugen
X02=document.createElement('DIV');
X00=(X02!=null);
if(X00)
{StundenZeiger_Container=document.body.appendChild(X02);
X02=StundenZeiger_Container.style;
X02.position='absolute';
X02.top='0px';
X02.left='0px';
// - - - Ringelemente je als DIV erzeugen
StundenZeiger_ElementFeld=new Array();
for(X01=0;X01<StundenZeiger_Element_StringFeldLaenge;X01++)
{X02=document.createElement('DIV');
X03=(X02!=null);
if(X03)
{StundenZeiger_ElementFeld[X01]=StundenZeiger_Container.appendChild(X02);
X02=StundenZeiger_ElementFeld[X01].style;
X02.position='absolute';
X02.top='0px';

```




```

X02.left='0px';
X02.height=Uhr_ElementImContainer_HoeheUndBreite;
X02.width=Uhr_ElementImContainer_HoeheUndBreite;
X02.textAlign='center';
X02.fontWeight='bold';
X02.color=StundenZeiger_Farbe;
}
else{ X00=false;}
}
}
}

// ----- Minutenzeiger erzeugen, wird eigenständig animiert
if(X00)
{
  // - - - Container-DIV erzeugen
  X02=document.createElement('DIV');
  X00=(X02!=null);
  if(X00)
  {
    MinutenZeiger_Container=document.body.appendChild(X02);
    X02=MinutenZeiger_Container.style;
    X02.position='absolute';
    X02.top='0px';
    X02.left='0px';
    // - - - Ringelemente je als DIV erzeugen
    MinutenZeiger_ElementeFeld=new Array();
    for(X01=0;X01<MinutenZeiger_Elemente_StringFeldLaenge;X01++)
    {
      X02=document.createElement('DIV');
      X03=(X02!=null);
      if(X03)
      {
        MinutenZeiger_ElementeFeld[X01]=MinutenZeiger_Container.appendChild(X02);
        X02=MinutenZeiger_ElementeFeld[X01].style;
        X02.position='absolute';
        X02.top='0px';
        X02.left='0px';
        X02.height=Uhr_ElementImContainer_HoeheUndBreite;
        X02.width=Uhr_ElementImContainer_HoeheUndBreite;
        X02.textAlign='center';
        X02.fontWeight='bold';
        X02.color=MinutenZeiger_Farbe;
      }
      else{ X00=false;}
    }
  }
}

// ----- Sekundenzeiger erzeugen, wird eigenständig animiert
if(X00)
{
  // - - - Container-DIV erzeugen
  X02=document.createElement('DIV');
  X00=(X02!=null);
  if(X00)
  {
    SekundenZeiger_Container=document.body.appendChild(X02);
    X02=SekundenZeiger_Container.style;
    X02.position='absolute';
    X02.top='0px';
    X02.left='0px';
    // - - - Ringelemente je als DIV erzeugen
    SekundenZeiger_ElementeFeld=new Array();
    for(X01=0;X01<SekundenZeiger_Elemente_StringFeldLaenge;X01++)
    {
      X02=document.createElement('DIV');
      X03=(X02!=null);
      if(X03)
      {
        SekundenZeiger_ElementeFeld[X01]=SekundenZeiger_Container.appendChild(X02);
        X02=SekundenZeiger_ElementeFeld[X01].style;
        X02.position='absolute';
        X02.top='0px';
        X02.left='0px';
        X02.height=Uhr_ElementImContainer_HoeheUndBreite;
        X02.width=Uhr_ElementImContainer_HoeheUndBreite;

```



```

    X02.textAlign='center';
    X02.fontWeight='bold';
    X02.color=SekundenZeiger_Farbe;
  }
  else{X00=false;}
}
}
}

// +++++ DANN Datum-Uhrzeit-Ring füllen +++++
if(X00)
{
  // ----- Datum-Ring füllen
  for(X01=0;X01<DatumRing_Elemente_StringFeldLaenge;X01++)
  {
    DatumRingZeichenAnzahl
    {DatumRing_ElementeFeld[X01].innerText=DatumRing_Elemente_StringFeld[X01];}
  }
  // ----- Uhrziffern-Ring füllen
  for(X01=0;X01<Uhr_Ziffern_StringFeldLaenge;X01++)
  {UhrZiffernRing_ElementeFeld[X01].innerText=Uhr_Ziffern_StringFeld[X01];}
  // ----- Stundezeiger füllen
  for(X01=0;X01<StundenZeiger_Elemente_StringFeldLaenge;X01++)
  {StundenZeiger_ElementeFeld[X01].innerText=StundenZeiger_Elemente_StringFeld[X01];}
  // ----- Minutenzeiger
  for(X01=0;X01<MinutenZeiger_Elemente_StringFeldLaenge;X01++)
  {MinutenZeiger_ElementeFeld[X01].innerText=MinutenZeiger_Elemente_StringFeld[X01];}
  // ----- Sekundenzeiger füllen
  for(X01=0;X01<SekundenZeiger_Elemente_StringFeldLaenge;X01++)
  {SekundenZeiger_ElementeFeld[X01].innerText=SekundenZeiger_Elemente_StringFeld[X01];}
}

// +++++ DANN onmousemove-Handler einbinden
if(X00){X00=document.body.attachEvent('onmousemove',EventHandler_OnMousMove);}

// +++++ DANN global merken ob Init erfolgreich +++++
UhrInitErfolgreich=X00;

// +++++ DANN Uhr rekursiv animieren +++++
if(X00){Ticken();} // rekursiv

return X00;
}

alert(init());
</SCRIPT>
</BODY>
</HTML>

```

20. Style Sheets und DHTML

20.1. Cascading Style Sheets (CSS) kodieren

20.1.1. CSS-Klassen erzeugen

Beispiel 1

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      H1.meineKlasse { color: red; }
    </STYLE>
  </HEAD>
  <BODY>
    <H1 class="meineKlasse">Eine &Uuml;berschrift</H1>
    <H1>Noch eine &Uuml;berschrift</H1>
    <H1 class="meineKlasse">Eine weitere &Uuml;berschrift</H1>
  </BODY>
</HTML>

```

Beispiel 2



```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      .rot {      color: red;
                font-style: italic; }
    </STYLE>
  </HEAD>

  <BODY>
    <H1 class="rot">Eine &Uuml;berschrift</H1>
    Normaler Text.
    <P class="rot">Dies ist ein Paragraph.</P>
  </BODY>
</HTML>

```

Beispiel 3

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      BODY { background: black; }
      H1   { color: blue; }
    </STYLE>
  </HEAD>

  <BODY>
    <H1>Eine &Uuml;berschrift</H1>
  </BODY>
</HTML>

```

Beispiel 4 Entfernen des Strichs unter einem Text-Link

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      A { text-decoration: none; }
    </STYLE>
  </HEAD>

  <BODY>
    <A HREF="http://www.test.de/">Ein Link ohne Unterstreich</A>
  </BODY>
</HTML>

```

20.1.2. Tag-unabhängige Kodierung (mit ID)**20.1.2.1. allgemein**

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #einElement { color: red; }
    </STYLE>
  </HEAD>

  <BODY>
    <H1 ID="einElement">Eine &Uuml;berschrift</H1>
    <H1>Noch eine &Uuml;berschrift</H1>
  </BODY>
</HTML>

```

20.1.2.2. mit Tag

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #test { color: red; }
    </STYLE>
  </HEAD>

  <BODY>

```



```

        Dies ist ein <SPAN id="test">Test</SPAN>...
    </BODY>
</HTML>

```

20.1.2.3. mit <DIV> Tag

Beispiel 1

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #test { color: red; }
    </STYLE>
  </HEAD>
  <BODY>
    Dies ist ein <DIV id="test">Test</DIV>...
  </BODY>
</HTML>

```

Beispiel 2 Positionierung eines Textes

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #element1 {
        position: absolute;
        p: 50px;
        ft: 100px; }
    </STYLE>
  </HEAD>
  <BODY>
    <DIV="element1">Cascading Style Sheets!</DIV>
  </BODY>
</HTML>

```

Beispiel 3 Positionierung eines Bildes

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #bild {
        position: absolute;
        p: 50px;
        ft: 100px; }
    </STYLE>
  </HEAD>
  <BODY>
    <DIV ID="bild">
      <IMG
        RC="obj.gif"
        WIDTH="278"
        HEIGHT="216">
    </DIV>
  </BODY>
</HTML>

```

20.1.2.4. mit <Hx> Tag

Beispiel Überschrift mit permanentem Farbwechsel

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      <!--
        # farbwechsel { color: #FF0000; }
      //-->
    </STYLE>

    <SCRIPT LANGUAGE="JavaScript1.2" TYPE="text/javascript">
      <!--
        var farbwechsel_anzahl = 32;          // Anzahl der Schritt von
                                                //Start bis Ende

```



```

var farbwert_start = [255, 0, 0]           // Rot, Grün, Blau
var farbwert_ende   = [0, 255, 255]        // Rot, Grün, Blau
var farbwechsel_tempo = 25                 // in ms zwischen zwei Schritten

var farbwechsel_zahler = 0
var farbwert_wechselrichtung = 0

var hexaziffern_feld =
[
    "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
    "A", "B", "C", "D", "E", "F"
]

function dezimalwert_zu_hexaziffern(dezimal_wert)
{
    var high = Math.floor(dezimal_wert / 16);
    var low  = Math.floor(dezimal_wert - (high * 16));

    return(hexaziffern_feld[high] + "" + hexaziffern_feld[low]);
}

function farbewechseln(ueberschrift_id)
{
    var i
    var farbe_dezimal = new Array(3);           // Rot, Grün, Blau

    for(i = 0; i < 3; i++)
    {
        farbe_dezimal[i] =
            farbwert_start[i]
            - (
                (farbwert_start[i] - farbwert_ende[i])
                * farbwechsel_zahler
                / farbwechsel_anzahl
            )
    }

    if(farbwechsel_zahler >= farbwechsel_anzahl)
    { farbwert_wechselrichtung = -1 }
    else
    {
        if(farbwechsel_zahler <= 1)
        { farbwert_wechselrichtung = 1 }
    }

    farbwwechsel_zahler += farbwert_wechselrichtung

    var farbe_hexa =
        "#"
        + dezimalwert_zu_hexaziffern(farbe_dezimal[0])
        + dezimalwert_zu_hexaziffern(farbe_dezimal[1])
        + dezimalwert_zu_hexaziffern(farbe_dezimal[2]);

    if(document.ids)
    { document.ids[ueberschrift_id].color = farbe_hexa }           // NS
    else
    { document.all[ueberschrift_id].style.color = farbe_hexa }     // IE

    id = window.setTimeout(
        "farbewechseln('" + ueberschrift_id + "')", farbwwechsel_tempo);
    );
}

function start( ueberschrift_id,
                startfarbe_rot, startfarbe_gruen, startfarbe_blau,
                endfarbe_rot, endfarbe_gruen, endfarbe_blau
                )
{
    farbwert_start[0] = startfarbe_rot
    farbwert_start[1] = startfarbe_gruen
    farbwert_start[2] = startfarbe_blau
    farbwert_ende[0] = endfarbe_rot

```



```

        farbwert_ende[1] = endfarbe_gruen
        farbwert_ende[2] = endfarbe_blau
        farbwechsel_zahler = 0
        farbwechselIn(ueberschrift_id)
    }
    //-->
</SCRIPT>
</HEAD>

<BODY ... onLoad="start('farbwechsel', 0, 255, 0, 255, 0, 255)">
.....
    <H1 ID="farbwechsel">&Uuml;berschrift</H1>
</BODY>
</HTML>

```

20.1.2.5. Clipping (Ausschnitt erzeugen)

Beispiel 1

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #bild {
        position: absolute;
        p: 50px;
        ft: 100px;
        ip: rect(40px 200px 150px 40px); }
    </STYLE>
  </HEAD>
  <BODY>
    <DIV ID="bild">
      <IMG
        RC="obj.gif"
        WIDTH="278"
        HEIGHT="216">
    </DIV>
  </BODY>
</HTML>

```

Beispiel: Grafik-Thumbnails

6 Bilder in je zwei Dimensionen

Klein als Thumbnail: bklein1.gif bis bklein5.gif

bklein0.gif ist das Ausgangsbild und kann nicht gethumbnailt werden

enthält z.B. eine allgemeine Grafik

wird durch Thumbnail von bild1-5 ersetzt

bklein0 bis 5 mit gemeinsamen Dimensionen: Breite = 160, Höhe=120

Gross als maximal (normal): HTML-Datei b0.html bis b5.html

stellen das jeweilige Bild in 320x240 als maximale Größe dar,

also **nicht** bklein0.gif bis bklein5.gif

müssen vorgefertigt sein (beliebiger Aufbau)

Wichtig ist die Nummernfolge 0 bis 5, die als Feldindex und für einen Zähler verwendet werden.

Wird der Bilddlink angeklickt, so maximiert sich das Bild per b0.html bis b5.html

Wird mit der Maus über den Link gefahren, so wird ein Bildausschnitt von bklein0 bis 5.gif

vergrößert dargestellt laut Darstellungsvorgabe der CSS-Klasse clip_klasse

Wird mit der Maus der Link verlassen nach dem Überfahren, so wird das Bild wieder in Kleindarstellung

also laut bklein0 bis 5.gif angezeigt.

```

<HTML>
<HEAD>
  <TITLE> ... <TITLE>
  <STYLE TYPE="text/css">
    <!--      h1 {position:absolute;top:10px;left:160px;}
              ..clip_klasse {position:absolute;background-color:#FFDD80;
              width:100px;border-style:none;border-width:thin;
              clip:rect(0px 200px 20px auto);}
    </-->
  </STYLE>
  <SCRIPT LANGUAGE="JavaScript">
  <!--

```



```

        bild_feld=new Array(6);
        for (i=0; i<=5; i++)
        {
            bild_feld [i]=new Image();           // bklein0 bis 5.gif in RAM laden
        bild_feld [i].SRC="bklein* + i +".gif";    // beim Laden der HTML-Seite
        }
        function los_mit_bild0()                  // Bild0 als Ausgangsbild anzeigen
                                                // das ersetzt, also gethumbnailt wird
        {
            thumbnail_fenster =window.open("b0.html", "bilder");
                                                // mit Laden der HTML-Seite

            thumbnail_fenster.resizeTo(320,240);
            thumbnail_fenster.moveTo(1,1);
            self.focus();
        }

        function clipping(bild_im_dokument,startX,startY,endX,endY)
        {
            if (document.all)
            {
                // Internet Explorer
                document.all[bild_im_dokument].style.clip=
                    "rect("+startY+" "+endx+" "+endY+" "+startX+")";
            }
            else
            {
                // Netscape
                if (document.layers)
                {
                    document.layers[bild_im_dokument].clip.left=startX;
                    document.layers[bild_im_dokument].clip.top=startY;
                    document.layers[bild_im_dokument].clip.rigth=endX;
                    document.layers[bild_im_dokument].clip.bottom=endY;
                }
            }
        }
    }

```



```

function bild_ausschnitt_vergroessern(bild_datei_nr)
{
    bild_nr_im_dokument="bild"+ bild_datei_nr; // Bild im Dokument referenzieren
                                                // anand Bildnummer der Grafikdatei

    clipping (bild_nr_im_dokument,0,0,200,200); // Ausschnitt vergrößern
    // bklein0.gif ersetzen
    if(document.all)
    {
        // Internet Explorer
        document.zu_ersetzendes_bild.SRC= bild_feld [bild_datei_nr].src;

    }
    else
    {
        // Netscape
        if (document.layers)
        {document.fuer_netscape.document.zu_ersetzendes_bild.SRC=
            bild_feld [bild_datei_nr].src;

        }

    }
}

function bild_ausschnitt_verkleinern(bild_datei_nr)
{
    bild_nr_im_dokument="bild"+ bild_datei_nr; // Bild im Dokument referenzieren
                                                // anand Bildnummer der Grafikdatei

    clipping (bild_nr_im_dokument,0,0,200,20); // Ausschnitt verkleinern

    // wieder bklein0.gif anzeigen, also Ersetzung aufheben
    if(document.all)
    {
        // Internet Explorer
        document.zu_ersetzendes_bild.SRC= bild_feld [0].src;

    }
    else
    {
        if(document.layers)
        {
            // Netscape
            document.fuer_netscape.document.zu_ersetzendes_bild.SRC=
                bild_feld [0].src;

        }

    }
}

function bild_maximieren(html_datei_nr)
{
    thumbnail_fenster.location.HREF="b"+html_datei_nr+".html";
    thumbnail_fenster.focus();
}

//-->
</SCRIPT>
</HEAD>
<BODY
    TEXT="#000000"
    BGCOLOR="#FFFF98"
    LINK="#FF0000"
    ALINK="#FF0000"
    VLINK="#FF0000"
    onLoad="los()"
>

    <H1>freier_text</H1> // Ausgabe nach CSS-Classe für H1
    <DIV ID="fuer_netscape"
        STYLE="position:absoute;top:110px;left:185px;"
    >

        <IMG SRC="bklein0.gif" // wird durch Thumbnails ersetzt
            NAME="zu_ersetzendes_bild"
            BORDER="1"
            WIDTH="160"
            HEIGHT="120"
            ALT="zu_ersetzendes_bild"

        >

    </DIV>

```




```

<DIV ID="bild1"
CLASS="clip_klasse"
STYLE="top:60;left:10;"
>
  <A HREF="javascript:bild_maximieren(1)"
onMouseover="bild_ausschnitt_vergroessern(1)"
onMouseout="bild_ausschnitt_verkleinern(1)"
>Bild 1
  </A><BR>
  kurzbeschreibung zu bild1
</DIV>
<DIV ID="bild2"
CLASS="clip_klasse"
STYLE="top:60;left:400;"
>
  <A HREF="javascript:bild_maximieren(2)"
onMouseover="bild_ausschnitt_vergroessern(2)"
onMouseout="bild_ausschnitt_verkleinern(2)"
>Bild 2
  </A><BR>
  kurzbeschreibung zu bild2
</DIV>
<DIV ID="bild3"
CLASS="clip_klasse"
STYLE="top:250;left:10;"
>
  <A HREF="javascript:bild_maximieren(3)"
onMouseover="bild_ausschnitt_vergroessern(3)"
onMouseout="bild_ausschnitt_verkleinern(3)"
>Bild 3
  </A><BR>
  kurzbeschreibung zu bild3
</DIV>
<DIV ID="bild4"
CLASS="clip_klasse"
STYLE="top:250;left:400;"
>
  <A HREF="javascript:bild_maximieren(4)"
onMouseover="bild_ausschnitt_vergroessern(4)"
onMouseout="bild_ausschnitt_verkleinern(4)"
>Bild 4
  </A><BR>
  kurzbeschreibung zu bild4
</DIV>
<DIV ID="bild5"
CLASS="clip_klasse"
STYLE="top:350;left:200;"
>
  <A HREF="javascript:bild_maximieren(5)"
onMouseover="bild_ausschnitt_vergroessern(5)"
onMouseout="bild_ausschnitt_verkleinern(5)"
>Bild 5
  </A><BR>
  kurzbeschreibung zu bild5
</DIV>
</BODY>
</HTML>

```

20.1.2.6. z-Index

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #bild1 { position: absolute;
              top: 50px;
              left: 100px;
              z-index: 1; }
      #bild2 { position: absolute;
              top: 100px;
              left: 140px;

```



```

                                z-index: 2; }
</STYLE>
</HEAD>
<BODY>
  <DIV ID="bild1">
    <IMG SRC="obj1.gif"
        WIDTH="136"
        HEIGHT="135">
  </DIV>
  <DIV ID="bild2">
    <IMG SRC="obj2.gif"
        WIDTH="172"
        HEIGHT="120">
  </DIV>
</BODY>
</HTML>

```

20.1.3. Tag-abhängige Kodierung (ohne ID)

20.1.3.1. Style Sheet Definition direkt in einem Tag

```

<HTML>
  <BODY>
    <H1 style="color: red;">Eine &Uuml;berschrift</H1>
  </BODY>
</HTML>

```

20.1.3.2. mit <LINK> Tag

```

<HTML>
  <HEAD>
    <link rel="stylesheet" HREF="test.css" TYPE="text/css">
  </HEAD>
  <BODY>
    <H1>Eine &Uuml;berschrift</H1>
  </BODY>
</HTML>

```

20.2. Das Layer-Objekt im Netscape Communicator

20.2.1. Layer-Objekt erzeugen

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #element1 {
        position: absolute;
        top: 50px;
        left: 100px; }
    </STYLE>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function verschiebe()
        {
          document.layers["element1"].left = 200;
          document.layers["element1"].top = 100;
        }
      //-->
    </SCRIPT>
  </HEAD>
  <BODY>
    <FORM>
      <INPUT TYPE="button"
        VALUE="Verschieben"
        onClick="verschiebe()">
    </FORM>
    <DIV ID="element1">JavaScript</DIV>

```



```
</BODY>  
</HTML>
```



20.2.2. Layer-Objekt bewegen

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #element1 {
        position: absolute;
        top: 10px;
        left: 10px; }
    </STYLE>

    <SCRIPT LANGUAGE="JavaScript">
      <!--
        var position = 10;

        function bewege()
        {
          document.layers["element1"].left = position;

          // Ausgabe der Position auf der Statusleiste
          status = document.layers["element1"].left;

          // position um eins erhoehen
          position++;

          // auf Ausgangsposition zuruecksetzen, wenn
          // position > 300 wird
          if (position > 300) {position = 10;}
        }
      <!-->
    </SCRIPT>
  </HEAD>

  <BODY onLoad="setInterval('bewege()',30)">
    <DIV ID="element1">JavaScript</DIV>
  </BODY>
</HTML>

```

20.2.3. Die Layer-Eigenschaft document

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #element1 {
        position: absolute;
        top: 100px;
        left: 200px; }
    </STYLE>

    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function wertAusgeben()
        {
          var doc1 = document.layers["element1"].document;
          var wert = doc1.forms["meinFormular"].eingabe.value;
          alert(wert);
        }
      <!-->
    </SCRIPT>
  </HEAD>

  <BODY>
    <FORM>
      <INPUT TYPE="button"
        VALUE="Wert ausgeben"
        onClick="wertAusgeben()">

    </FORM>

    <DIV ID="element1">

    <FORM NAME="meinFormular">
      <INPUT TYPE="text"

```



```

NAME="eingabe"
VALUE="">
</FORM>
</DIV>
</BODY>
</HTML>

```

20.2.4. JavaScript-Code und das Layer-Objekt

```

<HTML>
<HEAD>
<STYLE TYPE="text/css">
#element1 {
position: absolute;
top: 100px;
left: 200px; }
</STYLE>

<SCRIPT LANGUAGE="JavaScript">
<!--
function test()
{alert("Test 1");}
</SCRIPT>
</HEAD>

<BODY>
<DIV ID="element1">

<SCRIPT LANGUAGE="JavaScript">
<!--
function test()
{alert("Test 2");}
//-->
</SCRIPT>

<FORM NAME="meinFormular">
<INPUT TYPE="button"
VALUE="Test"
onClick="test()">

</FORM>

</DIV>
</BODY>
</HTML>

```

20.2.5. Laden neuer HTML-Seiten in ein Layer-Objekt

```

<HTML>
<HEAD>
<STYLE TYPE="text/css">
#ausgabe { position: absolute;
top: 150px;
left: 100px;
clip: rect(0px 400px 300px 0px); }
</STYLE>

<SCRIPT LANGUAGE="JavaScript">
<!--
function anzeigen(URL)
{document.layers["ausgabe"].SRC= URL;}
</SCRIPT>

</HEAD>

<BODY>
<FORM NAME="meinFormular">
<INPUT TYPE="text"
NAME="URL"
VALUE="http://">
<INPUT TYPE="button"
VALUE="Dokument laden"

```



```

onOnclick="anzeigen(this.form.URL.value)">
</FORM>
<DIV ID="ausgabe">Bitte geben Sie eine Adresse ein...</DIV>
</BODY>
</HTML>

```

20.2.6. Layer-Objekt clipping

```
<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #bild1 { position: absolute;
              top: 100px;
              left: 100px;
              clip: rect(0px 0px 0px 0px); }
    </STYLE>

    <SCRIPT LANGUAGE="JavaScript">
      <!--
        var bild1, stepX, stepY;
        var schritte = 20;
        var num = 0;

        function init()
        {
          bild1 = document.layers["bild1"];
          var bildObj = bild1.document.images["einBild"];
          stepX = Math.round(bildObj.width / schritte);
          stepY = Math.round(bildObj.height / schritte);
        }

        function bereichVeraendern()
        {
          ld1.clip.width += stepX;
          d1.clip.height += stepY;

          num++;
          if (num < schritte) { setTimeout("bereichVeraendern()", 50); }
        }
      //-->
    </SCRIPT>
  </HEAD>

  <BODY>
    <DIV ID="bild1">
      <IMG SRC="obj.gif"
          NAME="einBild"
          WIDTH="278"
          HEIGHT="216"
          onLoad="init();window.bereichVeraendern()">
    </DIV>
  </BODY>
</HTML>
```

Dynamisches Erzeugen von Layer-Objekten

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function erzeugen(formular)
        {
          // Konstruktor aufrufen
          var l = new Layer(formular.breite.value);

          // Werte setzen
          l.left = formular.x.value;
          l.top = formular.y.value;
          l.clip.height = formular.hoehe.value;
          l.zIndex = formular.zIndex.value;
        }
      </SCRIPT>
    </HEAD>
  </HTML>
```



```

        l.bgColor = formular.bgColor[
        formular.bgColor.selectedIndex].value;

        // Inhalt erzeugen
        l.document.open();
        l.document.write(formular.inhalt.value);
        l.document.close();

        // Layer anzeigen
        l.visibility= "show";
    }
    <!-->
</SCRIPT>
</HEAD>
<BODY>
    <FORM>
        left: <INPUT TYPE="text" NAME="x" VALUE="300"><BR>
        top: <INPUT TYPE="text" NAME="y" VALUE="50"><BR>
        clip.width: <INPUT TYPE="text" NAME="breite" VALUE="100"><BR>
        clip.height: <INPUT TYPE="text" NAME="hoehe" VALUE="50"><BR>
        zIndex: <INPUT TYPE="text" NAME="zIndex" VALUE="10"><BR>

        bgColor: <SELECT NAME="bgColor">
                    <OPTION VALUE="blue">blau
                    <OPTION VALUE="red">rot
                    <OPTION VALUE="yellow">gelb
                </SELECT><BR>

        Inhalt:
        <TEXTAREA NAME="inhalt" ROWS="5" COLS="30">Hallo!</TEXTAREA>
        <BR><BR>
        <INPUT TYPE="button" VALUE="Layer-Objekt erzeugen"
            onClick="erzeugen(this.form);">
    </FORM>
</BODY>
</HTML>

```

20.2.7. Verschachtelte Layer-Objekte mit dem Layer()-Konstruktor

```

<HTML>
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    <!--
        function init()
        {
            var layer1 = new Layer(200);
            layer1.left = 100;
            layer1.top = 50;
            layer1.clip.height = 100;
            layer1.bgColor = "blue";
            layer1.visibility = "show";

            var layer2 = new Layer(50, layer1);
            layer2.left = 10;
            layer2.top = 10;
            layer2.clip.height = 50;
            layer2.bgColor = "red";
            layer2.visibility = "show";
        }
    <!-->
</SCRIPT>
</HEAD>

<BODY onLoad="init()">
</BODY>
</HTML>

```

20.2.8. Das <LAYER>-Tag

```

<HTML>

```



```
<BODY>  
  <LAYER left="100" top="50" bgColor="red">  
    Das &lt;layer>-Tag.  
  </LAYER>  
</BODY>  
</HTML>
```



20.3. Das style-Objekt im Microsoft Internet Explorer

20.3.1. style-Objekt erzeugen

```
<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #obj {
        position:absolute;
        left:30;
        top:200;
        width: 300;
        z-index:1;
        font-size: 10;
        color: red}
    </STYLE>

    <SCRIPT LANGUAGE="JavaScript">
      <!--
        function verschieben()
        {
          document.all["obj"].style.pixelLeft = 100;
          document.all["obj"].style.pixelTop = 50;
        }
      <!-->
    </SCRIPT>
  </HEAD>

  <BODY>
    <FORM>
      <INPUT TYPE="button"
        VALUE="verschieben"
        onclick="verschieben()">
    </FORM>

    <DIV ID="obj">Dies ist ein positionierbares Element</DIV>
  </BODY>
</HTML>
```

20.3.2. style-Objekt bewegen

```
<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #element1 {
        position: absolute;
        top: 10px;
        left: 10px;
        width: 200; }
    </STYLE>

    <SCRIPT LANGUAGE="JavaScript">
      <!--
        var position = 10;

        function bewege()
        {
          document.all["element1"].style.pixelLeft = position;

          // Ausgabe der Position auf der Statusleiste
          status = document.all["element1"].style.pixelLeft;

          // position um eins erhoehen
          position++;

          // auf Ausgangsposition zuruecksetzen, wenn
          // position > 300 wird
          if (position > 300) {position = 10;}
        }
      <!-->
    </SCRIPT>
  </HEAD>
```



```

<BODY onLoad="setInterval('bewege()',30)">
  <DIV ID="element1">JScript</DIV>
</BODY>
</HTML>

```

20.3.3. Instant Page Update

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      .link1 { font-size: 15pt;
              text-decoration: none;
            }
      .link2 { font-size: 20pt;
              font-weight: bold;
              font-style: italic;
              letter-spacing: 3pt;
              text-decoration: none;
              color: red;
            }
    </STYLE>
  </HEAD>
  <BODY>
    <A class="link1"
      onMouseover="this.className= 'link2';"
      onMouseout="this.className= 'link1';"
      HREF="seite1.htm">Link 1</A>
    <BR>
    <A class="link1"
      onMouseover="this.className= 'link2';"
      onMouseout="this.className= 'link1';"
      HREF="seite2.htm">Link 2</A>
    <BR>
    <A class="link1"
      onMouseover="this.className= 'link2';"
      onMouseout="this.className= 'link1';"
      HREF="seite3.htm">Link 3</A>
  </BODY>
</HTML>

```

20.3.4. style-Objekt clipping

```

<HTML>
  <HEAD>
    <STYLE TYPE="text/css">
      #obj {position:absolute; left:20; top:50; width:600; z-index:1;
            clip: rect(auto 278px auto 0px); }
      #teil1 {position:absolute; left:-278; top:0; width: 278; z-index:10;}
      #teil2 {position:absolute; left:278; top:0; width: 278; z-index:20;}
    </STYLE>

    <SCRIPT LANGUAGE="JavaScript">
      <!--
        var pos1 = -278;
        var pos2 = 278;
        var schrittweite = 10;
        var timer;

        function start()
        {timer = setInterval("bewegen()", 50);}

        function bewegen()
        {
          pos1 += schrittweite;
          pos2 -= schrittweite;

          // Animation zu Ende?
          if (pos1 >= 0)
          {

```



```

        pos1 = 0;
        pos2 = 0;
        clearInterval(timer);
    }

    // Verschieben
    document.all["teil1"].style.pixelLeft = pos1;
    document.all["teil2"].style.pixelLeft = pos2;
}
//-->
</SCRIPT>
</HEAD>

<BODY onLoad="start()">
    <DIV ID="obj">
        <DIV ID="teil1"><IMG SRC="clip1.gif" WIDTH="278" HEIGHT="216"></DIV>
        <DIV ID="teil2"><IMG SRC="clip2.gif" WIDTH="278" HEIGHT="216"></DIV>
    </DIV>
</BODY>
</HTML>

```

20.4. Unterscheidung Layer und Style je nach aktuellem Browser

if document.all	true für Explorer
if document.layers	true für Netscape
if (!document.all && !document.layers)	true für WEDER Explorer NOCH Netscape

Beispiel 1

```

<HTML>
    <HEAD>
        // ***** Layer für Netscape anlegen
        <STYLE TYPE="text/css">
            #layer { position: absolute;
                    top: 50px;
                    left: 100px; }
        </STYLE>

        // ***** Style für Microsoft anlegen
        <STYLE TYPE="text/css">
            #style { position: absolute;
                    left: 30;
                    top: 200;
                    width: 300;
                    z-index: 1;
                    font-size: 10;
                    color: red }
        </STYLE>

        //***** Verschieben je nach Browser
        <SCRIPT LANGUAGE="JavaScript">
        <!--
            function verschiebe (left,top)
            {
                if (!document.all && !document.layers)
                {alert("Browser ist nicht Microsoft bzw. Netscape "
                    +"und kann nicht Layer bzw. Style behandeln.");
                }
                else
                {
                    if (navigator.appName == "Microsoft")
                    {
                        document.all["style"].style.pixelLeft = left;
                        document.all ["style"].style.pixelTop = top;
                    }
                    else
                    {
                        document.layers["layer"].left = 200;
                        document.layers ["layer"].top = 100;
                    }
                }
            }
        }
    }

```



```
    }  
</HEAD>
```



```

<BODY>
  <FORM>
    <INPUT TYPE="button"
      VALUE="Verschieben 200 Pixel links und 100 Pixel hoch"
      onClick="verschiebe(200,100)">
    </FORM>
  <DIV ID="layer">Verschiebung von Layerobjekt des Netscape</DIV>
  <DIV ID="style">Verschiebung von Styleobjekt des Explorers</DIV>
</BODY>
</HTML>

```

Beispiel 2: *schwebender Text von rechts nach links bei 640 Pixel Bildschirmbreite*

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
  // var objekt NICHT deklarierbar, da Typ der Variable je nach Browserart per Zuweisung
  // erzeugt wird

  function starte_bewegung()
  {
    if (document.layers) // Netscape
    {
      objekt=document.bewegtes_objekt; // Typ laut Netscape
      bewegen();
    }
    else
    {
      if (document.all) // Microsoft
      {
        objekt=bewegtes_objekt.style; // Typ laut Microsoft
        bewegen();
      }
    }
  }

  function bewegen() // rekursive Funktion mit setTimeout auf 200 Millisekunden Wartezeit
  {
    ganze_zahl=parseInt(objekt.left); // parseInt erzeugt ganze Zahl, also auch negativ
    objekt.left=ganze_zahl - 3; // Bewegung um 3 Pixel

    if (ganze_zahl < 0) {objekt.right=640}
    setTimeout('bewegen()',200);
  }
//-->
</SCRIPT>
</HEAD>

<BODY onload="starte_bewegung()">
// Microsoft unterstützt CSS-Layer !
<DIV ID="bewegtes_objekt"
  STYLE="position:absolute;top:100px;left:250px;z-index:2;height:50px;width:150px"
  // Achtung: STYLE alles 1 Zeile da Zeichenkettenparameter
  // keine Blanks wegen Netscape
>
<H1 ALIGN="center">bewegter text</H1>
</DIV>
</BODY>
</HTML>

```

21. Java-JavaScript Kommunikation

21.1. Gibt einen Text auf der Java Konsole aus

Javakonsole muss aktiviert sein.

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
  <!--
    function gruss()
    {

```



```

        java_objekt_system = java.lang.System;
        java_objekt_system.out.println("Gruss von JavaScript");
    }
    // -->
</SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <INPUT TYPE="button"
            VALUE="Bitte dr&uuml;cken"
            lick="gruss()">
    </FORM>
</BODY>
</HTML>

```

21.2. Zeigt Informationen zum Betriebssystem und den Java-Klassen

```

<HTML>
    <CENTER>
        <H1>Informationen, die man durch java.lang.System erh&auml;lt</H1>
        <BR>

        <SCRIPT LANGUAGE="JavaScript">
            <!--
                java_objekt_system = java.lang.System;
                betriebssystem_name = java_objekt_system.getProperty("os.name");
                betriebssystem_version = java_objekt_system.getProperty("os.version");
                computersystem = java_objekt_system.getProperty("os.arch");
                javaversion = java_objekt_system.getProperty("java.version");
                classversion = java_objekt_system.getProperty("java.class.version");
                javahersteller = java_objekt_system.getProperty("java.vendor");
                javahersteller_url = java_objekt_system.getProperty("java.vendor.url");

                // Ausgabe der Tabelle
                document.write("<TABLE WIDTH=50% border=2>");
                document.write("<TD WIDTH=50%>Betriebssystem</TD>");
                document.write("<TD WIDTH=50%>" + betriebssystem_name + "</TD><TR>");
                document.write("<TD>Versions-Nummer</TD>");
                document.write("<TD>" + betriebssystem_version + "</TD><TR>");
                document.write("<TD>Computer-System</TD>");
                document.write("<TD>" + computersystem + "</TD><TR>");
                document.write("<TD>Java-Version</TD>");
                document.write("<TD>" + javaversion + "</TD><TR>");
                document.write("<TD>Class-Version</TD>");
                document.write("<TD>" + classversion + "</TD><TR>");
                document.write("<TD>Hersteller</TD>");
                document.write("<TD>" + javahersteller + "</TD><TR>");
                document.write("<TD>URL des Herstellers</TD>");
                document.write("<TD>" + javahersteller_url + "</TD><TR>");
                document.write("</TABLE>");

            // -->
        </SCRIPT>
    </CENTER>
</HTML>

```

21.3. Programmierung eines Java-Stacks zum temporären Zwischenspeichern von Daten in den RAM

Die Reihenfolge der Datenauf dem Stack ist **UNBEDINGT** zu beachten:
 Der zuletzt auf dem Stack abgelegte Wert ist zuerst auszulesen.
 (Last in First Out = LiFo)

```

<HTML>
    <HEAD>
        <SCRIPT LANGUAGE="JavaScript">
            <!--
                // Neues Stack-Objekt erzeugen
                java_objekt_stack = new java.util.Stack();

```



```

        function in_stack_ablegen(wert)
        {java_objekt_stack.push(wert);}

        function aus_stack_holen()
        {return java_objekt_stack.pop();}

// -->
</SCRIPT>
</HEAD>

<BODY>
    <FORM>
        <INPUT TYPE="text"
                NAME="wert">
        <INPUT TYPE="button"
                VALUE="In Stack ablegen"
                onClick=" in_stack_ablegen (this.form.wert.value)">
        <INPUT TYPE="button"
                VALUE="aus Stack holen"
                onClick="alert(aus_stack_holen ())">
    </FORM>
</BODY>
</HTML>

```

22. Text

22.1. Text-Kapitälchen

Kapitälchen bilden: Text derat umwandeln, dass

Kleinbuchstaben nach Großbuchstaben gewandelt UND
in der Höhe etwas kleiner darstellen, als die der im Text vor der
Umwandlung vorhandenen Großbuchstaben

Großbuchstabe
in der Höhe etwas größer darstellen, als die der im Text vor der Umwandlung vorhandenen
Großbuchstaben

```

<SCRIPT ...>
<!--
    function erzeuge_kapitaelchen(text)
    {
        var umgewandelter_text="";
        var zeichen="";
        for ( var zahler=0; zahler<=text.length-1; zahler++)
        {
            zeichen=text.charAt(zahler); // Zeichen aus Text holen an akt. Position+

            if (zeichen==zeichen.toLowerCase())
            {umgewandelter_text += zeichen.toUpperCase().fontsize("-1")
              // zeichen war Kleinbuchstabe, also
              // nach Großbuchstabe UND Fonthöhe etwas kleiner
            }
            else
            {umgewandelter_text += zeichen.toUpperCase().fontsize("+1")
              // Fonthöhe etwas größer
            }
        }
        return umgewandelter_text;
    }

    var text=new("Hallo Du !");
    text.erzeuge_kapitaelchen(); // Funktion als neue Eigenschaft von Text per Prototyping

    oder writeln(erzeuge_kapitaelchen("Hallo Du !")); // ohne Prototyping

// -->
</SCRIPT>

```

22.2. Textmeldung in Statuszeile, wenn Maus über einen Link (Verweis) fährt

Variante 1

```

.....
<A HREF="link_oder_html_datei"
    onMouseOver="window.status='text_in_statuszeile'; return true;"
    onMouseOut="window.status="";

```



```
>
.... text ...
</A>
```

Variante 2

```
.....
<SCRIPT .....>
  <!--
      function meldung_in_statuszeile_ausgeben(melde_text)
      {
          window.status=melde_text;
          setTimeout("meldung_in_status_zeile_loeschen()",2000);
      }

      function meldung_in_statuszeile_loeschen()
      {
          window.status="";
      }

  //-->
</SCRIPT>
.....
<A HREF="link_oder_html_datei"
    onMouseOver="meldung_in_statuszeile_ausgeben('Das ist ein Link');
    return true;"
>
.... text ...
</A>
```

22.3. Textticker

```
<SCRIPT ..... >
<!--
    var argument_laenge=0;
    var position_im_tickertext=0;
    var zahler=0;

    function ticken()
    {
        argument_laenge=ticker.arguments.length;
        for (i=0; i<argument_laenge; i++)      this[i]=ticker.arguments[i];
    }

    text=new ticker("text_1", ..... , "text_n");

    var text_laenge=text[position_im_tickertext].length;

    function textticker()
    { dokument.formular_name.input_name.value=
        text[position_im_tickertext].substring(0,argument_laenge) + " _ ";
      if (position_im_tickertext++ ==1)
      { position_im_tickertext=0;
        setTimeout("textticker()",1000);
        zahler++;
        if (zahler==argument_laenge) zahler=0;
        laenge=text_laenge[zahler].lengt;
      }
      else setTimeout("textticker()",50);
    }
  //-->
</SCRIPT>
.....
<BODY onLoad="textticker()">
  <FORM NAME="formular_name">
    <INPUT NAME="input_name"
          TYPE=button
          VALUE="Tickerfeld">
  </FORM>
</BODY>
```

alternativ wäre auch möglich:

1. var gesamt_kette="ticker_text";
2. Funktion textticker definieren mit
 - 2.1. aus text das linkeste Zeichen per substring() isolieren
 - 2.2. die Teilkette neben dem linkesten Zeichen per substring() isolieren



- 2.3. linkeste Zeichen rechts an die Teilkette anfügen zu Gesamtkette
- 2.4. `gesamt_kette` ausgeben z.B.
 - `self.status=gesamt_kette;` im eigenen Fenster
 - `top.status=gesamt_kette;` im obersten Fenster
3. `setInterval("textticker",verzögerung);`

23. Spiele

23.1. Break Out

Break Out benötigt intensiv CPU-Ressourcen.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JScript">
<!--
// Break Out-Spiel für den Internet Explorer

// ##### änderbare Variablen #####

// +++++ Ball
var Ball_Bild_Name="ball.gif";
    // Achtung: Ball im Bild muss direkt an der Bildgrenze anstossen
    //      also keinen Leerraum um Ball herum !!
    // Grund: Die Ecken des Bildes sind die Berührungspunkte.
    //      Je grösser der Leerraum der Ecke (Ball ist rund, aber Bild
    //      ist eckig), umso ungenauer sieht der Spieler die
    //      Berührung.
    // Ideal wäre daher ein 4-eckiger Ball, der Bild komplett ausfüllt.
    // Bild muss bei rundem Ball in den Farben der Ecken transparent sein,
    //      damit Ecken nicht sichtbar sind (nur GIF-Bild)
var Ball_Bild_Breite=10;
var Ball_Bild_Hoehe=10;

var BallNummer_Maximal=2;
var Ball_Geschwindigkeit_Max=10;      // Millisekunden pro Bewegungsschritt, >=10
                                        // nur 10, 20, 30 etc.
                                        // maximaler Speed=kleinster Wert
var Ball_Geschwindigkeit_Min=100;    // Millisekunden pro Bewegungsschritt, 1000=1 Sekunde
                                        // > Ball_Geschwindigkeit_Max
                                        // nur 10, 20, 30 etc.
                                        // minimaler Speed=grösster Wert

// +++++ Spielfläche
//      Ursprung (0,0) des Grafik-Koordinatensystemes liegt links oben
var Spielflaeche_AbstandVonFensterRand_Oben=16; // >=0
var Spielflaeche_AbstandVonFensterRand_Links=16; // >=0

var Spielflaeche_BlockZeilen_Anzahl=14; // > 4 max. 14
var Spielflaeche_BlockSpalten_Anzahl=20; // > 3

var Spielflaeche_HintergrundFarbe='black'; // oder #xxxxx als freier Hexawert
//      Hintergrundfarbe darf NICHT in BlockZeile_MoeglicheFarbenFeld vertreten sein
//      Farbe des Schlägers sein
//      Hintergrundfarbe ist zugleich Löschfarbe eines Blockes
//      Farbe der Blockzwischenräume

// +++++ Schläger
var Schlaeger_Farbe='#B0B0FF'; // Beachte Spielflaeche_HintergrundFarbe !!

// +++++ Block
var Block_Hoehe=10; // >=Ball_Bild_Hoehe
var Block_Breite=30; // >=Ball_Bild_Breite

// +++++ Blockfeld
var BlockSpalte_BlockAbstand=15; // > Block_Hoehe
// Hinweis: Es gibt immer nur schräge Ball-Bewegungen, also keine direkt
//      waagerechten und senkrechten Bewegungen.
//      Jedes Berühren eines Blockes bewirkt immer Schräge !
//      Trifft Ball auf eine Lücke zwischen 2 Blöcke, so ist die nächste Richtung
```



```
// leicht schräg, kann aber zum Hinweinswandern in die Lücke führen.
// Wenn die Lücke breit genug ist, wandert der Ball im Zickzack die
// Lücke entlang.

var BlockZeile_BlockAbstand=35; // > Block_Breite
// Hinweis: Es gibt immer nur schräge Ball-Bewegungen, also keine direkt
// waagerechten und senkrechten Bewegungen.
// Jedes Berühren eines Blockes bewirkt immer Schräge !
// Trifft Ball auf eine Lücke zwischen 2 Blöcke, so ist die nächste Richtung
// leicht schräg, kann aber zum Hinweinswandern in die Lücke führen.
// Wenn die Lücke breit genug ist, wandert der Ball im Zickzack die
// Lücke entlang.

var BlockSpalte_AbstandVonSpielfeldRand_Oben=10; // >=0
// Hinweis: Es gibt immer nur schräge Ball-Bewegungen, also keine direkt
// waagerechten und senkrechten Bewegungen.
// Jedes Berühren eines Blockes bewirkt immer Schräge !
// Je kleiner der Wert um so schmaler ist der Rand oberhalb der 1. Blockzeile.
// Je grösser der Rand oberhalb der 1. Blockzeile Zeile ist, um so mehr Blöcke können durch
// Ball-Zickzack im oberen Bereich gelöscht werden, was das Spiel
// natürlich vereinfacht. Es kann dann sein, dass der Ball ewig braucht,
// um in den Bereich des Schlägers zu gelangen --> Button "Nächster Ball" hilft ab.

var BlockSpalte_AbstandVonSpielfeldRand_LinksBzwRechts=10; // >=0
// Hinweis: Es gibt immer nur schräge Ball-Bewegungen, also keine direkt
// waagerechten und senkrechten Bewegungen.
// Jedes Berühren eines Blockes bewirkt immer Schräge !
// Je kleiner der Wert um so schmaler ist der Rand rechts neben der 1. Blockspalte
// Je grösser der Rand rechts neben der 1. Blockspalte ist, um so mehr Blöcke können durch
// Ball-Zickzack im seitlichen Bereich gelöscht werden, was das Spiel
// natürlich vereinfacht. Analog dazu auch im rechten Bereich neben der letzten Blockspalte.
// Es kann dann sein, dass der Ball ewig braucht,
// um in den Bereich des Schlägers zu gelangen --> Button "Nächster Ball" hilft ab.

// +++++ Sound
// Sound-Wiedergabe ist SEHR träge, so dass das Ping-Pong-Geräusch nicht erzeugt werden kann,
// da das Spiel sonst nicht flüssig läuft !!!
// ist in Echtzeit nicht möglich !!!
// Feld der Urls
// Sounds können auch mid sein
// genau 2 Feldelemente nötig !
var SoundFeld_Url=new Array
(
'0.wav', // Spiel verloren
'1.wav' // Spiel gewonnen
)

var SoundFeld_Laenge=0;
var SoundFeld_Objekt;
var SoundFeldIndex=0; // ab 0

// ##### interne Variablen #####
// alle Variablen möglichst Wert-Typ-gerecht initialisieren (bei Zeiger kein "nil" oder
// "null" verwenden, also Zeiger ohne init).

// +++++ Spielfläche
// Ursprung (0,0) des Grafik-Koordinatensystemes liegt links oben
var Div_SpielFlaeche_Breite=0;
var Div_SpielFlaeche_Hoehe=0;

// +++++ Block-Feld
// Ursprung (0,0) des Grafik-Koordinatensystemes liegt links oben

// Blockspalte: Alle Spalten sind gleich lang
// Obere Kante jeder Blockspalte bildet obere Grenze (Top) des Blockfeldes
// Untere Kante jeder Blockspalte bildet untere Grenze (Down) des Blockfeldes
var BlockSpalte_Top=0; // identisch mit Top des Blockfeldes
var BlockSpalte_Down=0; // identisch mit Down des Blockfeldes
var BlockSpalte_Zahler=0; // ab 0
```



```

var BlockAnzahl_Gesamt=0;
var BlockAnzahl_Geloescht=0;
var BlockZeiger;

// Blockzeile: Alle Zeilen sind gleich lang
// Linke Kante jeder Blockzeile bildet linke Grenze (Left) des Blockfeldes
// Rechte Kante jeder Blockzeile bildet rechte Grenze (Right) des Blockfeldes
var BlockZeile_Left=0; // identisch mit Left des Blockfeldes
var BlockZeile_Right=0; // identisch mit Right des Blockfeldes
var BlockZeile_Zahler=0; // ab 0
var BlockZeile_Nummer=0; // ab 0
var BlockZeilenNummerFeld;
var BlockZeilenNummerFeld_Index=0; // ab 0

// 14 mögliche Farben der Blockzeilen (siehe auch BlockZeilen_FarbFeld)
var BlockZeile_MoeglicheFarbenFeld=new Array // 14 ist Maximalwert von Spielflaeche_BlockZeilen_Anzahl
// nur vordefinierte Farbnamen des Internet Explorer ab 4.0
(
'aqua',
'lightgrey', // mit e und nicht a; dagegen "gray" mit a und nicht mit e
'lightblue',
'gold',
'yellow',
'olive',
'lime',
'green',
'fuchsia',
'purple',
'red',
'maroon',
'teal',
'navy'
)

// Farbfeld der Blockfarben je nach Zustand: Werte stammen aus BlockZeile_MoeglicheFarbenFeld
// Blöcke sollten nicht sofort gelöscht werden, sondern pro Berührung die Farbe ändern.
// Erst ab einer Anzahl von Berührungen sollte der Block gelöscht werden.
// Anzahl der Berührungen=Anzahl (ab 1) der Farben MINUS 1
// Jede Blockzeile hat ihre eigene Farbe.
var BlockZeilen_FarbFeld;

// +++++ Ball
// Ursprung (0,0) des Grafik-Koordinatensystemes liegt links oben

var Ball_Nummer=0;
var Ball_Start_Top=0;
var Ball_Start_Left=0;
var Ball_Geschwindigkeit_Vorzeichen=0;
var Ball_Geschwindigkeit_Button_Top=0;
var Ball_Geschwindigkeit_Button_Left=0;
var Ball_Geschwindigkeit_SchrittWeite=0;
var Ball_Geschwindigkeit_Aktuell=0;
var Ball_Geschwindigkeit_Aktuell_Anzeige=0;
var Ball_Geschwindigkeit_Text1='Speed down '; // Leerzeichen am Ende !
var Ball_Geschwindigkeit_Text2='Speed up '; // Leerzeichen am Ende !
var Ball_Aktuell_Left=0; // enthält Richuung, also < 0 oder > 0
var Ball_Aktuell_Top=0; // enthält Richuung, also < 0 oder > 0
var Ball_Neu_Left=0; // enthält Richuung, also < 0 oder > 0
var Ball_Neu_Top=0; // enthält Richuung, also < 0 oder > 0
var Ball_Max_Left=0;
var Ball_Max_Top_BisSchlaeger=0;
var Ball_Max_Top_BisHinterSchlaeger=0;
var Ball_Korrektur_Top=0;
var Ball_Korrektur_Top_Neu_Left=0;
var Ball_Korrektur_Top_Neu_Top=0;
var Ball_SchrittWeite_Horizontal=0;
var Ball_SchrittWeite_Horizontal_Halbiert=0;
var Ball_SchrittWeite_Horizontal_Neu=0; // < 0 so nach links, > 0 so nach rechts
var Ball_SchrittWeite_Vertikal=0;

```



```

var Ball_SchrittWeite_Vertikal_Halbiert=0;
var Ball_SchrittWeite_Vertikal_Neu=0;    // < 0 so nach oben, > 0 so nach unten
var Ball_Schritt_Horizontal_StartRichtung=0; // -1 nach links; 1 nach rechts
var Ball_Schritt_Vertikal_StartRichtung=0; // -1 nach oben; 1 nach unten
var Ball_Schritt_Vertikal_Richtung=false; // false nach oben, true nach unten
var Ball_Abstand_BallLeft_Zu_SchlaegerLeft=0;
var Ball_Abstand_BallLeft_Zu_BlockZeileLeft=0;
var Ball_Abstand_BallTop_Zu_BlockSpalteTop=0;
var Ball_AnzahlBlockZeilen=0;
var Ball_AnzahlNichtGetroffenerBloেকে_In_GetroffenerZeile=0;
var Ball_RichtungsFaktor_Top1=0;
var Ball_RichtungsFaktor_Top2=0;
var Ball_RichtungsFaktor_Top3=0;
var Ball_RichtungsFaktor_Left1=0;
var Ball_RichtungsFaktor_Left2=0;
var Ball_Geschwindigkeit_Aendern_Kette="";

// +++++ Anzeige
var Div_Anzeige_Font="font-size:24px;color='#44CC44'";
var AnzeigeText_AllClear="New Game !";
var AnzeigeText_GameOver="Game over!";
var AnzeigeText_GamePause="Pause  !";
var Div_Anzeige_innerHTML1='<FONT STYLE="'+Div_Anzeige_Font+" ">';
var Div_Anzeige_innerHTML2='</FONT>';
var Div_Anzeige_Top=0;
var Div_Anzeige_Left=0;
var Div_Anzeige_Erlaubt=false;
var Div_Anzeige_Angezeigt=false; // true, so Anzeige aktiv
var Div_GamePause_Angezeigt=false; // true, so AnzeigeText_GamePause bereits sichtbar

// +++++ Info
var Div_Info_Font="font-size:24px;color='#000044'";
var Div_Info_LeftPos=0;

// +++++ Startbutton
var Div_StartButton_Top=0;
var Div_StartButton_Left=0;
var Div_StartButton_Label="Start";

// +++++ Schläger
var Schlaeger_MaxLeft=0;
var Schlaeger_AktuellLeft=0;
var Schlaeger_StartLeft=0;
var Schlaeger_Breite_Gedrittelt=0;
var Schlaeger_Breite=0;
var Schlaeger_Hoehe=0;
var Schlaeger_AbstandVomFensterRand_Oben=0;

// +++++ Maus
var MousePositionHorizontal=0;

// +++++ Spielsteuerung
var AktivesSpiel_Pausiert=false; // false, so nicht pausiert
    // true, so pausiert
    // manipuliert von Div_SpielFlaeche_OnMouseOut_Handler
    //      Div_SpielFlaeche_OnMouseOver_Handler
    // Maus befindet sich bei Spielstart über der Spielfläche, da
    // das Start-Button auf der Spielfläche liegt. Damit
    // wird onmouseover erzeugt also Negation des Flags auf
    // false, also auf nicht gestoppt.

var SpielAktiv=false; // true, so aktiv

// +++++ Timer
var BallBewegen_TimerID;

// +++++ Sonstiges
var Wert1=0;
var Wert2=0;

```



```

var Kette="";
var Flag1=false;
var Flag2=false;
var Flag3=false;
var Flag4=false;
var StyleZeiger;

// ##### Eventhandler #####

function Div_SpielFlaeche_OnMouseMove_Handler()
{
// prüfen ob Spiel aktiv ist
if (SpielAktiv)
{
// Horizontale Position der Maus ermitteln
MousePositionHorizontal=event.x;

// neue Schlägerposition horizontal ermitteln
Schlaeger_AktuellLeft=MousePositionHorizontal;

// Damit Maus inner die Mitte des Schlägers zeigt, muss der Schläger verschoben werden.
// Das ist besonders wichtig, damit die Maus nicht aus Versehen den Spielbereich
// verlässt und so das Spiel stoppt, z.B. ke näher man den Grenzen des Spielfeldes kommt.
Schlaeger_AktuellLeft=Math.floor(Schlaeger_Breite / 2);

// prüfen ob Schläger den Spielbereich verlässt
// wenn verlässt, so Grenzen einstellen und kein Verschieben
if (Schlaeger_AktuellLeft < SpielFlaeche_AbstandVonFensterRand_Links)
{Schlaeger_AktuellLeft=SpielFlaeche_AbstandVonFensterRand_Links;}

if (Schlaeger_AktuellLeft > Schlaeger_MaxLeft)
{Schlaeger_AktuellLeft=Schlaeger_MaxLeft;}
}
}

function Div_SpielFlaeche_OnMouseOut_Handler()
{AktivesSpiel_Pausiert=true;}
// Flagwert direkt setzen und nicht negieren.
// Grund: Falls Event vom Browser "verschluckt"
// also in Echtzeit nicht erkannt wird,
// könnte die Negation fehlerhaft werden !

function Div_SpielFlaeche_OnMouseOver_Handler()
{AktivesSpiel_Pausiert=false;}
// Flagwert direkt setzen und nicht negieren.
// Grund: Falls Event vom Browser "verschluckt"
// also in Echtzeit nicht erkannt wird,
// könnte die Negation fehlerhaft werden !

function Div_Schlaeger_OnMouseOver_Handler()
{AktivesSpiel_Pausiert=false;}
// Flagwert direkt setzen und nicht negieren.
// Grund: Falls Event vom Browser "verschluckt"
// also in Echtzeit nicht erkannt wird,
// könnte die Negation fehlerhaft werden !

function Div_Ball_OnMouseOver_Handler()
{AktivesSpiel_Pausiert=false;}
// Flagwert direkt setzen und nicht negieren.
// Grund: Falls Event vom Browser "verschluckt"
// also in Echtzeit nicht erkannt wird,
// könnte die Negation fehlerhaft werden !

// ##### Ball-Routinen #####

function Ball_NummerAnzeigen()
{ID_Div_BallNummer_Anzeige.innerText='Ball: ' + Ball_Nummer + ' | ' + BallNummer_Maximal;}

```



```

function Ball_GeloeschteBlocksAnzahlAnzeigen()
{ID_Div_ClearedBlocks_Anzeige.innerText= 'Block: '
    + BlockAnzahl_Geloescht
    + ' | '
    + (BlockAnzahl_Gesamt - BlockAnzahl_Geloescht)
    + ' | ' + BlockAnzahl_Gesamt;
}

function Ball_Geschwindigkeit_Aendern()
{
    // maximaler Speed=kleinster Wert
    // minimaler Speed=größter Wert

    // prüfen auf Änderung der Richtung
    if (Ball_Geschwindigkeit_Aktuell >= Ball_Geschwindigkeit_Min) // Speed am kleinsten , Wert am höchsten
    {
        Ball_Geschwindigkeit_Aktuell=Ball_Geschwindigkeit_Min;
        Ball_Geschwindigkeit_Aktuell_Anzeige=Ball_Geschwindigkeit_Max;
        Ball_Geschwindigkeit_Vorzeichen=-1; // Wert absenken, also Speed erhöhen
        Ball_Geschwindigkeit_Aendern_Kette=Ball_Geschwindigkeit_Text2;
    }

    if (Ball_Geschwindigkeit_Aktuell <= Ball_Geschwindigkeit_Max) // Speed am höchsten, Wert am kleinsten
    {
        Ball_Geschwindigkeit_Aktuell=Ball_Geschwindigkeit_Max;
        Ball_Geschwindigkeit_Aktuell_Anzeige=Ball_Geschwindigkeit_Min;
        Ball_Geschwindigkeit_Vorzeichen=1; // Wert erhöhen, also Speed senken
        Ball_Geschwindigkeit_Aendern_Kette=Ball_Geschwindigkeit_Text1;
    }

    // Geschwindigkeit in aktueller Richtung ändern
    Ball_Geschwindigkeit_Aktuell+=(Ball_Geschwindigkeit_SchrittWeite * Ball_Geschwindigkeit_Vorzeichen);
    Ball_Geschwindigkeit_Aktuell_Anzeige+=(Ball_Geschwindigkeit_SchrittWeite * Ball_Geschwindigkeit_Vorzeichen);

    // Button-Text ändern
    Kette=Ball_Geschwindigkeit_Aendern_Kette + Ball_Geschwindigkeit_Aktuell_Anzeige.toString();
    ID_Input_BallGeschwindigkeit.value=Kette;
}

function Ball_EinhaltungGrenzenSpielBereich_Pruefen()
// prüft auch auf Spielende
{
    // +++++ prüfen ob Ball den Spielbereich verlässt: Horizontal

    // ---- linke Grenze
    if (Ball_Aktuell_Left < SpielFlaeche_AbstandVonFensterRand_Links)
    {
        // auf Grenze setzen
        Ball_Aktuell_Left=SpielFlaeche_AbstandVonFensterRand_Links;

        // und Richtung ändern
        Ball_SchrittWeite_Horizontal_Neu=-Ball_SchrittWeite_Horizontal_Neu;
    }

    // ---- rechte Grenze
    if (Ball_Aktuell_Left > Ball_Max_Left)
    {
        // auf Grenze setzen
        Ball_Aktuell_Left=Ball_Max_Left;

        // und Richtung ändern
        Ball_SchrittWeite_Horizontal_Neu=-Ball_SchrittWeite_Horizontal_Neu;
    }

    // +++++ prüfen ob Ball den Spielbereich verlässt: Vertikal

    // ---- obere Grenze
    if (Ball_Aktuell_Top < SpielFlaeche_AbstandVonFensterRand_Oben)

```



```

{
// auf Grenze setzrn
Ball_Aktuell_Top=SpielFlaeche_AbstandVonFensterRand_Oben;

// und Richtung ändern
Ball_SchrittWeite_Vertikal_Neu=-Ball_SchrittWeite_Vertikal_Neu;
}

// --- untere Grenze: bezüglich Schläger prüfen, also Schläger erreicht oder
// überschritten wird
if (Ball_Aktuell_Top >= Ball_Max_Top_BisSchlaeger) // erreichen oder überschreiten
{
// prüfen ob vertikale Richtung nach oben ist
if (!Ball_Schritt_Vertikal_Richtung)
{
// nach oben

// nächste Position horizontal ermitteln zum Zweck der Reaktion
// auf alle Arten des Erreichens des Schläger
// als Bremsen oder Beschleunigen des Balls
// als Richtungsänderung des Balls
Wert1= Schlaeger_AbstandVomFensterRand_Oben
- Ball_SchrittWeite_Vertikal
- Ball_Aktuell_Top;

// Hinweis: Schrittweiten enthalten Richtung
Ball_Neu_Left=Ball_SchrittWeite_Horizontal_Neu / Ball_SchrittWeite_Vertikal_Neu;
Ball_Neu_Left*=Wert1;
Ball_Neu_Left+=Ball_Aktuell_Left; // enthält Richtung, kann also auch < 0 sein
// < 0 so Bewegung nach links

// Umgebung um die nächste Position abklappern: alle Arten des Erreichens des Schlägers
// Bremsen oder Beschleunigen des Balls
// Richtungsänderung des Balls
// Ball_Neu_Left kann > 0 oder < 0 sein
// Schlaeger_AktuellLeft immer >= 0
// Umgebung: links und rechts vom Schläger
// ergibt sich auch der Ungenauigkeit der Schlägerbewegung
// wegen Trägheit des Schlägerbewegung

// Umgebung ermitteln
Wert1=Schlaeger_AktuellLeft - Schlaeger_Breite_Gedrittelt;
Wert2=Schlaeger_AktuellLeft + Schlaeger_Breite;

// auf Betreten der Umgebung prüfen
if ( (Ball_Neu_Left >= Wert1)
&& (Ball_Neu_Left <= Wert2)
)
{
// Ball liegt in der Umgebung

// +++++ Reaktion für vertikale Richtung

// Ball 1 Ballschritt VOR die Obertfläche des Schlägers setzen
// nächster Ballschritt würde also die Oberfläche des Schlägers berühren
// wenn Bildhöhe > Ballschritt, so taucht der Ball in dem Schläger ein,
// als ob der Ball festklebt
Ball_Aktuell_Top= Schlaeger_AbstandVomFensterRand_Oben
- Ball_SchrittWeite_Vertikal;

// Richtung ändern
Ball_SchrittWeite_Vertikal_Neu=-Ball_SchrittWeite_Vertikal_Neu;

// +++++ Reaktion für horizontale Richtung

// nächsten Schritt verwenden
Ball_Aktuell_Left=Ball_Neu_Left;

// Bremsen bzw. Beschleunigen ermitteln anhand des Abstand

```



```

//      Left von Ball zum Left von Schläger
//      ermitteln
Ball_Abstand_BallLeft_Zu_SchlaegerLeft=Ball_Neu_Left - Schlaeger_AktuellLeft;

// neue Schrittweite ohne Bremsen bzw. Beschleunigen
//      aber mit Richtung
//      ermitteln
Wert1=Math.abs(Ball_SchrittWeite_Horizontal_Neu); // Richtung entfernen wegen Division

Ball_SchrittWeite_Horizontal_Neu=Math.floor( Ball_SchrittWeite_Horizontal
                                           * Wert1 // keine Richtung
                                           / Ball_SchrittWeite_Horizontal_Neu // Richtung wieder drin
                                           );

// neue Schrittweite mit Bremsen bzw. Beschleunigen ermitteln
//      Grad für Bremsen und Beschleunigen ist der Abstand von Ball und Schläger
//      Wie in der Natur sind Beschleunigungen nicht linear:
//      wenn der Ball auf spitze Kante trifft, so ist Beschleunigung meisst höher
//      als auf Fläche (Impulsübertragung als Punkt bei Kante).
//      Die neuen Werte für Ball_SchrittWeite_Horizontal_Neu
//      sind empirisch ermittelt, da Funktionsberechnungen zu aufwendig sind
//      (auch in der Laufzeit des Spieles).

if (Ball_Abstand_BallLeft_Zu_SchlaegerLeft < 0)
{
    if (Ball_Abstand_BallLeft_Zu_SchlaegerLeft < -4)
    {Ball_SchrittWeite_Horizontal_Neu=-15;}

    if ( (Ball_Abstand_BallLeft_Zu_SchlaegerLeft >= -4)
        && (Ball_Abstand_BallLeft_Zu_SchlaegerLeft <= -1)
        )
    {Ball_SchrittWeite_Horizontal_Neu=-11;}
}
else
{
    // >=0

    if (Ball_Abstand_BallLeft_Zu_SchlaegerLeft <= 4)
    {Ball_SchrittWeite_Horizontal_Neu=-4;}

    if ( (Ball_Abstand_BallLeft_Zu_SchlaegerLeft >= 14)
        && (Ball_Abstand_BallLeft_Zu_SchlaegerLeft <= 16)
        )
    {Ball_SchrittWeite_Horizontal_Neu=-2;}

    if ( (Ball_Abstand_BallLeft_Zu_SchlaegerLeft >= 17)
        && (Ball_Abstand_BallLeft_Zu_SchlaegerLeft <= 20)
        )
    {Ball_SchrittWeite_Horizontal_Neu=2;}

    if ( (Ball_Abstand_BallLeft_Zu_SchlaegerLeft >= 28)
        && (Ball_Abstand_BallLeft_Zu_SchlaegerLeft <= 32)
        )
    {Ball_SchrittWeite_Horizontal_Neu=4;}

    if ( (Ball_Abstand_BallLeft_Zu_SchlaegerLeft >= 33)
        && (Ball_Abstand_BallLeft_Zu_SchlaegerLeft <= 36)
        )
    {Ball_SchrittWeite_Horizontal_Neu=11;}

    if (Ball_Abstand_BallLeft_Zu_SchlaegerLeft > 36)
    {Ball_SchrittWeite_Horizontal_Neu=15;}
}

// Richtung der vertikale Schrittweite merken
Ball_Schritt_Vertikal_Richtung=(Ball_SchrittWeite_Vertikal_Neu > 0);
// false nach oben, true nach unten
}

```




```

else
{
    // Ball liegt nicht in der Umgebung

    // prüfen vertikal bezüglich Spielbereichgrenze: unten

    if (Ball_Aktuell_Top > Ball_Max_Top_BisHinterSchlaeger)
    {
        // es müsste der nächste Schritt nach oben sein, also merken
        Ball_Schritt_Vertikal_Richtung=false;

        // Ballanzahl dezimieren
        Ball_Nummer-=1;

        // auf Spielende prüfen
        Spiel_Ende();
    }
}
}

function Ball_BlockBeruehrung_Pruefen_HilfsRoutine(Zahl)
{
    var Funktionswert=Ball_RichtungsFaktor_Left2 - Ball_Neu_Left + Zahl;
    Funktionswert*= Ball_RichtungsFaktor_Top1;
    Funktionswert+=Ball_Neu_Top;
    Funktionswert=Math.floor(Funktionswert);

    return Funktionswert;
}

function Ball_BlockBeruehrung_Pruefen_Korrektur_Vertikal(Zahl)
{
    Ball_Korrektur_Top_Neu_Left=Ball_RichtungsFaktor_Top3 - Ball_Neu_Top + Zahl;
    Ball_Korrektur_Top_Neu_Left*=Ball_RichtungsFaktor_Left1;
    Ball_Korrektur_Top_Neu_Left+=Ball_Neu_Left;
    Ball_Korrektur_Top_Neu_Left=Math.floor(Ball_Korrektur_Top_Neu_Left);

    Ball_Korrektur_Top_Neu_Top=Ball_RichtungsFaktor_Top3 + Zahl;

    Ball_Aktuell_Left=Ball_Korrektur_Top_Neu_Left - Ball_SchrittWeite_Horizontal_Halbiert;
    Ball_Aktuell_Top=Ball_Korrektur_Top_Neu_Top - Ball_SchrittWeite_Vertikal_Halbiert;

    // Richtung ändern
    Ball_SchrittWeite_Vertikal_Neu*=-1;
}

function Ball_BlockBeruehrung_Pruefen_Korrektur_Horizontal(Zahl)
{
    Ball_Korrektur_Top_Neu_Left=Ball_RichtungsFaktor_Left2 + Zahl;
    Ball_Korrektur_Top_Neu_Top =Ball_BlockBeruehrung_Pruefen_HilfsRoutine(Zahl);

    Ball_Aktuell_Left=Ball_Korrektur_Top_Neu_Left - Ball_SchrittWeite_Horizontal_Halbiert;
    Ball_Aktuell_Top=Ball_Korrektur_Top_Neu_Top - Ball_SchrittWeite_Vertikal_Halbiert;

    // Richtung ändern
    Ball_SchrittWeite_Horizontal_Neu*=-1;
}

function Ball_BlockBeruehrung_Pruefen_Korrektur_VertikalUndHorizontal(Zahl)
{
    // Korrektur von Top per Richtungsfaktor
    Ball_Korrektur_Top=Ball_BlockBeruehrung_Pruefen_HilfsRoutine(Zahl);

    if (Ball_Korrektur_Top > (Ball_RichtungsFaktor_Top3 + 18))
    {Ball_BlockBeruehrung_Pruefen_Korrektur_Vertikal(18);}
    else
    {
        if (Ball_Korrektur_Top < (Ball_RichtungsFaktor_Top2 + 44))

```



```

{Ball_BlockBeruehrung_Pruefen_Korrektur_Vertikal(0);}
else
{Ball_BlockBeruehrung_Pruefen_Korrektur_Horizontal(Zahl);}
}
}

function Ball_BlockBeruehrung_Pruefen()
// prüft auch auf Spielende
{
// +++++ neuen Ballschritt einstellen
Ball_Neu_Top =Ball_Aktuell_Top + Ball_SchrittWeite_Vertikal_Halbiert;
Ball_Neu_Left=Ball_Aktuell_Left + Ball_SchrittWeite_Horizontal_Halbiert;

// +++++ prüfen ob Ball mit diesem Schritt in den Bereich des Blockfeldes gelangen wird
// Ursprung (0,0) des Grafik-Koordinatensystemes liegt links oben
Flag1=(Ball_Neu_Top >= BlockSpalte_Top); // Ball ab Oberkante der Blockspalte
Flag2=(Ball_Neu_Top <= BlockSpalte_Down); // Ball bis Unterkante der Blockspalte
Flag3=(Ball_Neu_Left >= BlockZeile_Left); // Ball ab linker Kante der Blockzeilefalse
Flag4=(Ball_Neu_Left <= BlockZeile_Right); // Ball bis rechte Kante der Blockzeile

if (Flag1 && Flag2 && Flag3 && Flag4)
{
// Ball hat das Blockfeld betreten
// Ball muss daher 1 Block berührt haben, da der Ball im Blockfeld liegt

// +++++ vom Ball getroffenen Bereich ermitteln: Es werden jeweils einzeln erkannt
// Berühren eines ungelöschten Blockes
// Betreten des Bereiches eines gelöschten Blockes
// Betreten des Zwischenraumes zwischen Blöcken (auch zwischen gelöschten Blöcken)

// Erkannte Bereiche werden in Abhängigkeit der Ballrichtung behandelt, also
// verschieden VOR bzw. nach der Ball-Reflexion durch Blockberührung

// Der Algorithmus ist so angelegt, dass nicht exakt jede Berührung erkannt wird, sondern
// zufälliges Ignorieren einer Berührung erfolgen kann, was das Spiel erschwert und
// verlängert.

// Erkannt wird nicht die Reflexion ausserhalb des Blockfeldes, also z.B. an der Spielbereichsgrenze

// Achtung: Aufgrund der Eigenschaften des Ballbildes z.B. runder Ball im eckigen Bild mit
// transparenten Ecken, kann es passieren, dass die Berührung des Blockes mit einer Ballecke
// ungenau aussieht, da der transparente Bereich der Ecke des Bilds die Berührungskante ist.
// Je dicker die Ecken bzw. je mehr transparenter Bereich um den Ball ist, um so ungenauer
// die sichtbare Berührung !

// Berührung des Balles bzw. Betreten:
// Immer linke obere Ecke des Bildes ist der Bezugspunkt,
// da Left und Top-Angaben des Bildes immer relativ zur linken oberen Ecke sind.

// Wichtig: Math.floor() liefert immer die nächst kleiner ganze Zahl aus dem Ergebnis
// des Ausdruckes innerhalb der Klammern
// rundet also ganzzahlig ab
// Bsp.: Math.floor(1/2) identisch mit Math.floor(0.5) also 0 geliefert

// ----- Blockspalte: Blockzeile VOR der getroffenen Blockzeile ermitteln
// ist bezüglich Beginn (Top) der Blockspalte ermittelbar

// Abstand aktuelles Top zum Top oberster Block in Blockspalte
// also Position des Balles bezüglich Top des Blockfeldes
// Position immer >= 0 da Ball im Blockfeld liegt
// gibt Lage des Balles bezüglich der getroffenen BLOCKZEILE an
Ball_Abstand_BallTop_Zu_BlockSpalteTop=Ball_Neu_Top - BlockSpalte_Top

// Anzahl Zeilen ermitteln je nach Ballbewegung

// Bewegung nach oben und VOR der Richtungsänderung (Reflexion):
// Anzahl ganzer Zeilen, die oberhalb der getroffenen Zeile liegen
// Wird Zwischenraum erkannt, dann Anzahl der Zeilen inklusive der
// Zeile, oberhalb derer der Zwischenraum liegt

```



```

// Bewegung nach unten und NACH der Richtungsänderung (Reflexion):
//   Anzahl Zeilen inklusive der getroffenen Zeile
//   Wird Zwischenraum erkannt, dann Anzahl der Zeilen inklusive der
//   Zeile, die unterhalb des Zwischenraumes liegt
//   Wird gelöschter Block erkannt, dann Anzahl der Zeilen inklusive der
//   Zeile, die den gelöschten Block enthält

// Hinweis:
//   Es gibt immer nur schräge Ball-Bewegungen, also keine direkt
//   waagerechten und senkrechten Bewegungen.
//   Jedes Berühren eines Blockes bewirkt immer Schräge !
//   Bsp.: Trifft Ball auf eine senkrechte Lücke zwischen 2 Blöcke, so
//   ist die nächste Richtung leicht schräg, kann aber
//   zum Hinweindwandern in die Lücke führen. Wenn die Lücke breit
//   genug ist, wandert der Ball im Zickzack nach oben.
//   Analog dazu im Bereich zwischen Spielfeld und Blockfeld, also
//   im Rahmen um das Blockfeld. Je grösser der Rahmen oberhalb
//   der obersten Zeile ist, um so mehr Blöcke können durch
//   Ball-Zickzack im oberen Bereich gelöscht werden, was das Spiel
//   natürlich vereinfacht.
Ball_AnzahlBlockZeilen=Math.floor( Ball_Abstand_BallTop_Zu_BlockSpalteTop
                                   / BlockSpalte_BlockAbstand
                                   );

// ---- Blockzeile: getroffene Blockspalte ermitteln
//   Spalte ist bezüglich Beginn (Left) der Blockzeile ermittelbar

//   Abstand aktuelles Left des Balles zum Left linkerster Block in Blockzeile
//   also Position des Balles bezüglich Left des Blockfeldes
//   Position immer >= 0 da Ball im Blockfeld liegt
//   gibt Lage des Balles bezüglich der getroffenen BLOCKSPALTE an
Ball_Abstand_BallLeft_Zu_BlockZeileLeft=Ball_Neu_Left - BlockZeile_Left;

//   Anzahl der nicht getroffenen Blöcke in der getroffenen Zeile
//   0 wenn getroffener Block der erste (linkste) in der Blockzeile ist
//   nicht getroffen bedeutet: Zwischenraum oder bereits gelöschter Block
Ball_AnzahlNichtGetroffenerBloecke_In_GetroffenerZeile=Math.floor( Ball_Abstand_BallLeft_Zu_BlockZeileLeft
                                                                    / BlockZeile_BlockAbstand
                                                                    );

// ---- Index (ab 0) des Blockes, der erkannt wurde (gelöscht oder ungelöscht)
BlockZeilenNummerFeld_Index= (Ball_AnzahlBlockZeilen * SpielFlaeche_BlockSpalten_Anzahl)
                             + Ball_AnzahlNichtGetroffenerBloecke_In_GetroffenerZeile;

// +++++ Zeilennummer (ab 0) des Blockes aus Zeilennummer-Feld holen
BlockZeile_Nummer=BlockZeilenNummerFeld[BlockZeilenNummerFeld_Index];

//   prüfen ob Zeilennummer gültig ist
//   ungültige Zeilennummer IMMER wenn erkannt wurde:
//   bereits gelöschter Block
//   Zwischenraum der Blöcke (auch zwischen gelöschten Blöcken)
//   also Ball im leeren Blockfeldbereich
if (BlockZeile_Nummer < SpielFlaeche_BlockZeilen_Anzahl)
{
// Zeilennummer gültig. Es wurde kein Zwischenraum und kein gelöschter Block erkannt

// +++++ Farbe des Blockes ändern: Es muss die nächste Farbe im BlockZeilen_FarbFeld verwendet werden.
//   also Blockzeilen-Nummer um 1 erhöhen, da Blockzeilen-Nummer auch der Index auf das Farbfeld ist
BlockZeile_Nummer++;

//   merken, dass Farbe geändert wurde, also diese Blockzeilen-Nummer merken, also diesen
//   Index der geänderten Farbe
BlockZeilenNummerFeld[BlockZeilenNummerFeld_Index]=BlockZeile_Nummer;

//   Farbe ändern
eval('BlockZeiger=ID_Block' + BlockZeilenNummerFeld_Index + ');
BlockZeiger.bgColor=BlockZeilen_FarbFeld[BlockZeile_Nummer];

```



```

// +++++ prüfen ob Farbe des gelöschten Blockes verwendet wurde: Wenn ja, so Anzahl gelöschter Blöcke erhöhen.
//   Farbe des gelöschten Blockes ist die LETZTE Farbe im Farbfeld
//   Elementanzahl im Farbfeld ist identisch mit der Blockzeilenanzahl der Spielfläche
if (BlockZeile_Nummer == SpielFlaeche_BlockZeilen_Anzahl)
{BlockAnzahl_Geloescht++;} // Löschfarbe wurde verwendet, also Anzahl gelöschter Blöcke erhöhen

// +++++ prüfen ob alle Blöcke gelöscht sind (alle Blöcke die Löschfarbe haben): Wenn ja, dann Spielende
if (BlockAnzahl_Geloescht >= BlockAnzahl_Gesamt)
{Spiel_Ende();}

// +++++ neue Ball-Bewegungsrichtung verändern je nach Situation der Berührung

// ---- Faktoren der Richtungskorrekturen ermitteln
//   Die Richtung wird mit Betreten des Zwischenraumes zweier Blöcke geändert
//   NACH dem Berühren des ungelöschten Blockes geändert
Ball_RichtungsFaktor_Top1=Math.floor(Ball_SchrittWeite_Vertikal_Neu / Ball_SchrittWeite_Horizontal_Neu);
Ball_RichtungsFaktor_Top2=BlockSpalte_BlockAbstand * Ball_AnzahlBlockZeilen;
Ball_RichtungsFaktor_Top3=Ball_RichtungsFaktor_Top2 + BlockSpalte_Top;

Ball_RichtungsFaktor_Left1=Math.floor(Ball_SchrittWeite_Horizontal_Neu / Ball_SchrittWeite_Vertikal_Neu);
Ball_RichtungsFaktor_Left2=BlockZeile_BlockAbstand * Ball_AnzahlNichtGetroffenerBloecke_In_GetroffenerZeile;
Ball_RichtungsFaktor_Left2+=BlockZeile_Left;

// ---- horizontal prüfen auf Bewegung nach rechts
if (Ball_SchrittWeite_Horizontal_Neu > 0)
{
    // Bewegung horizontal nach rechts korrigieren
    Ball_BlockBeruehrung_Pruefen_Korrektur_VertikalUndHorizontal(0);
}
else
{
    // Bewegung horizontal nach links korrigieren
    Ball_BlockBeruehrung_Pruefen_Korrektur_VertikalUndHorizontal(44);
}
}
}

function BallBewegen()
{
    // aktuellen Timer löschen
    clearTimeout(BallBewegen_TimerID);

    // prüfen ob, aktives Spiel nicht gestoppt ist
    if (AktivesSpiel_Pausiert) // false, so nicht pausiert
                             // true, so pausiert
                             // manipuliert vom Div_SpielFlaeche_OnMouseOutUndOver_Handler
    {
        // gestoppt, also Game-Pause anzeigen
        Spiel_Anzeige(AnzeigeText_GamePause);
    }
    else
    {
        // nicht gestoppt

        // Spiel-Anzeige abschalten (egal welche)
        Spiel_Anzeige_Aus();

        // Anzahl gelöschter Blöcke anzeigen
        Ball_GeloeschteBlocksAnzahlAnzeigen();

        // neue Ballposition ermitteln: inklusive Richtung in der Ball_SchrittWeite
        Ball_Aktuell_Left+=Ball_SchrittWeite_Horizontal_Neu;
        Ball_Aktuell_Top +=Ball_SchrittWeite_Vertikal_Neu;

        // und prüfen
        Ball_EinhaltungGrenzenSpielBereich_Pruefen(); // prüft auch auf Spielende
        Ball_BlockBeruehrung_Pruefen(); // prüft auch auf Spielende
    }
}

```



```

// Ball neu positionieren
ID_Div_Ball.style.posTop=Ball_Aktuell_Top;
ID_Div_Ball.style.posLeft=Ball_Aktuell_Left;

// Schlaeger neu positionieren
ID_Div_Schlaeger.style.posLeft=Schlaeger_AktuellLeft;
}

// Rekursion bei aktivem Spiel, aber auch dann,wenn Spiel gestoppt ist, damit späterer Start möglich wird
// ohne Neustart des Spieles
if (SpielAktiv)
{BallBewegen_TimerID=setTimeout("BallBewegen()", Ball_Geschwindigkeit_Aktuell);}
}

// ##### Spiel-Routinen #####

function Spiel_Anzeige_Aus()
{
// prüfen ob irgendeine Anzeige aktiv ist
if (Div_Anzeige_Angezeigt)
{
// aktiv also abschalten
ID_Div_Anzeige.style.visibility='hidden';

Div_Anzeige_Angezeigt=false;
Div_GamePause_Angezeigt=false; // Game-Pause-Meldung auch abgeschaltet
}
}

function Spiel_Anzeige(Text)
{
// Annahme: nichts anzeigen
var Anzeigen=false;

// prüfen ob Anzeige erlaubt ist
if (Div_Anzeige_Erlaubt)
{
// Egal ob die Anzeige bereits aktiv ist oder nicht: Sie kann immer aktiviert werden.
// Ausnahme: Game-Pause-Anzeige
// Annahme: Text anzeigen
Anzeigen=true;

// prüfen ob Game-Pause zu anzuzeigen ist
if (Text == AnzeigeText_GamePause)
{
// ist anzuzeigen

// prüfen ob Game-Pause NICHT bereits angezeigt wird
Anzeigen=!Div_GamePause_Angezeigt; // true geliefert, so Game-Pause noch nicht sichtbar
}
else
{
// nicht Game-Pause anzeigen, also Flag löschen
Div_GamePause_Angezeigt=false;
}

if (Anzeigen)
{
ID_Div_Anzeige.innerHTML= Div_Anzeige_innerHTML1
+ Text
+ Div_Anzeige_innerHTML2;
ID_Div_Anzeige.style.visibility='visible';
Div_Anzeige_Angezeigt=true;
}
}
}

function Spiel_Start_NaechsterBall()

```



```

{
// Ballnummer einstellen und prüfen
Ball_Nummer--;

if (Ball_Nummer > 0)
{Spiel_Start();}
else
{Spiel_End();}
}

function Spiel_Start()
{
// prüfen ob Spielende durch Break Out aller Blöcke erreicht wurde
// wenn ja, so neues Spiel
// wenn nein, dann nur neuer User-Anlauf im Spiel
if (BlockAnzahl_Geloescht >= BlockAnzahl_Gesamt)
{
// alle Blöcke wurden durch den User herausgebrochen, also init auf neues Spiel

BlockAnzahl_Geloescht=0;

Ball_Nummer=BallNummer_Maximal;

// Reset-Button ausblenden
ID_Div_Reset.style.visibility="visible";

// Anzeigen aller Art wieder zulassen
Div_Anzeige_Erlaubt=true;

// BlockZeilenNummerFeld init und Farben init
// pro Blockzeile: die Zeilennummer als Blockstatus in BlockZeilenNummerFeld
BlockZeilenNummerFeld_Index=-1; // ab 0
for (BlockZeile_Zahler=0; BlockZeile_Zahler < SpielFlaeche_BlockZeilen_Anzahl; BlockZeile_Zahler++) // 5
{
for (BlockSpalte_Zahler=0; BlockSpalte_Zahler < SpielFlaeche_BlockSpalten_Anzahl; BlockSpalte_Zahler++) // 8
{
BlockZeilenNummerFeld_Index++; // fortlaufend

eval('BlockZeiger=ID_Block' + BlockZeilenNummerFeld_Index + ');');
BlockZeiger.bgColor=BlockZeilen_FarbFeld[BlockZeile_Zahler];

BlockZeilenNummerFeld[BlockZeilenNummerFeld_Index]=BlockZeile_Zahler;
}
}
}

// aktuelle Anzeige löschen
Spiel_Anzeige_Aus();

// Ballnummer anzeigen
Ball_NummerAnzeigen();

// Anzahl gelöschter Blöcke anzeigen
Ball_GeloeschteBlocksAnzahlAnzeigen();

// Start-Button unsichtbar machen
ID_Div_StartButton.style.visibility='hidden';

// Spiel als aktiv markieren
SpielAktiv=true;

// BallRichtung zufällig festlegen
Ball_Schritt_Horizontal_StartRichtung=-1; // nach links oder rechts
if (Math.random() < 0.5){Ball_Schritt_Horizontal_StartRichtung=1;}

// Ballschrittweite festlegen
Ball_Schritt_Vertikal_Richtung=false; // immer nach oben

Ball_SchrittWeite_Horizontal_Neu= Ball_SchrittWeite_Horizontal

```



```

        * Ball_Schritt_Horizontal_StartRichtung;

Ball_SchrittWeite_Vertikal_Neu= Ball_SchrittWeite_Vertikal
        * Ball_Schritt_Vertikal_StartRichtung;

// Ball positionieren
Ball_Aktuell_Left=Ball_Start_Left;
Ball_Aktuell_Top =Ball_Start_Top;

// Schläger positionieren
Schlaeger_AktuellLeft=Schlaeger_StartLeft;

// Ballbewegen starten
BallBewegen_TimerID=setTimeout("BallBewegen()", Ball_Geschwindigkeit_Aktuell);
}

function Spiel_Ende()
{
    // Annahme: All Clear wirt nicht angezeigt
    var NichtAngezeigt=true;

    // Spiel als inaktiv markieren
    SpielAktiv=false;

    // Ballnummer anzeigen
    Ball_NummerAnzeigen();

    // prüfen ob alle Blöcke herausgebrochen wurden
    if (BlockAnzahl_Geloescht >= BlockAnzahl_Gesamt)
    {
        // alle Blöcke gelöscht worden (herausgebrochen)

        SoundErklingen(1);
        // Index 0 Spiel verloren
        //    1 Spiel gewonnen

        Spiel_Anzeige(AnzeigeText_AllClear);
        NichtAngezeigt=false;

        // Reset-Button unsichtbar machen
        ID_Div_Reset.style.visibility='hidden';
    }

    // prüfen ob alle Bälle verbraucht wurden
    if (Ball_Nummer <= 0)
    {
        // alle Bälle verbraucht

        // Game over anzeigen aber nur, wenn nicht bereits All Clear angezeigt wird
        if (NichtAngezeigt)
        {Spiel_Anzeige(AnzeigeText_GameOver);}

        SoundErklingen(0);
        // Index 0 Spiel verloren
        //    1 Spiel gewonnen

        // alle Blöcke als gelöscht markieren
        // wichtig für Spiel_Start()
        BlockAnzahl_Geloescht=BlockAnzahl_Gesamt;

        // weitere Anzeige blockieren bis zum nächsten Start
        Div_Anzeige_Erlaubt=false;

        // Reset-Button unsichtbar machen
        ID_Div_Reset.style.visibility='hidden';
    }

    // Start-Button sichtbar machen
    ID_Div_StartButton.style.visibility='visible';
}

```



```

}

// ##### Sound-Routine #####

function SoundObjektErzeugen(Url) // Konstruktor zum Vorladen
    // wird per new verwendet
{this.SoundFileUrl=Url;}

function SoundErklingen(SoundObjektIndex) // mit Url-Zuweisung erfolgt erklingen
{
    ID_Sound.src=SoundFeld_Objekt[SoundObjektIndex].SoundFileUrl;
    // Index 0 Spiel verloren
    //    1 Spiel gewonnen
}

// ##### init #####

function init()
{
    var IDZaehler; // fortlaufend ab 0
    var BlockZeilen_FarbFeldIndex;
    var TopWert;
    var LeftWert;

    // +++++ Sounds vorladen und als Objekte erzeugen

    // Anzahl der Soundfiles
    SoundFeld_Laenge=SoundFeld_Url.length; // ab 1

    // Objektfeld für Soundfiles erzeugen
    // Index 0 Spiel verloren
    //    2 Spiel gewonnen
    SoundFeld_Objekt=new Array(SoundFeld_Laenge);
    for (SoundFeldIndex=0; SoundFeldIndex < SoundFeld_Laenge; SoundFeldIndex++)
    {SoundFeld_Objekt[SoundFeldIndex]=new SoundObjektErzeugen(SoundFeld_Url[SoundFeldIndex]);}

    // +++++ Blockangaben prüfen

    if (Block_Breite < Ball_Bild_Breite)
    {Block_Breite=Ball_Bild_Breite;}

    if (Block_Hoehe < Ball_Bild_Hoehe)
    {Block_Hoehe=Ball_Bild_Hoehe;}

    if (BlockZeile_BlockAbstand <= Block_Breite)
    {BlockZeile_BlockAbstand=Block_Breite+1;}

    if (BlockSpalte_BlockAbstand <= Block_Hoehe)
    {BlockSpalte_BlockAbstand=Block_Hoehe + 1;}

    // +++++ Spielfläche als Tabelle erzeugen
    // Als DIV: Da kein Inhalt vorhanden, so nichts angezeigt
    // Als TABLE: Es wird in der Zelle automatisch ein Inhalt erzeugt, also angezeigt

    if (SpielFlaeche_BlockZeilen_Anzahl < 5) {SpielFlaeche_BlockZeilen_Anzahl=5;}
    if (SpielFlaeche_BlockZeilen_Anzahl > 14) {SpielFlaeche_BlockZeilen_Anzahl=14;}
    if (SpielFlaeche_BlockSpalten_Anzahl < 3) {SpielFlaeche_BlockSpalten_Anzahl=4;}

    // ---- Spielfläche
    Wert1=(SpielFlaeche_BlockSpalten_Anzahl +1) * Block_Breite;
    Wert2=Math.floor(BlockZeile_BlockAbstand / 3);
    Div_SpielFlaeche_Breite=Wert1 + Wert2;

    Wert1=SpielFlaeche_BlockZeilen_Anzahl * Block_Hoehe;
    Wert2=(SpielFlaeche_BlockZeilen_Anzahl-1) * BlockZeile_BlockAbstand;
    Div_SpielFlaeche_Hoehe=Wert1 + Wert2;

    document.write( '<TABLE ID="ID_Tabelle_SpielFlaeche"'
        + ' WIDTH=' + Div_SpielFlaeche_Breite

```




```

        + 'HEIGHT=' + Div_SpielFlaeche_Hoehe
        + 'BGCOLOR=' + SpielFlaeche_HintergrundFarbe + ''
        + '>'
        + '<TD></TD>' // leer !
    + '</TABLE>'
    );

StyleZeiger=ID_Tabelle_SpielFlaeche.style;
StyleZeiger.position='absolute';
StyleZeiger.posTop =SpielFlaeche_AbstandVonFensterRand_Oben;
StyleZeiger.posLeft=SpielFlaeche_AbstandVonFensterRand_Links;

// Handler für Spielfläche: Alle Elemente AUF der Spielfläche überlagern diese.
//      Wird ein solches Element per Maus betreten, wird für die
//      Spielfläche das Event onmouseout erzeugt. Das
//      Betreten eines Elementes AUF der Spielfläche bedeutet
//      Verlassen der Spielfläche --> siehe Schläger !!!

// Handler für Bewegen des Schlägers über gesamten Spielhintergrund, da
// der Schläger so dünn ist. Würde der Handler für den Schläger
// implementiert werden, dann muss der User mit der Maus EXAKT auf dem Schläger
// bleiben, was zu schwierig ist !
ID_Tabelle_SpielFlaeche.onmousemove=Div_SpielFlaeche_OnMouseMove_Handler;

// Handler für gestartetes (aktives) Spiel: Spieltstop bzw. Spielweitemachen
//      manipulieren AktivesSpiel_Pausiert false, so aktives nicht pausiert
//      true, so aktives pausiert
ID_Tabelle_SpielFlaeche.onmouseover=Div_SpielFlaeche_OnMouseOver_Handler;
ID_Tabelle_SpielFlaeche.onmouseout=Div_SpielFlaeche_OnMouseOut_Handler;

// +++++ Blockanzahl gesamt init
BlockAnzahl_Gesamt=SpielFlaeche_BlockZeilen_Anzahl * SpielFlaeche_BlockSpalten_Anzahl;

// +++++ Blockfarben-Feld init

// Farbfeld der Blockfarben je nach Zustand:
//   Blöcke sollten nicht sofort gelöscht werden, sondern pro Berührung die Farbe ändern.
//   Erst ab einer Anzahl von Berührungen sollte der Block gelöscht werden.
//   Anzahl der Berührungen=Anzahl (ab 1) der Farben MINUS 1
//   Jede Blockzeile hat ihre eigene Farbe

// Farben in das Blockzeilen-Farbenfeld kopieren, ausser in des letzte Feldelement
BlockZeilen_FarbFeld=new Array(SpielFlaeche_BlockZeilen_Anzahl);
for (BlockZeile_Zahler=0; BlockZeile_Zahler < SpielFlaeche_BlockZeilen_Anzahl; BlockZeile_Zahler++) // Index ab 0
{BlockZeilen_FarbFeld[BlockZeile_Zahler]=BlockZeile_MoeglicheFarbenFeld[BlockZeile_Zahler];}

// letztes Element im Blockzeilen-Farbenfeld immer der Block-Löschfarbe belegen -->
//   im Programm fest kodierter Bezug auf dieses Element !
BlockZeilen_FarbFeld[BlockZeilen_Anzahl]=SpielFlaeche_HintergrundFarbe;

// wenn BlockZeile_Zahler == 0
//   Farbe für Blockzeile, in der Block die maximale Berührungsanzahl nötig hat, eh er gelöscht wird
//   Farbe der obersten Blockzeile bei Spielstart
// wenn BlockZeile_Zahler == SpielFlaeche_BlockZeilen_Anzahl-2
//   Farbe für Blockzeile, in der Block mit der übernächsten Berührung gelöscht wird.
//   Farbe der vorletzten Blockzeile bei Spielstart
// wenn BlockZeile_Zahler == SpielFlaeche_BlockZeilen_Anzahl-1
//   Farbe für Blockzeile, in der Block mit der nächsten Berührung gelöscht wird.
//   Farbe der letzten Blockzeile bei Spielstart

// +++++ Blocknummer-Feld init
//   pro Blockzeile: die Zeilennummer als Blockstatus
BlockZeilenNummerFeld=new Array(BlockAnzahl_Gesamt);
BlockZeilenNummerFeld_Index=-1; // ab 0
for (BlockZeile_Zahler=0; BlockZeile_Zahler < SpielFlaeche_BlockZeilen_Anzahl; BlockZeile_Zahler++)
{
    for (BlockSpalte_Zahler=0; BlockSpalte_Zahler < SpielFlaeche_BlockSpalten_Anzahl; BlockSpalte_Zahler++)

```



```

{
    BlockZeilenNummerFeld_Index++; // fortlaufend
    BlockZeilenNummerFeld[BlockZeilenNummerFeld_Index]=BlockZeile_Zahler;
}
}

// +++++ Blockfeld auf der Spielfläche erzeugen
//      überlagert das Spielfeld --> siehe Eventhandler zum Spielfeld

//      Ursprung (0,0) des Grafik-Koordinatensystemes liegt links oben
//      Blockfeld besteht aus Blockzeilen und Blockspalten.
//      Blockspalte: Alle Spalten sind gleich lang
//      Obere Kante jeder Blockspalte bildet obere Grenze (Top) des Blockfeldes
//      Untere Kante jeder Blockspalte bildet untere Grenze (Down) des Blockfeldes
//      Blockzeile: Alle Zeilen sind gleich lang
//      Linke Kante jeder Blockzeile bildet linke Grenze (Left) des Blockfeldes
//      Rechte Kante jeder Blockzeile bildet rechte Grenze (Right) des Blockfeldes

BlockSpalte_Top= SpielFlaeche_AbstandVonFensterRand_Oben // >=0
    + BlockSpalte_AbstandVonSpielfeldRand_Oben; // >=0

BlockZeile_Left= SpielFlaeche_AbstandVonFensterRand_Links // >=0
    + BlockSpalte_AbstandVonSpielfeldRand_LinksBzwRechts; // >=0
IDZaehler=-1;
BlockZeilen_FarbFeldIndex=-1;

TopWert=BlockSpalte_Top - BlockSpalte_BlockAbstand;

// blockzeilenweise
for (BlockZeile_Zahler=0; BlockZeile_Zahler < SpielFlaeche_BlockZeilen_Anzahl; BlockZeile_Zahler++)
{
    // inits zu aktuellen Blockzeile
    //      nächste Farbe der Blockzeile einstellen
    BlockZeilen_FarbFeldIndex++;

    //      Topwert der aktuellen Blockzeile
    TopWert=TopWert+BlockSpalte_BlockAbstand;

    //      erster Leftwert, also Leftwert des 1. Blockes (Spalte)
    LeftWert=BlockZeile_Left-BlockZeile_BlockAbstand;

    // aktuellen Blockzeile blockweise (spaltenweise) abklappern
    for (BlockSpalte_Zahler=0; BlockSpalte_Zahler < SpielFlaeche_BlockSpalten_Anzahl; BlockSpalte_Zahler++)
    {
        // inits zum aktuellen Block der Blockzeile
        //      Leftwert ermitteln
        LeftWert=LeftWert+BlockZeile_BlockAbstand;

        //      nächstes ID des Blockes
        IDZaehler++;

        // Block erzeugen:
        // Als DIV: Da kein Inhalt vorhanden, so nichts angezeigt
        // Als TABLE: Es wird in der Zelle automatisch ein Inhalt erzeugt, also angezeigt
        document.write( '<TABLE ID="ID_Block' + IDZaehler + "'
            + ' WIDTH=' + Block_Breite
            + ' HEIGHT=' + Block_Hoehe
            + ' BGCOLOR=' + BlockZeilen_FarbFeld[BlockZeilen_FarbFeldIndex]
            + ' STYLE="position:absolute;'
            + 'top:' + TopWert + ';'
            + 'left:' + LeftWert
            + '""
            + '>'
            + '<TD></TD>' // leer !
            + '</TABLE>'
        );

        // Handler: Da JEDER einzelne Block die Spielfläche überlagert, bewirkt das Betreten des
        //      Blockes ein Verlassen der Spielfläche.

```



```

// Das Verlassen und Betreten der Spielfläche wird abgefragt per
// Div_SpielFlaeche_OnMouseOver_Handler
// Spiel wird als nicht gestoppt markiert
// Div_SpielFlaeche_OnMouseOut_Handler
// Spiel wird als gestoppt markiert und das auch dann,
// wenn Maus auf einem Element, das AUF der
// Spielfläche liegt, steht.
// z.B. Maus auf Block.
// Soll das Stoppen des Spieles mit Betreten eines Blocke verhindert werden:
// siehe Schläger !
}
}

// rechte Kante des letzten Blockes der letzten Zeile merken
// alle Zeilen sind ja gleich lang !
BlockZeile_Right=LeftWert+Block_Breite;
BlockSpalte_Down=TopWert+Block_Hoehe;

// Breite der Spielfläche anpassen
if (Div_SpielFlaeche_Breite != BlockZeile_Right)
{
    Div_SpielFlaeche_Breite=BlockZeile_Right;
    ID_Tabelle_SpielFlaeche.style.width=Div_SpielFlaeche_Breite;
}

// +++++ Schlagfläche-DIV anzeigen auf der Spielfläche
// überlagert das Spielfeld --> siehe Eventhandler zum Spielfeld
Wert1=2* Ball_Bild_Hoehe;
Schlaeger_AbstandVomFensterRand_Oben=Div_SpielFlaeche_Hoehe-Wert1 ;
// SpielFlaeche_AbstandVonFensterRand_Oben ist der Abstand
// von Schläger zur unteren Schpielflächengrenze
Schlaeger_Breite=1.5*Block_Breite;
Schlaeger_Hoehe=Math.floor(Block_Hoehe / 10);
if (Schlaeger_Hoehe < 4) {Schlaeger_Hoehe=4;}

Schlaeger_Breite_Gedrittelt=Math.floor(Schlaeger_Breite / 3);
Wert1=Div_SpielFlaeche_Breite-Schlaeger_Breite;
Schlaeger_StartLeft=Math.floor(Wert1 / 2);
Schlaeger_AktuellLeft=Schlaeger_StartLeft;

Schlaeger_MaxLeft= SpielFlaeche_AbstandVonFensterRand_Links // >=0
+ Div_SpielFlaeche_Breite
- Schlaeger_Breite;

// Als DIV: Da kein Inhalt vorhanden, so nichts angezeigt
// Als TABLE: Es wird in der Zelle automatisch ein Inhalt erzeugt, also angezeigt
document.write( '<TABLE ID="ID_Div_Schlaeger"'
+ ' WIDTH=' + Schlaeger_Breite
+ ' HEIGHT=' + Schlaeger_Hoehe
+ ' BGCOLOR="' + Schlaeger_Farbe + "'
+ '>'
+ '<TD></TD>' // leer !
+ '</TABLE>'
);

StyleZeiger=ID_Div_Schlaeger.style;
StyleZeiger.position='absolute';
StyleZeiger.posTop =Schlaeger_AbstandVomFensterRand_Oben;
StyleZeiger.posLeft=Schlaeger_StartLeft;

// Handler: Da der Schläger die Spielfläche überlagert, bewirkt das Betreten des
// Schlägers ein Verlassen der Spielfläche.
// Das Verlassen und Betreten der Spielfläche wird abgefragt per
// Div_SpielFlaeche_OnMouseOver_Handler
// Spiel wird als nicht gestoppt markiert
// Div_SpielFlaeche_OnMouseOut_Handler
// Spiel wird als gestoppt markiert und das auch dann,
// wenn Maus auf einem Element, das AUF der
// Spielfläche liegt, steht.

```



```

//          z.B. Maus auf Schläger.
//      Da die Maus auf Schläger auf keinen Fall das Spiel stoppen darf,
//      MÜSSTE im Event-Handler zur Spielfläche der Zustand "Maus auf Schläger"
//      abgefragt werden.
//      Da die Verarbeitung der Eventfolge einer Trägheit und internen Reihenfolge des
//      Browsers unterliegt, kann es sein, dass der Eventhandler zur Spielfläche
//      nicht direkt nach dem mouseover auf den Schläger erfolgt.
//      Daher muss also im Eventhandler des Schlägers verhindert werden, dass das
//      Spiel stoppt.
ID_Div_Schlaeger.onmouseover=Div_Schlaeger_OnMouseOver_Handler;
// setzt AktivesSpiel_Pausiert auf false
ID_Div_Schlaeger.onmousemove=Div_SpielFlaeche_OnMouseMove_Handler;
// lässt Bewegung des Schlägers auch dann zu, wenn
//   Maus auf dem Schläger steht.
// Hinweis: Div_SpielFlaeche_OnMouseMove_Handler sorgt für die
//   Bewegung des Schlägers.

// +++++ Ball-DIV anzeigen auf der Spielfläche
//   überlagert das Spielfeld --> siehe Eventhandler zum Spielfeld

// Bild vorladen
var BallObjekt=new Image(Ball_Bild_Breite,Ball_Bild_Hoehe);
BallObjekt.src=Ball_Bild_Name;

// Anzahl verbrauchbarer Bälle
Ball_Nummer=BallNummer_Maximal;

// Startrichtung des Balles: zufällig für horizontal
Ball_Schritt_Horizontal_StartRichtung=-1; // nach links oder rechts
if (Math.random() < 0.5){Ball_Schritt_Horizontal_StartRichtung=1;}
Ball_Schritt_Vertikal_StartRichtung=-1; // immer nach oben

// GENERELLE Schrittweite des Balles
Ball_SchrittWeite_Horizontal=8; // ganzzahlig teilbar !!!
Ball_SchrittWeite_Horizontal_Halbiert=Ball_SchrittWeite_Horizontal / 2;

Ball_SchrittWeite_Vertikal=8; // ganzzahlig teilbar !!
Ball_SchrittWeite_Vertikal_Halbiert=Ball_SchrittWeite_Vertikal / 2;

// Startposition des Balles
Ball_Start_Top=Schlaeger_AbstandVomFensterRand_Oben-Ball_Bild_Hoehe;
Ball_Start_Left=Math.floor((Div_SpielFlaeche_Breite-Ball_Bild_Breite) / 2);

// maximaler Bereich der Bewegung des Balles
Ball_Max_Left=  SpielFlaeche_AbstandVonFensterRand_Links
                + Div_SpielFlaeche_Breite
                - Ball_Bild_Breite;

Ball_Max_Top_BisSchlaeger=  Schlaeger_AbstandVomFensterRand_Oben
                           - Ball_Bild_Hoehe;

Ball_Max_Top_BisHinterSchlaeger=  Schlaeger_AbstandVomFensterRand_Oben
                                  + Ball_Bild_Hoehe;

// Ballgeschwindigkeit wird beim Speed-Div ermittelt (siehe tiefer)

document.write( '<DIV ID="ID_Div_Ball"'
                + ' WIDTH=' + Ball_Bild_Breite
                + ' HEIGHT=' + Ball_Bild_Hoehe
                + '>'
                + '<IMG ID="ID_Img_Ball"'
                + ' WIDTH=' + Ball_Bild_Breite
                + ' HEIGHT=' + Ball_Bild_Hoehe
                + ' BORDER=0'
                + ' GALLERYIMG="no"'
                + '>'
                + '</DIV>'
                );

```



```

StyleZeiger=ID_Div_Ball.style;
StyleZeiger.position='absolute';
StyleZeiger.posTop =Ball_Start_Top;
StyleZeiger.posLeft=Ball_Start_Left;
ID_Img_Ball.src=BallObjekt.src;

// Handler: Das das Bild die Spielfläche überlagert, bewirkt das Betreten des
// Bildes ein Verlassen der Spielfläche.
// Das Verlassen und Betreten der Spielfläche wird abgefragt per
// Div_SpielFlaeche_OnMouseOver_Handler
// Spiel wird als nicht gestoppt markiert
// Div_SpielFlaeche_OnMouseOut_Handler
// Spiel wird als gestoppt markiert und das auch dann,
// wenn Maus auf einem Element, das AUF der
// Spielfläche liegt, steht.
// z.B. Maus auf Block.
// Das Stoppen des Spieles mit Betreten des Bildes wird analog zum Schläger verhindert.
ID_Div_Ball.onmouseover=Div_Ball_OnMouseOver_Handler;

// +++++ Startbutton-DIV erzeugen auf der Spielfläche
// überlagert das Spielfeld --> siehe Eventhandler zum Spielfeld
// muss innerhalb der Spielfläche liegen wegen AktivesSpiel_Pausiert
Wert1=4* Ball_Bild_Hoehe;
Div_StartButton_Top =Schlaeger_AbstandVomFensterRand_Oben -Wert1;
Div_StartButton_Left=Schlaeger_StartLeft;

document.write( '<DIV ID="ID_Div_StartButton">'
+ '<INPUT TYPE="button"'
+ ' VALUE="'+ Div_StartButton_Label + '"'
+ ' onclick="Spiel_Start();blur();"'
+ '>'
+ '</DIV>'
);

StyleZeiger=ID_Div_StartButton.style;
StyleZeiger.position='absolute';
StyleZeiger.posTop =Div_StartButton_Top;
StyleZeiger.posLeft=Div_StartButton_Left;

// +++++ Div der Anzeige im Spielfeld erzeugen, aber nicht anzeigen
// überlagert das Spielfeld --> siehe Eventhandler zum Spielfeld
// keine eigenen Eventhandler, da reine Textanzeige zum Zeitpunkt,
// vor dem das Spiel beginnt
// nach dem das Spiel geendet ist

Wert1=6* Ball_Bild_Hoehe;
Div_Anzeige_Top =Div_StartButton_Top - Wert1;
Div_Anzeige_Left=Div_StartButton_Left;

document.write( '<DIV ID="ID_Div_Anzeige">'
+ '<FONT STYLE="'+ Div_Anzeige_Font + '">'
+ AnzeigeText_AllClear // egal welcher Text, nur nicht leer
+ '</FONT>'
+ '</DIV>'
);

StyleZeiger=ID_Div_Anzeige.style;
StyleZeiger.position='absolute';
StyleZeiger.visibility='hidden';
Div_Anzeige_Angezeigt=false;
StyleZeiger.posTop =Div_Anzeige_Top;
StyleZeiger.posLeft=Div_Anzeige_Left-40;

// Anzeige generell zulassen
Div_GamePause_Angezeigt=false; // true, so AnzeigeText_GamePause bereits sichtbar
Div_Anzeige_Erlaubt=true; // Anzeige zulassen

// +++++ Info als DIV anzeigen rechts NEBEN dem Spielfeld

```



```

Div_Info_Left=Div_SpielFlaeche_Breite+38;

document.write( '<DIV ID="ID_Div_Info">'
    + '<PRE>'
    + '<FONT STYLE="" + Div_Info_Font + "">'
    + '<B>BREAK OUT</B>'
    + '</FONT>'
    + '<BR>'
    + '<DIV ID="ID_Div_BallNummer_Anzeige"></DIV>'
    + '<BR>'
    + '<DIV ID="ID_Div_ClearedBlocks_Anzeige"></DIV>'
    + '</PRE>'
    + '</DIV>'
);

StyleZeiger=ID_Div_Info.style;
StyleZeiger.position='absolute';
StyleZeiger.posTop =SpielFlaeche_AbstandVonFensterRand_Oben;
StyleZeiger.posLeft=Div_Info_Left;

// Startwerte der veränderlichen Infos anzeigen
Ball_NummerAnzeigen();
Ball_GeloeschteBlocksAnzahlAnzeigen();

// +++++ Speed-Button als DIV anzeigen rechts NEBEN dem Spielfeld

// Wertebereich prüfen
// maximaler Speed=kleinster Wert
// minimaler Speed=grösster Wert

// Annahme: Wertebereich nicht eingehalten
Wert1=10; // als Ersatz für Max-Speed
Wert2=100; // als Ersatz für Min-Speed

if ( (Ball_Geschwindigkeit_Max > 0) // Millisekunden pro Bewegungsschritt, nur 10, 20, 30 etc.
    && (Ball_Geschwindigkeit_Min > 10) // Millisekunden pro Bewegungsschritt, nur 10, 20, 30 etc.
    && (Ball_Geschwindigkeit_Max > Ball_Geschwindigkeit_Min)
)
{ // Wertebereich wurde eingehalten
Wert1=Ball_Geschwindigkeit_Max;
Wert2=Ball_Geschwindigkeit_Min;
}

Wert1=Ball_Geschwindigkeit_Max;
Wert2=Ball_Geschwindigkeit_Min;

// Anfangswert der Geschwindigkeit
Ball_Geschwindigkeit_Aktuell=Ball_Geschwindigkeit_Min * 0.4;
Ball_Geschwindigkeit_Aktuell_Anzeige=Ball_Geschwindigkeit_Aktuell;

// Erhöhung-Schrittweite, daher Max und Min nur 1, 10, 20 ....
Ball_Geschwindigkeit_SchrittWeite=10;

// Speed-Erhöhung erlauben, also Anfangswert senken
Ball_Geschwindigkeit_Vorzeichen=-1; // 1 für Wert erhöhen, also Speed senken
// -1 für Wert senken, also Speed erhöhen

// Text des Buttons
Ball_Geschwindigkeit_Aendern_Kette=Ball_Geschwindigkeit_Text2; // Annahme: Wert senken, also Speed erhöhen
if (Ball_Geschwindigkeit_Vorzeichen > 0)
{Ball_Geschwindigkeit_Aendern_Kette=Ball_Geschwindigkeit_Text1;} // Wert erhöhen, also Speed senken

Kette=Ball_Geschwindigkeit_Aendern_Kette + Ball_Geschwindigkeit_Aktuell_Anzeige.toString();

// Button rendern
Ball_Geschwindigkeit_Button_Top=BlockSpalte_Down;
Ball_Geschwindigkeit_Button_Left=Div_Info_Left;

document.write( '<DIV ID="ID_Div_BallGeschwindigkeit">'

```



```

        + '<INPUT ID="ID_Input_BallGeschwindigkeit"'
        + ' TYPE="button"'
        + ' VALUE="" + Kette + ""'
        + '>'
        + '</DIV>'
    );

StyleZeiger=ID_Div_BallGeschwindigkeit.style;
StyleZeiger.position='absolute';
StyleZeiger.posTop =Ball_Geschwindigkeit_Button_Top;
StyleZeiger.posLeft=Ball_Geschwindigkeit_Button_Left;
ID_Div_BallGeschwindigkeit.onclick=Ball_Geschwindigkeit_Aendern;

// +++++ Reset-Button als DIV anzeigen rechts NEBEN dem Spielfeld

document.write( '<DIV ID="ID_Div_Reset">'
    + '<INPUT ID="ID_Input_Reset"'
    + ' TYPE="button"'
    + ' VALUE="Nächster Ball"'
    + '>'
    + '</DIV>'
);

StyleZeiger=ID_Div_Reset.style;
StyleZeiger.position='absolute';
StyleZeiger.posTop =Ball_Geschwindigkeit_Button_Top + 100;
StyleZeiger.posLeft=Ball_Geschwindigkeit_Button_Left;
ID_Div_Reset.onclick=Spiel_Start_NaechsterBall;

// +++++ Sound leer rendern
document.write('<BG SOUND ID="ID_Sound" LOOP="1">'); // Sound 1x wiedergeben
}
// -->
</SCRIPT>
</HEAD>

<BODY>
<SCRIPT LANGUAGE="JScript">
<!--
init();
// -->
</SCRIPT>
</BODY>
</HTML>

```

24. ActiveX-Controls

Microsoft ändert fortlaufend die Active-X-Eigenschaften von Windows und somit auch des Internet Explorers

Diese fortlaufenden Änderungen muss der Programmierer in Erfahrung bringen.

Der Programmierer kann sich definitiv nicht auf Verfügbarkeit von Active-X-Controls verlassen und muss damit rechnen, dass seine Webseiten schlagartig nicht mehr komplett laufen weil u.a. Programmcode noch nicht angepasst ist. Ebenfalls muss der Programmierer Varianten von Windows und Patchzustände beachten, die prinzipiell Kostenprobleme verursachen können.

Mit anderen Worten: Wer Microsoft-Komponenten nutzt, muss wissen, was ihm blüht ... siehe nachfolgende Beispiel für Risiken.

Prinzipielle Lizenzprobleme für den Programmierer

Microsoft verlangt Lizenzierung von Windows. Bezüglich Windows-Versionen gibt es die Updatestufen z.B. per Servicepacks

Ein Windows mit Servicepack fällt unter die Lizenz des geupdateten Windows.

Ein Windows mit Vorversion zum Servicepack bedarf einer anderen Lizenz.

Will man z.B. den Internet Explorer 7 und 6 parallel testen, benötigt man 2 Windowslizenzen, da beide Versionen nicht parallel installierbar. Dazu kommt, dass es den IE 6 in 2 Versionen gibt: Win SP1 und SP2 (IE 7 nur ab Win SP2). Für 3 Browserversionen benötigt man 3 Windowslizenzen, will man parallel testen.

Ein Blick auf Browser-Konkurrenzprodukte klärt die Sachlage unschlagbar: Opera ist z.B. parallel installierbar.

Hinweis: Man suche doch mal im Internet nach einem kostenlosen HTTP-Server vom Microsoft, um IE-Seite testen zu können, die JScript nutzen (inklusive Debugger). Denn sollte kein kostenloses Angebot findbar sein, kommen die Kosten von Entwicklungssoftware zum IE hinzu. Ein Blick auf Konkurrenz-HTTP-Server klärt die Sachlage: Apache-HTTP-



Server ist kostenlos, allerdings nicht einfach einzurichten (Hinweis: Der HTTP-Server sollte virtuelle Hosts einrichten können und korrekt mit der Firewall des Users zusammenarbeiten können).

Abänderungen wegen Sicherheitspatches der jeweiligen Windows-Versionen

Abschaltungen von Active-X-Controls erfolgen auch im Rahmen der Sicherheitspatches zu Windows-Versionen. Es ist auch möglich, dass wegen Sicherheitslücken abgeschaltet wird und somit Komponenten einer Webseite je nach Windowsversion nicht mehr laufen.

Im Rahmen der Sicherheitspatches ist es Microsoft sogar gelungen, Webseiten, die den MS-Encoder zur Komprimierung von HTML- und JScript-Code nutzen, schlagartig unnutzbar zu machen: Ein Bug in einem Patch zu Windows XP - Q918899 Das Patch verursacht IE-Browser-Absturz bei per MS ScriptEncoder gepacktem JScript unter SP1 und 2 wenn HTTP 1.1 mit Kompression genutzt wird z.B. bei onclick-Handler auf IMG klick ins Fenster per aktivem Popup

Der Absturz ist "read" -Fehler von immer ein und derselben Speicherstelle.

User, die dieses Patch installiert haben, können ab sofort keine IE-Seiten mit codiertem Script mehr ansehen.

Microsoft stellt Abhilfe nach geraumer Zeit zur Verfügung, jedoch spezifisch nach Windows XP-Version:

Patch Q918899 für

Windows XP SP1
SP2

Download für jedermann bereitgestellt

nur auf kostenpflichtige telefonische Anfrage des Users per Downloadlink bereitgestellt, da

Microsoft explizit die User registriert haben will, bei denen das

Patchproblem auftritt (User muss sich Telefonnummer besorgen)

Solange also das Patch zum fehlerhaften Patch vom User nicht installiert wird,

z.B. weil der User keine Ahnung hat, dass und wo er sich die Telefonnummer

von Microsoft besorgen muss bzw. zu besorgen hat, wird der User

IE-Seiten mit komprimierten Code dauerhaft nicht nutzen können.

(Microsoft-Support ist z.T. nur in Englisch).

Abänderungen wegen Browser-Inkompatibilität

Popupblocker-Fehler

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()

Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.

Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.

Der Fehler tritt nicht auf, wenn ein Fenster per window.open() erzeugt wurde.

Bedingung:

Scriptfehleranzeige ist erlaubt im IE 7

Popupblocker ist im IE abgeschaltet

ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 window.popup per .show() erzeugt.

ein weiteres Fenster (Register) z.B. leere Seite (about:blank)

beide (Register) liegen in einer gemeinsamen IE-Instanz

Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per .show() erzeugt,

bricht der Browser das Dokument mit .show() ab (Scriptfehler).

Der Popupblocker für die leere Seite verursacht den Programmfehler im Dokument mit .show(). Es wird folgende

Meldung angezeigt (in der Informationsleiste):

'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'

Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:

Der Popupblocker hat ein Popupfenster geblockt. Sie können den Popupblocker deaktivieren

oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.

Die Realität zur obigen Meldung ist völlig anders:

Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter

Popupblocker einschalten

weitere Informationen

jedoch keine Möglichkeit wie laut Bedeutung

Damit gilt: Der abgeschaltete Popupblocker ist in Wirklichkeit aktiv.

Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster einer Windowsanwendung z.B. einer anderen IE-Instanz)

Der Popupblocker des IE bemerkt aber NUR Webseite, die das Popup erzeugt.

Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer

anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popupblocker verwaltet wird.

Der Popupblocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen

Webseiten (die nicht das Popup erzeugt haben).

Der Popupblocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.

Der Popupblockerfehler verändert die Eventverwaltung:

Es werden u.a. ignoriert

onfocus




```

onblur
onfocusin
onfocusout
und viele andere, so dass trotz Events z.B. des Body der Popublockerfehler entsteht.

// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell
window.focus();
window.document.focus();
if(document.body!=null)
{if(document.body.style!='hidden')          // wenn hidden so focus() nicht möglich (Scriptfehler erzeugt)
{document.body.focus();}
}
// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt
popupzeiger.show(...);

```

Hinweis: Der Popupfehler ist so elementar, dass die vielen Beta-Testphasen des IE mehr als fragwürdig erscheinen, wie die Angabe von Microsoft, dass Code neu programmiert wurde, um den IE sicherer zu machen.

focus-Methode beim IE 7

windows.focus() document.focus() und body.focus() funktionieren NICHT
zwischen Register in einem IE-Fenster
zwischen Fensters z.B. in Taskleiste

Hinweis:

.focus() setzt Element aktiv, gibt dem Element den Focus und feuert dann onfocus
.setActive() ist Teilmenge von .focus(): nur das aktiv setzen
funktioniert nicht mit allen Elementen, mit denen .focus() funktioniert

animierte Gif (mit Timer)

Animierte Gifs (mit Timer), die unter IE 6 korrekt laufen, müssen unter IE 7 im Timer nicht mehr laufen:
z.B. garnicht mehr sichtbar, oder Timer nicht verwendet.
Dann müssen animierte Gif-Bilder nach IE-Version bereitgestellt werden.

Abänderungen wegen Rechtstreitigkeiten von Microsoft mit Fremdanbietern

Ein sehr bekanntes Beispiel ist die nachträglich eingeführte Einschränkung von Active-X-Controls wegen Patentwahrung durch Microsoft, wobei für den JScript-Programmierer massive Änderungen eintreten.

Wegen Patentwahrung hat Microsoft ein zunächst freiwilliges Patch herausgegeben, dass bei ActiveX-Control per APPLET, EMBED oder OBJECT, die auf dem Bildschirm rendern (mit oder ohne Userschnittstelle), dafür sorgt, dass bei mouseover über das Control eine Sprechblase erscheint, die darauf hinweist, dass das Objekt als ActiveX-Control klickbar ist.
Diese Sprechblase erscheint auch, wenn das Control keine Userschnittstelle hat, also diese gar nicht klickbar ist.

Es wurde das Eventmodell gleichzeitig geändert:

Es werden alle Events solange unterdrückt, bis der User die Sprechblase geklickt hat.
Das Klicken muss auf das Objekt im Sprechblasenrahmen erfolgen, der so groß ist, wie die Dimension, in der gerendert wurde.
Es muss also ERST per Mausklick das Control aktiviert werden, ehe das Control klickbar und damit die Eventsteuerung aktiviert ist.
Ein Control, dass programmtechnisch zwar was rendert, aber ansonsten ohne sichtbare programmtechnisch startet, muss ebenfalls geklickt werden, obwohl es bereits läuft und es nichts zu klicken gäbe (wenn keine Eventsteuerung eingebaut wurde).
Wegen blockierter Eventsteuerung ist also die Sprechblase z.B. nicht automatisch klickbar.
Die Eventauslösung per nicht-objekteigenen Eventhandler, der für das Objekt per fireEvent() ein Event auslöst, ist solange blockiert, bis der User die Sprechblase geklickt hat.

style.visibility='hidden' wird ignoriert

Die Sprechblase erscheint auch dann, wenn das Control mit style.visibility='hidden' belegt ist, also sich unsichtbar rendert: Der Sprechblasenrahmen hat genau die Dimension wie die des unsichtbaren Controls. Der Sprechblasenrahmen erscheint also Zusammenhangslos, und der User weiß nicht, warum er klicken soll, wenn er nichts sieht. Vor allem weiß er nicht, WAS er klickt ... ideale Basis für Schadsoftware per Script.

Diese Sprechblase erscheint nur DANN NICHT, wenn die Userschnittstelle mit Breite == Höhe == 0 gerendert wird. Sollte die Userschnittstelle in einem Container liegen, z.B. DIV, dann wird der Container, wenn er in der Dimension kleiner ist, also die Userschnittstelle, angepasst. Daher muss der Container ebenfalls mit Breite == Höhe == 0 gerendert



werden. Wegen Dimensionierung auf 0 sollte `style.visibility="hidden"` sein. Im Falle eines Containers reicht es, den `style` des Containers zu ändern, da `visibility` normalerweise vererbt wird an Kinder, also auch an das Control.

Abänderung wegen Abschaltungen

DirectX ist wegen Abschaltung von Active-X--Controls nicht mehr abwärtskompatibel:

Z.B. wurde bei Win XP SP2 Direct Animation aus DirectX schlagartig durch Abschaltung von Bibliotheken dezimiert, die es bei Win XP SP1 aber noch gibt.

Hier ein Beispiel aus dem Jahr 2004: Abschaltungen von Active-X-Controls

ActiveX-Controls und Unterstützung/Verbot 20041215

erlaubt sind noch

Tabular Data-Steuerelement {333C7BC4-460F-11D0-BC04-0080C7055A83} Das TDC (Tabular Data-Steuerelement) ermöglicht die Weiterverarbeitung von Daten, die nur im Textformat vorliegen, beispielsweise durch Darstellung in einer Tabelle oder Sortierung. Weitere Informationen:•

http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp(http://msdn.microsoft.com/workshop/database/tdc/tabular_data_control_node_entry.asp)

Microsoft Agent Control - Version 2.0 {D45FD31B-5C6E-11D1-9EC1-00C04FD7081F} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Weitere Informationen:•

<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>(<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Microsoft MSChat-Steuerelement-Objekt 2.0 - 2.5 {D6526FE0-E651-11CF-99CB-00C04FD64497}

Dieses Steuerelement wird von Webautoren verwendet, um text- und graphisch basierte Chatgemeinden für Echtzeitkonversationen im Web zu erstellen.

Microsoft ActiveX Upload-Steuerelement, Version 1.5 {886e7bf0-c867-11cf-b1ae-00aa00a3f2c3} Dieses Steuerelement kann auf vielerlei Art genutzt werden, um auf einfache Weise Webinhalte via Drag and Drop zu veröffentlichen. Weitere Informationen:• 230298 (<http://support.microsoft.com/kb/230298/DE/>) - Posting Acceptor Release Notes

• http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp
(http://msdn.microsoft.com/workshop/management/tools/reference/file_upload_control.asp)

verboten sind

Datenbindung RDS {BD96C556-65A3-11D0-983A-00C04FC29E36} {BD96C556-65A3-11D0-983A-00C04FC29E33} Die RDS (Remote Data Service) Steuerelemente ermöglichen dem Browser, client-basierte SQL Abfragen an einen Webserver zu stellen. Inzwischen wurde RDS jedoch durch neuere Standards wie SOAP abgelöst, von einer weiteren Verwendung von RDS wird daher abgeraten. Weitere Informationen:• 184375 (<http://support.microsoft.com/kb/184375/DE/>) - Sicherheitsaspekte bei RDS 1.5, IIS 3.0 oder 4.0 und ODBC

<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>
(<http://msdn.microsoft.com/library/en-us/iissdk/iis/remotedatabindingwithremotedataservice.asp>)

http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp
(http://msdn.microsoft.com/library/en-us/dnmdac/html/data_mdacroadmap.asp)



XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer {550dda30-0541-11d2-9ca9-0060b0ec3d39} {CFC399AF-D876-11d0-9C10-00C04FC99C8E} {e54941b2-7756-11d1-bc2a-00c04fb925f3} {7108ECB4-AFDC-11D1-ADC1-00805FC752D8} XMLDSO, XMLDocument, DOMDocument, und XMLIslandPeer ermöglichen die Verarbeitung von XML Daten, etwa die Bindung von HTML Elementen an einen XML Datensatz, oder das Einlesen, Manipulieren, und Zurückschreiben von XML Daten.

Die Steuerelemente DOMDocument und XMLIslandPeer bzw. die dazugehörigen ClassIDs sind nicht mehr aktuell, so dass von einer generellen Freigabe dieser Steuerelementgruppe abgeraten wird. Weitere Informationen:•
http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp(http://msdn.microsoft.com/library/en-us/xmlsdk/htm/xml_concepts2_7ook.asp)

Internet Explorer

Active Setup / IE Active Setup-Steuerelement {F72A7B0E-0DD8-11D1-BD6E-00AA00B92AF1} Dieses Steuerelement enthält die in Microsoft Security Bulletin MS99-037 beschriebene Sicherheitsanfälligkeit. Um eine weitere Ausführung zu verhindern wurde im Rahmen dieses Security Bulletins ein Kill-Bit gesetzt, so dass selbst bei einer Freigabe dieses Controls eine Ausführung blockiert wird. Weitere Informationen:•

<http://www.microsoft.com/technet/security/bulletin/ms99-037.msp>
(<http://www.microsoft.com/technet/security/bulletin/fq99-037.msp>)

240797 (<http://support.microsoft.com/kb/240797/DE/>) - So verhindern Sie die Ausführung von ActiveX-Steuerelementen in Internet Explorer

Media Player / Active Movie Runtime {A4001DE0-7075-11d0-89AB-00A0C9054129} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Runtime Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / ActiveMovie-Steuerelement {05589FA1-C356-11CE-BF01-00AA0055595A} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das Active Movie Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Microsoft NetShow Player {2179C5D3-EBFF-11CF-B6FD-00AA00B4E220} Die Funktionalität dieses Steuerelements wird nun durch das Windows Media Player ActiveX Steuerelement abgedeckt. Das NetShow Player Steuerelement wird daher nicht mehr unterstützt, von einer Freigabe wird abgeraten.

Media Player / Windows Media Player {22D6F312-B0F6-11D0-94AB-0080C74C7E95} Dies ist das Steuerelement für Windows Media Player version 6.4 und war Installationsbestandteil bis einschließlich Windows Media Player Version 8. Ab Windows Media Player 9 wurde diese ClassID durch die neue ClassID {6BF52A52-394A-11D3-B153-00C04F79FAA6} abgelöst, deren Verwendung stattdessen empfohlen wird. Ab Windows Media Player Version 9 wird ferner die alte ClassID anhand eines Wrappers automatisch auf die neue ClassID umgeleitet. Die ClassID für Windows Media Player Version 9 ist jedoch nicht in der Liste der vom Administrator genehmigten Steuerelemente enthalten, und muss bei Bedarf manuell hinzugefügt werden.

Animierte Schaltflächen {0482B100-739C-11CF-A3A9-00A0C9034920} Dieses Steuerelement erlaubte in frühen Versionen des Internet Explorer die Verwendung animierter Schaltflächen auf Webseiten. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von der Freigabe des Steuerelements wird daher abgeraten.



IE Label-Steuerelement

{99B42120-6EC7-11CF-A6C7-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 auch kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

IE Menu-Steuerelement {74701400-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ermöglicht die Handhabung von Menüstrukturen in Webseiten, wird jedoch nicht mehr unterstützt und dürfte nur noch selten Verwendung finden. Von einer Freigabe des Steuerelements wird daher abgeraten.

IE Preloader-Steuerelement {16E349E0-702C-11CF-A3A9-00A0C9034920} Dieses Steuerelement ermöglichte das Vorladen von Webseiten, ist jedoch inzwischen nicht mehr aktuell, wird nicht mehr unterstützt und dürfte nicht mehr im Einsatz sein. Aufgrund einer potentiellen Sicherheitsanfälligkeit in diesem Steuerelement wird von einer Freigabe abgeraten. Weitere Informationen: • 231452 (<http://support.microsoft.com/kb/231452/DE/>) - Update Available for "Legacy ActiveX Control" Issue

IE Timer-Steuerelement {59CCB4A0-727D-11CF-AC36-00AA00A47DD2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

MCSiMenü {275E2FE0-7486-11D0-89D6-00A0C90C9B67} Dieses Steuerelement dient der Anpassung von Popupmenüs, ist jedoch nicht mehr aktuell und wurde nach Windows 98 nicht mehr ausgeliefert. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten.

Popupmenüobjekt {7823A620-9DD9-11CF-A662-00AA00C066D2} Dieses Steuerelement ist nicht mehr aktuell und seit Internet Explorer Version 5 kein Bestandteil der Installation mehr. Das Steuerelement wird nicht mehr unterstützt und dürfte nur noch vereinzelt im Einsatz sein. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • 190045 (<http://support.microsoft.com/kb/190045/DE/>) - INFO: ActiveX Controls That Are Removed from Internet Explorer 5

Microsoft Agent Control - Version 1.5 {F5BE8BD2-7DE6-11D0-91FE-00C04FD701A5} Microsoft Agent repräsentiert die neue Generation des ursprünglichen Office-Assistenten. Anstatt den Assistenten jedoch innerhalb eines Rahmens darzustellen wird hier lediglich der Charakter bzw. Agent selbst dargestellt und kann auch in Webseiten verwendet werden. Diese Version des Steuerelements ist jedoch nicht mehr aktuell und wird nicht mehr unterstützt. Von einer Freigabe des Steuerelements wird daher abgeraten. Weitere Informationen: • <http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>
(<http://msdn.microsoft.com/library/partbook/egvb6/introducingmicrosoftagent.htm>)

Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden. Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS



exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Wenn der IE 6.x aktive Inhalte blockiert, wird NOSCRIPT-Tag aktiviert, Ausnahme: Frameset

FRAMESET ist ein aktiver Inhalt:

Da der Frameset anstelle <BODY> kodiert sein muss, gilt:

Alle Tags, die für BODY zulässig sind, werden ignoriert, auch NOSCRIPT.

Wird neben Frameset noch BODY kodiert, so wird Frameset ignoriert.

Die Freigabe der Scriptblockierung erzeugt Ausführung aller Script-Teile inklusive der Eventauslösungen

Bsp.: Folgendes funktioniert vom Dokument, das window.open() hat im geöffneten Dokument (Quelltext im Dokument das window.open() verwendet):

```
function Y_unload(X00){X85[X00].close();}

var X85=new Array();var X86=new Array();

X85[0]=window.open(...);

var X87='parent.Y_unload(0)'; X86[0]=new Function(",X87);

X85[0].document.body.onunload=X86[0];
```

Wird die Scriptblockierung im geöffneten Fenster abgeschaltet,
so wird das Fenster geschlossen, weil onunload ausgelöst wird.

Achtung: document.body.onunload funktioniert ev. nicht mehr
wenn z.B. mit attachevent() aktiviert wurde

Folgende Metatags sind für den IE 6.x aktiver Inhalt:

```
<META HTTP-EQUIV="imagetoolbar" CONTENT="no">
    unterdrückt NICHT IE-Kontextmenü rechte Maus auf Bild

<META HTTP-EQUIV="site-enter" CONTENT="revealtrans(duration=0.3, transition=12) ">
<META HTTP-EQUIV="site-exit" CONTENT="revealtrans(duration=0.3, transition=12) ">
```

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.
Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein, innerhalb dessen dann die neuen HTML-Elemente erzeugt werden.

24.1. Analog-Uhr

```
<HTML>
<HEAD>
```

```
<SCRIPT LANGUAGE="JScript">
<!--
```

```
// Analoguhr
// nur IE
// Es werden Active-X-Controls verwendet (DirectX)

// ##### veränderbare Variablen #####

// +++++ Uhr +++++

// Dimension der Uhr
var AnalogUhr_BreiteUndHoehe=330; // >=140 und <=330

// Lage der Uhr, alles Stringangaben
var AnalogUhr_Top=20;
var AnalogUhr_Left=20;

// +++++ Ziffern +++++

var AnalogUhr_Ziffern_Erzeugen=true; // false für keine Ziffern
var AnalogUhr_Ziffer_Fett=false; // false für nicht fett

var AnalogUhr_Ziffer_Standard='.';
```



```

var AnalogUhr_Ziffer_Stunde_Viertel='*';
var AnalogUhr_Ziffer_Stunde_Halbe='+';
var AnalogUhr_Ziffer_Stunde_DreiViertel='*';
var AnalogUhr_Ziffer_Stunde_Voll='#';

var AnalogUhr_Ziffern_Satz=2; // 0 für frei definiert per o.g. Variablen
                        // 1 für römisch (vordefiniert)
                        // 2 für normal (vordefiniert)

// Farbe aller Ziffern, alles Stringangaben
var AnalogUhr_Ziffer_Farbe_RotAnteil='255';
var AnalogUhr_Ziffer_Farbe_GruenAnteil='255';
var AnalogUhr_Ziffer_Farbe_BlauAnteil='255';

// +++++ Zeiger +++++
// Sekundenzeiger erzeugen
var AnalogUhr_Zeiger_Sekunden_Erzeugen=true; // false für keinen Sekundenzeiger

// Zeigerform
var AnalogUhr_Zeiger_Flaeche=true; // false für hohl (falls überhaupt renderbar)

// Farbe des Sekunden-Zeigers, alles Stringangaben
var AnalogUhr_Zeiger_Sekunden_Farbe_RotAnteil='255';
var AnalogUhr_Zeiger_Sekunden_Farbe_GruenAnteil='0';
var AnalogUhr_Zeiger_Sekunden_Farbe_BlauAnteil='0';

// Farbe des Minuten-Zeigers, alles Stringangaben
var AnalogUhr_Zeiger_Minuten_Farbe_RotAnteil='170';
var AnalogUhr_Zeiger_Minuten_Farbe_GruenAnteil='255';
var AnalogUhr_Zeiger_Minuten_Farbe_BlauAnteil='255';

// Farbe des Stunden-Zeigers, alles Stringangaben
var AnalogUhr_Zeiger_Stunden_Farbe_RotAnteil='204';
var AnalogUhr_Zeiger_Stunden_Farbe_GruenAnteil='238';
var AnalogUhr_Zeiger_Stunden_Farbe_BlauAnteil='255';

// +++++ Zeiger-Ursprung +++++
// Farbe des Zeiger-Ursprunges, alles Stringangaben
var AnalogUhr_Zeiger_Ursprung_Farbe_RotAnteil='255';
var AnalogUhr_Zeiger_Ursprung_Farbe_GruenAnteil='255';
var AnalogUhr_Zeiger_Ursprung_Farbe_BlauAnteil='0';

// +++++ Hintergrund der Uhr +++++
// Farbe des Hintergrundes, alles Stringangaben
var AnalogUhr_Hintergrund_Farbe_RotAnteil='26';
var AnalogUhr_Hintergrund_Farbe_GruenAnteil='78';
var AnalogUhr_Hintergrund_Farbe_BlauAnteil='133';

// Farbe des Hintergrund-Rahmens, alles Stringangaben
var AnalogUhr_HintergrundRahmen_Farbe_RotAnteil='200';
var AnalogUhr_HintergrundRahmen_Farbe_GruenAnteil='200';
var AnalogUhr_HintergrundRahmen_Farbe_BlauAnteil='200';
// -->
</SCRIPT>

<SCRIPT LANGUAGE="VBScript">

' ##### Steuerung der Uhr anhand ActiveX-Objekte #####

'++++ Variablen
' Variablen aus JScript sind in VBScript referenzierbar !

Dim AnalogUhr_Zeit
Dim AnalogUhr_Zeit_Minute
Dim AnalogUhr_Zeit_Sekunde
Dim AnalogUhr_Zeit_Stunde

```



```

'++++ aktuelle Zeit holen
Sub ZeitHolen
'aktuelle Zeit holen
AnalogUhr_Zeit = time
AnalogUhr_Zeit_Sekunde = second(AnalogUhr_Zeit)
AnalogUhr_Zeit_Minute = minute(AnalogUhr_Zeit)
AnalogUhr_Zeit_Stunde = hour(AnalogUhr_Zeit)

'prüfen ob aktuelle AnalogUhr_Zeit_Stunde > 12, wenn ja dann auf 12-Stundenbasis dezimieren
if AnalogUhr_Zeit_Stunde > 12 then
    AnalogUhr_Zeit_Stunde = AnalogUhr_Zeit_Stunde - 12
end if
end Sub

'++++ Zeiger bewegen
Sub ZeigerBewegen
'aktuelle Zeit holen
ZeitHolen

'Sekundenzeiger bewegen, falls Sekundenzeiger erzeugt wurde
if AnalogUhr_Zeiger_Sekunden_Erzeugen then
    Call ID_AnalogUhr_Zeiger_Sekunden.Rotate(0,0,6)
end if

'danach den Minutenzeiger
if AnalogUhr_Zeit_Sekunde = 0 then
    Call ID_AnalogUhr_Zeiger_Minuten.Rotate(0,0,6)

'danach den Stundenzeiger
if AnalogUhr_Zeit_Minute mod 2 = 0 then
    call ID_AnalogUhr_Zeiger_Stunden.Rotate(0,0,1)
end if
end if
End Sub

'++++ Uhrwerk-Objekt mit der VB-Routine verbinden, also Feder ins Uhrwerk einbauen
'    und Uhrwerk laufen lassen, also Zeiger bewegen
'    ID_AnalogUhr_UhrWerk ist Objekt
'    OnInit ist Ereignis
'    _ ist Kodierungsvorschrift
Sub ID_AnalogUhr_UhrWerk_OnInit
'Timer einrichten für das Objekt
'    ruft ZeigerBewegen auf
call ID_AnalogUhr_UhrWerk("ID_AnalogUhr_Timer").at(1.000, "ZeigerBewegen", -1,1.000, 1)
End Sub

'++++ Mit Laden des Dokumentes aktivierte Routine: Uhrwerk initialisieren
'    window ist Objekt
'    onload ist Ereignis
'    _ ist Kodierungsvorschrift
Sub window_onload
'aktuelle Zeit holen
ZeitHolen

'Objekte in Erstposition laut aktuelle Zeit setzen: Umrechnung bezogen auf 360 Grad
'    Positionierung per Methode des jeweiligen Objektes
'        AnalogUhr_Zeit_Sekunde*6 ist maximal 60 * 6 also 360 Grad
if AnalogUhr_Zeiger_Sekunden_Erzeugen then
    call ID_AnalogUhr_Zeiger_Sekunden.Rotate(0,0,(AnalogUhr_Zeit_Sekunde*6) - 90)
end if

call ID_AnalogUhr_Zeiger_Minuten.Rotate(0,0,(AnalogUhr_Zeit_Minute*6) - 90)

'        AnalogUhr_Zeit_Stunde*360/12 ist maximal 12*360/12 also 360 Grad
call ID_AnalogUhr_Zeiger_Stunden.Rotate(0,0,(AnalogUhr_Zeit_Stunde*360/12) - 90)

'        AnalogUhr_Zeit_Minute/2 ist maximal 60/2 also 30 Grad
call ID_AnalogUhr_Zeiger_Stunden.Rotate(0,0,int(AnalogUhr_Zeit_Minute/2))

```




```

'und Uhrwerk ticken lassen per Timer starten (.Play)
'    und damit Zeiger bewegen laut mit Timer verbundenen funktion ZeigerBewegen
'        siehe ID_AnalogUhr_UhrWerk_OnInit
call ID_AnalogUhr_UhrWerk("ID_AnalogUhr_Timer").Play
end Sub
</SCRIPT>

<SCRIPT LANGUAGE="JScript">
<!--

##### interne Variablen #####

// ID der ActiveX-Controls
var Objekt_ClassID_Uhrwerk="";
var Objekt_ClassID_UhrLayout="";

// Maximalwert für Veränderung von Breite und Höhe
var Uhr_MaximaleVeraenderung_BreiteUndHoehe=0;

// Zeigerabweichung für alle Zeiger
var ZeigerAbweichung_Horizontal=0;

// Ovalabweichung für Hintergrund
var Hintergrund_OvalVerschiebung_HorizontalUndVertikal=0;

##### Funktionen zur Erzeugung der Uhr #####

// ++++++ Uhrwerk-Objekt ++++++

function UhrWerk_Objekt_Erzeugen()
{
// Das Uhrwerk ist der Motor der Uhr und wird NICHT gerendert
document.write( '<OBJECT ID="ID_AnalogUhr_UhrWerk"'
+ ' CLASSID="' + Objekt_ClassID_Uhrwerk + '"'
+ '>'
+ '</OBJECT>'
);
}

// ++++++ Uhr-Layout-Objekt ++++++

// ----- Uhr-Layout-Objekt: Kopf erzeugen -----

function UhrLayout_ObjektKopf_Erzeugen(IDKette,BreiteAenderung,HoeheAenderung)
// Änderung der Breite bzw. Höhe dient zum optisch-symmetrischen Ausrichten der Uhrteile
//    z.B. Zifferänderung von Römisch (wie XII) auf 12 bewirkt andere Breite, also
//    muss die Lage des Hintergrundes korrigiert werden, damit optische Symmetrie anliegt.
{
var UhrLayout_ObjektKopf_Erzeugen_Wert1=0;
var UhrLayout_ObjektKopf_Erzeugen_Wert2=0;

// Maximalwerte der Veränderung einhalten
if (BreiteAenderung > Uhr_MaximaleVeraenderung_BreiteUndHoehe)
{BreiteAenderung = Uhr_MaximaleVeraenderung_BreiteUndHoehe;}

if (HoeheAenderung > Uhr_MaximaleVeraenderung_BreiteUndHoehe)
{HoeheAenderung = Uhr_MaximaleVeraenderung_BreiteUndHoehe;}

// Veränderung vollziehen
UhrLayout_ObjektKopf_Erzeugen_Wert1=AnalogUhr_BreiteUndHoehe + BreiteAenderung;
UhrLayout_ObjektKopf_Erzeugen_Wert2=AnalogUhr_BreiteUndHoehe + HoeheAenderung;

document.write( '<OBJECT ID="' + IDKette + '"'
+ ' CLASSID="' + Objekt_ClassID_UhrLayout + '"'
+ ' STYLE="position:absolute;'
+ 'width:' + UhrLayout_ObjektKopf_Erzeugen_Wert1 + ';'
+ 'height:' + UhrLayout_ObjektKopf_Erzeugen_Wert2 + ';'
+ 'top:' + AnalogUhr_Top + ';"'

```




```

        +           'left:' + AnalogUhr_Left
        +           ""
        + '>'
    );
}

// ----- Uhr-Layout-Objekt: Hintergrund -----

function UhrLayout_Hintergrund_Erzeugen()
{
    // Oval-Dimension ermitteln
    var Oval_BreiteUndHoehe=AnalogUhr_BreiteUndHoehe-10;

    // Oval-Abweichung horizontal ermitteln
    // Annahme: keine römischen Ziffern
    var OvalAbweichung_Horizontal=ZeigerAbweichung_Horizontal;

    // auf römische Ziffern prüfen
    if (AnalogUhr_Ziffern_Satz == 1)
    {OvalAbweichung_Horizontal = ZeigerAbweichung_Horizontal * 3};

    UhrLayout_ObjektKopf_Erzeugen('ID_AnalogUhr_Hintergrund',
        OvalAbweichung_Horizontal,
        ZeigerAbweichung_Horizontal
    );

    document.write(
        '<PARAM NAME="Line0001" VALUE="SetLineColor('
        + AnalogUhr_HintergrundRahmen_Farbe_RotAnteil + ','
        + AnalogUhr_HintergrundRahmen_Farbe_GruenAnteil + ','
        + AnalogUhr_HintergrundRahmen_Farbe_BlauAnteil
        + ')"'
        + '>'
        + '<PARAM NAME="Line0001" VALUE="SetLineStyle(5)">'
        + '<PARAM NAME="Line0002" VALUE="SetFillStyle(1)">'
        + '<PARAM NAME="Line0003" VALUE="SetFillColor('
        + AnalogUhr_Hintergrund_Farbe_RotAnteil + ','
        + AnalogUhr_Hintergrund_Farbe_GruenAnteil + ','
        + AnalogUhr_Hintergrund_Farbe_BlauAnteil
        + ')"'
        + '>'
        + '<PARAM NAME="Line0004" VALUE="SetGradientFill(0,0,'
        + Hintergrund_OvalVerschiebung_HorizontalUndVertikal + ','
        + Hintergrund_OvalVerschiebung_HorizontalUndVertikal + ','
        + '0)"'
        + '>'
        + '<PARAM NAME="Line0005" VALUE="Oval('
        + Hintergrund_OvalVerschiebung_HorizontalUndVertikal + ','
        + // Verschiebung Ovalkante horizontal bezüglich Zeigerursprung
        + // < 0 nach links, > 0 nach rechts
        + Hintergrund_OvalVerschiebung_HorizontalUndVertikal + ','
        + // Verschiebung vertikal bezüglich Zeigerursprung
        + // < 0 nach oben, > 0 nach unten
        + Oval_BreiteUndHoehe + ',' // Breite des Ovals
        + Oval_BreiteUndHoehe + ',' // Höhe des Ovals
        + '0)"'
        + '>'
        + '</OBJECT>'
    );
}

// ----- Uhr-Layout-Objekt: Ziffern -----

function UhrLayout_Ziffern_Erzeugen_ParameterErzeugen(ParameterNummerAlsString,Ziffer,Breite,Hoehe)

```



```

{
document.write( '<PARAM NAME="Line00' + ParameterNummerAlsString + '"
                + ' VALUE="Text(' + Ziffer + ',' + Breite + ',' + Hoehe + ',' + '0' + ')"'
                + '>'
                );
}

function UhrLayout_Ziffern_Erzeugen()
{
// +++++ Schriftart festlegen
var Ziffer_SchriftArt="times new roman"; // nicht ändern !!!

// +++++ Schrifthöhe berechnen
var Ziffer_SchriftHoehe=Math.floor(AnalogUhr_BreiteUndHoehe / 10); // nicht ändern !!

// Fett-Darstellung der Ziffer
// Annahme: nicht fett
var ZifferDicke=65; // nicht ändern
if (AnalogUhr_Ziffer_Fett)
{ZifferDicke*=10;} // 650 für fett, nicht ändern

// +++++ Ziffernart ermitteln

// ----- Annahme: frei definierte Ziffern verwenden
var Ziffer_Standard      =AnalogUhr_Ziffer_Standard;
var Ziffer_Stunde_Viertel =AnalogUhr_Ziffer_Stunde_Viertel;
var Ziffer_Stunde_Halbe   =AnalogUhr_Ziffer_Stunde_Halbe;
var Ziffer_Stunde_DreiViertel=AnalogUhr_Ziffer_Stunde_DreiViertel;
var Ziffer_Stunde_Voll     =AnalogUhr_Ziffer_Stunde_Voll;

// ----- auf Art prüfen
if (AnalogUhr_Ziffern_Satz > 0)
{
// prüfen auf römisch (Satz 1)
if (AnalogUhr_Ziffern_Satz==1)
{
// nicht ändern
Ziffer_Standard      ='.';
Ziffer_Stunde_Viertel ='III';
Ziffer_Stunde_Halbe   ='VI';
Ziffer_Stunde_DreiViertel='IX';
Ziffer_Stunde_Voll     ='XII';
}
else
{
// normal (Satz 2), nicht ändern
Ziffer_Standard      ='.';
Ziffer_Stunde_Viertel ='3';
Ziffer_Stunde_Halbe   ='6';
Ziffer_Stunde_DreiViertel='9';
Ziffer_Stunde_Voll     ='12';
}
}

// ----- Ziffern in das Format für Objekt-Erzeugung umwandeln
// Text-Funktion akzeptiert nur Strings innerhalb "" und nicht innerhalb "
Ziffer_Standard      ="" + Ziffer_Standard      + "";
Ziffer_Stunde_Viertel ="" + Ziffer_Stunde_Viertel + "";
Ziffer_Stunde_Halbe   ="" + Ziffer_Stunde_Halbe   + "";
Ziffer_Stunde_DreiViertel="" + Ziffer_Stunde_DreiViertel + "";
Ziffer_Stunde_Voll     ="" + Ziffer_Stunde_Voll     + "";

// +++++ Layout der Zifferndarstellung ermitteln
// ist abhängig von der Breite und Höhe der Analoguhr
// fixiert für Schriftart laut Ziffer_SchriftArt
// darf im Programmcode nicht geändert werden

// ----- Umrechnungsfaktor ermitteln
var Ziffer_Umrechnungsfaktor=AnalogUhr_BreiteUndHoehe / 210; // nicht ändern !!

```



```
// ----- Ziffern für Viertel- und Dreiviertelstunde

// - - - Layoutverschiebung vertikal bei Umrechnungsfaktor 1/1
var Hoehe_ViertelUndDreiviertelStunde = Math.floor(Ziffer_SchriftHoehe / 4); // nicht ändern !!

// Layoutverschiebung vertikal bei realem Umrechnungsfaktor
Hoehe_ViertelUndDreiviertelStunde=Math.floor(Hoehe_ViertelUndDreiviertelStunde * Ziffer_Umrechnungsfaktor);

// - - - Layoutverschiebung horizontal bei Umrechnungsfaktor 1/1
// Viertel : Abstand von Ziffer zum Zeigerursprung enthält NICHT Breite der Ziffer
// Dreiviertel: Abstand von Ziffer zum Zeigerursprung enthält Breite der Ziffer
// Problem: Breite der Ziffer nicht ermittelbar !
var Breite_ViertelStunde = 75; // nicht ändern
var Breite_DreiViertelStunde = -90; // nicht ändern

// Layoutverschiebung horizontal bei realem Umrechnungsfaktor
Breite_ViertelStunde =Math.floor(Breite_ViertelStunde * Ziffer_Umrechnungsfaktor);
Breite_DreiViertelStunde=Math.floor(Breite_DreiViertelStunde * Ziffer_Umrechnungsfaktor);

var Breite_Durchschnitt_ViertelUndDreiViertelStunde= Math.abs(Breite_ViertelStunde)
+ Math.abs(Breite_DreiViertelStunde);
Breite_Durchschnitt_ViertelUndDreiViertelStunde=Math.floor(Breite_Durchschnitt_ViertelUndDreiViertelStunde / 2);
var Breite_Drittel_ViertelUndDreiViertelStunde=Math.floor(Breite_Durchschnitt_ViertelUndDreiViertelStunde / 3);
var Breite_Fuenftel_ViertelUndDreiViertelStunde=Math.floor(Breite_Durchschnitt_ViertelUndDreiViertelStunde / 5);

// ----- Ziffern für volle und halbe Stunde

// - - - Layoutverschiebung vertikal bei Umrechnungsfaktor 1/1
var Hoehe_HalbeStunde=Math.floor(Ziffer_SchriftHoehe / 3); // nicht ändern
var Hoehe_VolleStunde=Hoehe_HalbeStunde;
Hoehe_HalbeStunde+=82; // nicht ändern
Hoehe_VolleStunde+=76; // nicht ändern

// Layoutverschiebung vertikal bei realem Umrechnungsfaktor
Hoehe_HalbeStunde=Math.floor(Hoehe_HalbeStunde * Ziffer_Umrechnungsfaktor);
Hoehe_VolleStunde=Math.floor(Hoehe_VolleStunde * Ziffer_Umrechnungsfaktor);
Hoehe_VolleStunde*=-1;

var Hoehe_Durchschnitt_HalbeUndVolleStunde= Math.abs(Hoehe_HalbeStunde)
+ Math.abs(Hoehe_VolleStunde);
Hoehe_Durchschnitt_HalbeUndVolleStunde=Math.floor(Hoehe_Durchschnitt_HalbeUndVolleStunde / 2);
var Hoehe_Drittel_HalbeUndVolleStunde=Math.floor(Hoehe_Durchschnitt_HalbeUndVolleStunde / 3);
var Hoehe_Fuenftel_HalbeUndVolleStunde=Math.floor(Hoehe_Durchschnitt_HalbeUndVolleStunde / 5);
var Hoehe_Siebtel_HalbeUndVolleStunde=Math.floor(Hoehe_Durchschnitt_HalbeUndVolleStunde / 7);

// - - - Layoutverschiebung horizontal bei Umrechnungsfaktor 1/1
// Problem: Breite der Ziffer nicht ermittelbar und damit Zentrierung nicht ermittelbar !
var Breite_HalbeStunde=0;
var Breite_VolleStunde=0;

// auf römische Ziffern prüfen (Satz 1)
if (AnalogUhr_Ziffern_Satz==1)
{
    Breite_HalbeStunde=-6; // nicht ändern
    Breite_VolleStunde=-11; // nicht ändern
}
else
{
    Breite_HalbeStunde=-3; // nicht ändern
    Breite_VolleStunde=-8; // nicht ändern
}

// Layoutverschiebung horizontal bei realem Umrechnungsfaktor
Breite_HalbeStunde=Math.floor(Breite_HalbeStunde * Ziffer_Umrechnungsfaktor);
Breite_VolleStunde=Math.floor(Breite_VolleStunde * Ziffer_Umrechnungsfaktor);

// ----- Ziffern zwischen viertel, halbe, dreiviertel und volle Stunde
```



```

//      links < 0 | rechts > 0
//      12
// oben < 0 h | a oben < 0
//      g | b
// --- 9 -----+----- 3 ---
//      f | c
// unten > 0 e | d unten > 0
//      6
// links < 0 | rechts > 0
//      |

// Spiegelsymmetrie vertikal (Höhe)
// h a
// g b
// f c
// e d

// Spiegelsymmetrie horizontal (Breite)
// a d
// e h
// b c
// f g

// - - - Layoutverschiebung horizontal
var a_d_Breite = Breite_Drittel_ViertelUndDreiViertelStunde
                + Breite_Fuenftel_ViertelUndDreiViertelStunde; // rechts
var h_e_Breite = -1 * a_d_Breite; // links
var b_c_Breite = Breite_Drittel_ViertelUndDreiViertelStunde * 2;
b_c_Breite += Breite_Fuenftel_ViertelUndDreiViertelStunde; // rechts
var g_f_Breite = -1 * b_c_Breite; // links

// - - - Layoutverschiebung vertikal
var a_h_Hoehe = Hoehe_Drittel_HalbeUndVolleStunde * 2;
a_h_Hoehe += Hoehe_Fuenftel_HalbeUndVolleStunde;
a_h_Hoehe *= -1; // oben
var d_e_Hoehe = -1 * a_h_Hoehe; // unten
var b_g_Hoehe = Hoehe_Drittel_HalbeUndVolleStunde
                + Hoehe_Siebtel_HalbeUndVolleStunde;
b_g_Hoehe *= -1; // oben
var c_f_Hoehe = -1 * b_g_Hoehe; // unten

// +++++ Layout der Zifferndarstellung erzeugen

// ----- Objektkopf erzeugen
UhrLayout_ObjektKopf_Erzeugen('ID_AnalogUhr_Ziffern',0,0);

// ----- Font als Parameter des Objektes erzeugen
document.write(
    '<PARAM NAME="Line0001" VALUE="SetLineColor(0, 0, 0)">'
    + '<PARAM NAME="Line0002" VALUE="SetLineStyle(0)">'
    + '<PARAM NAME="Line0003" VALUE="SetFillStyle(1)">'
    + '<PARAM NAME="Line0004" VALUE="SetFillColor('
    + AnalogUhr_Ziffer_Farbe_RotAnteil + ','
    + AnalogUhr_Ziffer_Farbe_GruenAnteil + ','
    + AnalogUhr_Ziffer_Farbe_BlauAnteil
    + ') '
    + ' '
    + '>'
    + '<PARAM NAME="Line0005" VALUE="SetFont('
    + Ziffer_SchriftArt + ','
    + Ziffer_SchriftHoehe + ','
    + ZifferDicke + ','
    + '0,0,0'
    + ') '
    + ' '
    + '>'
    + ');

```



```
// ----- Ziffern als Parameter des Objektes erzeugen
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('06',Ziffer_Standard,a_d_Breite,a_h_Hoehe); // a
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('07',Ziffer_Standard,b_c_Breite,b_g_Hoehe); // b
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('08',Ziffer_Stunde_Viertel,
    Breite_ViertelStunde,Hoehe_ViertelUndDreiviertelStunde);
    // Viertel Stunde
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('09',Ziffer_Standard,b_c_Breite,c_f_Hoehe); // c
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('10',Ziffer_Standard,a_d_Breite,d_e_Hoehe); // d
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('11',Ziffer_Stunde_Halbe,
    Breite_HalbeStunde,Hoehe_HalbeStunde); // Halbe Stunde
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('12',Ziffer_Standard,h_e_Breite,d_e_Hoehe); // e
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('13',Ziffer_Standard,g_f_Breite,c_f_Hoehe); // f
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('14',Ziffer_Stunde_DreiViertel,
    Breite_DreiViertelStunde,Hoehe_ViertelUndDreiviertelStunde);
    // DreiViertel Stunde
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('15',Ziffer_Standard,g_f_Breite,b_g_Hoehe); // g
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('16',Ziffer_Standard,h_e_Breite,a_h_Hoehe); // h
UhrLayout_Ziffern_Erzeugen_ParameterErzeugen('17',Ziffer_Stunde_Voll,
    Breite_VolleStunde,Hoehe_VolleStunde); // Volle Stunde

document.write( '</OBJECT>');
}

// ----- Uhr-Layout-Objekt: Zeiger -----

function UhrLayout_Zeiger_Erzeugen(IDKette,
    RotAnteil,GruenAnteil,BlauAnteil,
    Wert1,Wert2,Wert3,Wert4
)
{
    // +++++ Zeigerform
    // Annahme: Zeiger hohl
    var ZeigerForm=2;
    if (AnalogUhr_Zeiger_Flaeche)
    {ZeigerForm--;} // 1 für Fläche

    // +++++ Layout erzeugen

    // ----- Kopf des Objektes
    UhrLayout_ObjektKopf_Erzeugen(IDKette,ZeigerAbweichung_Horizontal,0);

    // ----- Zeigerfarben
    document.write( '<PARAM NAME="Line0001"'
        + ' VALUE="SetLineColor(' + RotAnteil + ',' + GruenAnteil + ',' + BlauAnteil + ')"'
        + '>'
    );

    document.write( '<PARAM NAME="Line0002" VALUE="SetLineStyle(1)">');

    document.write( '<PARAM NAME="Line0003"'
        + ' VALUE="SetFillColor(' + RotAnteil + ',' + GruenAnteil + ',' + BlauAnteil + ')"'
        + '>'
    );

    // ----- Zeigerform
    document.write( '<PARAM NAME="Line0004" VALUE="SetFillStyle(' + ZeigerForm + ')">');

    // ----- Zeigerposition im Layout
    document.write( '<PARAM NAME="Line0005"'
        + ' VALUE="Rect(' + Wert1 + ',' + Wert2 + ',' + Wert3 + ',' + Wert4 + ')"'
        + '>'
        + '</OBJECT>'
    );

    // Wert1 Abweichung vom Zeigerursprung vertikal, 0 keine Abweichung, > 0 nach oben, < 0 nach unten
    // Wert2 Abweichung vom Zeigerursprung horizontal, 0 keine Abweichung, > 0 nach links, < 0 nach rechts
    // Wert3 Zeigerlänge

```



```

// Wert4 Zeigerdicke
}

function UhrLayout_AlleZeiger_Erzeugen()
// Hinweis: Alle Zeiger sollten sich im Ursprung treffen: Beachte Dimension der Uhr !
// Wenn sich alle Zeiger treffen, dann nach Augenmass alle Zeiger
// in der Uhrfläche zentrieren.
// Wenn Zeiger zentriert wurden, dann Zeigerursprung als Ring darstellen und
// den Ursprung zentrieren.
{
// Variablen der Verschiebung
// vertikal: negativer Wert für Verschiebung nach unten (n unten)
// positiver Wert für Verschiebung nach oben (p oben)

// horizontal: negativer Wert für Verschiebung nach links (n links)
// positiver Wert für Verschiebung nach rechts (p rechts)
var Vertikale_Verschiebung=0;
var Horizontale_Verschiebung=0;

// +++++ Sekundenzeiger

// ----- Länge
// dient als Basis für die Ermittlung der Längen vom Minuten- und Stundenzeiger
var ZeigerLaenge=Math.floor(AnalogUhr_BreiteUndHoehe/2)-15;
var ZeigerLaenge_Differenz=Math.floor(ZeigerLaenge / 6);

// ----- prüfen ob Sekundenzeiger erzeugt werden soll
// Sekundenzeiger
if (AnalogUhr_Zeiger_Sekunden_Erzeugen)
{
// Sekundenzeiger soll erzeugt werden

// ----- Verschiebung
// auf römische Ziffern (Satz 1) prüfen
if (AnalogUhr_Ziffern_Satz==1)
{
Vertikale_Verschiebung=1; // nicht verändern
Horizontale_Verschiebung=0; // nicht verändern
}
else
{
Vertikale_Verschiebung=-2; // nicht verändern
Horizontale_Verschiebung=0; // nicht verändern
}

// ----- Zeiger erzeugen
UhrLayout_Zeiger_Erzeugen('ID_AnalogUhr_Zeiger_Sekunden',
                          AnalogUhr_Zeiger_Sekunden_Farbe_RotAnteil,
                          AnalogUhr_Zeiger_Sekunden_Farbe_GruenAnteil,
                          AnalogUhr_Zeiger_Sekunden_Farbe_BlauAnteil,
                          Vertikale_Verschiebung,Horizontale_Verschiebung,
                          ZeigerLaenge,0 // Zeigerdicke ist 0 für Standarddicke
                          );
}

// +++++ Minutenzeiger

// ----- Zeigerlänge
ZeigerLaenge=ZeigerLaenge_Differenz;

// ----- Verschiebung
// auf römische Ziffern (Satz 1) prüfen
if (AnalogUhr_Ziffern_Satz==1)
{
Vertikale_Verschiebung=1; // nicht verändern
Horizontale_Verschiebung=1; // nicht verändern
}
else

```



```

{
    Vertikale_Verschiebung=3; // nicht verändern
    Horizontale_Verschiebung=0; // nicht verändern
}

// ----- Zeiger erzeugen
UhrLayout_Zeiger_Erzeugen('ID_AnalogUhr_Zeiger_Minuten',
    AnalogUhr_Zeiger_Minuten_Farbe_RotAnteil,
    AnalogUhr_Zeiger_Minuten_Farbe_GruenAnteil,
    AnalogUhr_Zeiger_Minuten_Farbe_BlauAnteil,
    Vertikale_Verschiebung,Horizontale_Verschiebung,
    ZeigerLaenge,1 // Zeigerdicke ist 1
);

// +++++ Stundenzeiger

// ----- Zeigerlänge
ZeigerLaenge=ZeigerLaenge_Differenz;

// ----- Verschiebung
// auf römische Ziffern (Satz 1) prüfen
if (AnalogUhr_Ziffern_Satz==1)
{
    Vertikale_Verschiebung=0; // nicht verändern
    Horizontale_Verschiebung=-1; // nicht verändern
}
else
{
    Vertikale_Verschiebung=1; // nicht verändern
    Horizontale_Verschiebung=-1; // nicht verändern
}

// ----- Zeiger erzeugen
UhrLayout_Zeiger_Erzeugen('ID_AnalogUhr_Zeiger_Stunden',
    AnalogUhr_Zeiger_Stunden_Farbe_RotAnteil,
    AnalogUhr_Zeiger_Stunden_Farbe_GruenAnteil,
    AnalogUhr_Zeiger_Stunden_Farbe_BlauAnteil,
    Vertikale_Verschiebung,Horizontale_Verschiebung,
    ZeigerLaenge,2 // Zeigerdicke ist 2
);
}

// ----- Uhr-Layout-Objekt: Zeigerursprung -----
function UhrLayout_ZeigerUsprung_Erzeugen()
// Hinweis: Alle Zeiger sollten sich im Ursprung treffen: Beachte Dimension der Uhr !
// Wenn sich alle Zeiger treffen, dann nach Augenmass alle Zeiger
// in der Uhrfläche zentrieren.
// Wenn Zeiger zentriert wurden, dann Zeigerursprung als Ring darstellen und
// den Ursprung zentrieren.
{
    // +++++ Form des Ursprunges

    var ZeigerUsprung_Flaeche=true; // false für Ringform
    // Ringform geeignet zum Testen der Lage der Zeiger
    // im Zeigerursprung (Zentrierung)

    // Annahme: Ursprung als Ring
    var UrsprungForm=2;
    if (ZeigerUsprung_Flaeche)
    {UrsprungForm--;} // 1 für Fläche

    // +++++ Dimension des Ursprunges

    var Oval_BreiteUndHoehe=Math.floor(AnalogUhr_BreiteUndHoehe/12); // nicht verändern !!

    // ----- Veränderung der Breite des Ursprunges
    var BreiteVeraeänderung = Math.abs(AnalogUhr_BreiteUndHoehe - 210);
    BreiteVeraeänderung= Math.floor(BreiteVeraeänderung * 0.02);
    BreiteVeraeänderung*=-1;

```



```

// ----- Veränderung der Höhe des Ursprunges
var HoeheVeraenderung = Math.abs(AnalogUhr_BreiteUndHoehe - 210);
HoeheVeraenderung= Math.floor(HoeheVeraenderung * 0.08);
HoeheVeraenderung+=1;
HoeheVeraenderung*=-1;

// +++++ Verschiebung des Ursprunges

var Oval_Verschiebung_Horizontal=0;
var Oval_Verschiebung_Horizontal_Nenner=0;
var Oval_Verschiebung_Vertikal=0;
var Oval_Verschiebung_Vertikal_Nenner=0;

// ----- Verschiebung horizontal
//   auf römisch prüfen
if (AnalogUhr_Ziffern_Satz==1)
{Oval_Verschiebung_Horizontal_Nenner=45;} // > 0 !!!
else
{Oval_Verschiebung_Horizontal_Nenner=25;} // > 0 !!!
Oval_Verschiebung_Horizontal=Math.floor(AnalogUhr_BreiteUndHoehe/Oval_Verschiebung_Horizontal_Nenner);
Oval_Verschiebung_Horizontal*=-1; // Verschiebung nach links

// ----- Verschiebung vertikal
//   auf römisch prüfen
if (AnalogUhr_Ziffern_Satz==1)
{Oval_Verschiebung_Vertikal_Nenner=40;} // > 0 !!!
else
{Oval_Verschiebung_Vertikal_Nenner=30;} // > 0 !!!
Oval_Verschiebung_Vertikal=Math.floor(AnalogUhr_BreiteUndHoehe/Oval_Verschiebung_Vertikal_Nenner);
Oval_Verschiebung_Vertikal*=-1; // Verschiebung nach oben

// +++++ Objekt erzeugen

UhrLayout_ObjektKopf_Erzeugen('ID_AnalogUhr_Zeiger_Ursprung',BreiteVeraenderung,HoeheVeraenderung);

document.write(
    '<PARAM NAME="Line0001" VALUE="SetLineColor('
    + AnalogUhr_Zeiger_Ursprung_Farbe_RotAnteil + ','
    + AnalogUhr_Zeiger_Ursprung_Farbe_GruenAnteil + ','
    + AnalogUhr_Zeiger_Ursprung_Farbe_BlauAnteil
    + ') '
    + ' '
    + '>'
    + '<PARAM NAME="Line0001" VALUE="SetLineStyle(5)">'
    + '<PARAM NAME="Line0002" VALUE="SetFillStyle(' + UrsprungForm + ')">'
    + '<PARAM NAME="Line0003" VALUE="SetFillColor('
    + AnalogUhr_Zeiger_Ursprung_Farbe_RotAnteil + ','
    + AnalogUhr_Zeiger_Ursprung_Farbe_GruenAnteil + ','
    + AnalogUhr_Zeiger_Ursprung_Farbe_BlauAnteil
    + ') '
    + ' '
    + '>'
    + '<PARAM NAME="Line0004" VALUE="SetGradientFill(0,0,'
    + Hintergrund_OvalVerschiebung_HorizontalUndVertikal + ','
    + Hintergrund_OvalVerschiebung_HorizontalUndVertikal + ','
    + '0)'
    + ' '
    + '>'
    + '<PARAM NAME="Line0005" VALUE="Oval('
    + Oval_Verschiebung_Horizontal + ','
    + // Verschiebung Ovale horizontal bezüglich Zeigerursprung
    + // < 0 nach links, > 0 nach rechts
    + Oval_Verschiebung_Vertikal + ','
    + // Verschiebung vertikal bezüglich Zeigerursprung
    + // < 0 nach oben, > 0 nach unten
    + Oval_BreiteUndHoehe + ',' // Breite des Ovals
    + Oval_BreiteUndHoehe + ',' // Höhe des Ovals
    + '0)'

```




```

+
+ '>'
+ '</OBJECT>'
);
}

// ##### Initialisierung der Uhr #####

function AnalogUhr_Init()
{
    var Div_BreiteUndHoehe=0;
    var Div_Left=0;

    // +++++ Maximale Höhe und Breite der Uhr prüfen, nicht ändern
    if (AnalogUhr_BreiteUndHoehe < 140)
    {AnalogUhr_BreiteUndHoehe=140;}

    if (AnalogUhr_BreiteUndHoehe > 330)
    {AnalogUhr_BreiteUndHoehe=330;}

    // +++++ ID der ActiveX-Controls festlegen, nicht ändern !!
    Objekt_ClassID_Uhrwerk= 'CLSID:B0A6BAE2-AAF0-11d0-A152-00A0C908DB96';
    Objekt_ClassID_UhrLayout='CLSID:369303C2-D7AC-11D0-89D5-00A0C90833E6';

    // +++++ Ovalverschiebung für Hintergrund-Objekt berechnen
    Hintergrund_OvalVerschiebung_HorizontalUndVertikal=Math.floor(AnalogUhr_BreiteUndHoehe/2); // nicht ändern
    Hintergrund_OvalVerschiebung_HorizontalUndVertikal*=-1; // nach links bzw. oben nicht ändern

    // +++++ Zeigerabweichung horizontal berechnen
    ZeigerAbweichung_Horizontal=Math.floor(AnalogUhr_BreiteUndHoehe / 42); // nicht ändern

    // +++++ Maximale Veränderung von Breite und Höhe definieren (siehe UhrLayout_ObjektKopf_Erzeugen() )
    Uhr_MaximaleVeränderung_BreiteUndHoehe=Math.floor(AnalogUhr_BreiteUndHoehe / 10); // nicht ändern

    // +++++ Div, der dies gesamte Analoguhr umspannt
    //      Vorteil: Nur den Container verschieben und schon wandert die Uhr mitmit
    //      Die Uhr wird aus Objekten zusammengebaut, die in diesem Div liegen.

    // ----- Maximale Breite und Höhe
    Div_BreiteUndHoehe=AnalogUhr_BreiteUndHoehe + Uhr_MaximaleVeränderung_BreiteUndHoehe;

    // ----- Div-Left-Änderung ermitteln
    Div_Left=Math.floor(AnalogUhr_BreiteUndHoehe / 20); // nicht ändern
    Div_Left=AnalogUhr_Left - Div_Left;
    Div_Left+=3; // nicht ändern

    // ----- DIV-Kopf erzeugen
    document.write( '<DIV ID="ID_Div_AnalogUhr"'
+      ' STYLE="position:absolute;'
+      'width:' + Div_BreiteUndHoehe + ';'
+      'height:' + Div_BreiteUndHoehe + ';'
+      'top:' + AnalogUhr_Top + ';'
+      'left:' + Div_Left
+      '""
+ '>'
);

    // +++++ Das Uhrwerk erzeugen: Es ist der Motor der Uhr und wird NICHT gerendert
    UhrWerk_Objekt_Erzeugen();

    // +++++ Analoguhr aus Objekten zusammenbauen
    //      Reihenfolge der Objekte entspricht der Überlagerung im Layout

    // ----- Objekt des Hintergrundes der Analoguhr (nicht der Digitaluhr)
    //      Wird der Hintergrund nicht erzeugt, so hat die Uhr das Ziffernblatt in der Farbe des normalen Hintergrundes
    UhrLayout_Hintergrund_Erzeugen();

    // ----- Objekt des Ziffernblattes erzeugen bei Bedarf
    if (AnalogUhr_Ziffern_Erzeugen)

```



```

{UhrLayout_Ziffern_Erzeugen();}

// ----- Objekte der Zeiger
UhrLayout_AlleZeiger_Erzeugen();

// ----- Objekt des Zeigerursprunges
UhrLayout_ZeigerUsprung_Erzeugen();

// +++++ DIV-Ende
document.write( '</DIV>');
}
// -->
</SCRIPT>
</HEAD>

<BODY BGCOLOR="#3A6EA5">
<SCRIPT LANGUAGE="JScript">
<!--
// Es muss BODY angelegt sein: onload im BODY kommt zum Zeitpunkt, an dem der BODY noch nicht
// erzeugt wurde weil <BODY .....> noch nicht komplett geparkt wurde
//      Da init() auch document.write erzeugt, wird ein neuer weisser BODY erzeugt, der aber den
//      HEAD des alten Dokumentes nicht kennt und somit die Uhr zwar gerendert wird,
//      aber mangels der VBScripte NICHT startet.
AnalogUhr_Init();
// -->
</SCRIPT>
</BODY>
</HTML>

```

24.2. Video und Sound abspielen per Mediaplayer

Beispiele für Active-X-Control des Mediaplayers zum Abspielen eines Videos oder Sounds:

Achtung: CLASSID wird inzwischen nicht mehr sicher unterstützt (siehe unten Microsoft ändert die Active-X-Eigenschaften

```

<BODY>
<OBJECT ID="ActiveMovie1"
      WIDTH=356
      HEIGHT=328
      CLASSID="CLSID:05589FA1-C356-11CE-BF01-00AA0055595A"
>
      <PARAM NAME="MovieWindowWidth" VALUE="352">
      <PARAM NAME="MovieWindowHeight" VALUE="247">
      <PARAM NAME="FileName" VALUE="0.wmv">
</OBJECT>

```

Das Active-X-Control umfasst folgende PARAM

```

<PARAM NAME="Appearance" VALUE="0">
<PARAM NAME="AutoStart" VALUE="0">
<PARAM NAME="AllowChangeDisplayMode" VALUE="-1">
<PARAM NAME="AllowHideDisplay" VALUE="0">
<PARAM NAME="AllowHideControls" VALUE="-1">
<PARAM NAME="AutoRewind" VALUE="-1">
<PARAM NAME="Balance" VALUE="0">
<PARAM NAME="CurrentPosition" VALUE="0">
<PARAM NAME="DisplayBackColor" VALUE="0">
<PARAM NAME="DisplayForeColor" VALUE="16777215">
<PARAM NAME="DisplayMode" VALUE="0">
<PARAM NAME="Enabled" VALUE="-1">
<PARAM NAME="EnableContextMenu" VALUE="-1">
<PARAM NAME="EnablePositionControls" VALUE="-1">
<PARAM NAME="EnableSelectionControls" VALUE="0">
<PARAM NAME="EnableTracker" VALUE="-1">
<PARAM NAME="Filename" VALUE="">
<PARAM NAME="FullScreenMode" VALUE="0">
<PARAM NAME="MovieWindowSize" VALUE="0">
<PARAM NAME="PlayCount" VALUE="1">
<PARAM NAME="Rate" VALUE="1">
<PARAM NAME="SelectionStart" VALUE="-1">

```



```

<PARAM NAME="SelectionEnd" VALUE="-1">
<PARAM NAME="ShowControls" VALUE="-1">
<PARAM NAME="ShowDisplay" VALUE="-1">
<PARAM NAME="ShowPositionControls" VALUE="0">
<PARAM NAME="ShowTracker" VALUE="-1">
<PARAM NAME="Volume" VALUE="-600">

```

VALUE-Angaben sind nur Beispiele

Bezeichner der Eigenschaften für JScript sind IDENTISCH mit den Werten aus NAME-Attribut von PARAM !
Gross-Kleinschreibung beachten !

AllowChangeDisplayMode	Änderung des Anzeigemodus Zeitanzeige bzw. Framesanzeige durch User boolean true oder false
AllowHideControls	Änderung des Anzeigemodus der Steuerungstasten während Wiedergabe durch User boolean true oder false false (0) für nicht erlauben
AllowHideDisplay	Änderung des Anzeigemodus der Wiedergabe-Anzeige während Wiedergabe durch User boolean true oder false false (0) für nicht erlauben
AutoRewind	Automatisches Rückspulen nach Stop boolean true oder false false (0) für nicht rückspulen
AutoStart	Autostart der Wiedergabe ein oder aus boolean true oder false false (0) für nicht automatische Wiedergabe: siehe Aktionen zum Player
Balance	numerisch 0 für zentriert default -1000 ganz links +1000 ganz rechts
CurrentPosition	unklar
DisplayBackColor	Hintergrundfarbe der Control-Elemente numerischer Farbwert, kein Hexastring 0 Default für Black 16777215 für Weiss
DisplayForeColor	Hintergrundfarbe der Control-Elemente numerischer Farbwert, kein Hexastring 0 Default für Black 16777215 für Weiss
DisplayMode	Anzeigeart der Position 0 für Time 1 für Frames
Enabled	Control-Anzeige ein, aus boolean true oder false false (0) für nicht erlauben
EnableContextMenu	Rechte-Maus-Kontextmenü ein, aus boolean true oder false false (0) für nicht erlauben
EnablePositionControls	Positionierungs-Button im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
EnableSelectionControls	Selektions-Button im Control-Panel ein, aus boolean true oder false false (0) für nicht erlauben
EnableTracker	Tracker-Balken im Control-Panel ein, aus



	boolean true oder false false (0) für nicht erlauben
FileName	Mediendatei mit Pfad oder Url
FullScreenMode	Vollbildschirm ein, aus boolean true oder false false (0) für kein Vollbildmodus
MovieWindowSize	unklar, Grösse des Anzeigefensters
PlayCount	Anzahl der Wiedergaben der Mediendatei ab 1
Rate	Wiedergabefaktor numerisch 1 für 1-fache Geschwindigkeit 10 für 20-fache Geschwindigkeit
SelectionEnd	Bis-Position, bis zu der wiedergegeben werden soll in Sekunden bezüglich Anfang der Mediendatei (0)
SelectionStart	Von-Position, bis zu der wiedergegeben werden soll in Sekunden bezüglich Anfang der Mediendatei (0)
ShowControls	Start Stop buttons and tracker controls sichtbar ein, aus boolean true oder false false (0) für aus
ShowDisplay	Status-Anzeige-Panel sichtbar ein, aus boolean true oder false false (0) für aus
ShowPositionControls	Previous, Rewind, Forward and Next buttons sichtbar ein, aus boolean true oder false false (0) für aus
ShowSelectionControls	Selektions-Button im Control-Panel sichtbar ein, aus boolean true oder false false (0) für aus
ShowTracker	Trackerbar im Controlpanel sichtbar ein, aus: nur lesen boolean true oder false false (0) für aus
uiMode	Sichtbarkeit anstelle STYLE-visible "invisible" für unsichtbar
Volume	Laustärke als Basis dient die aktuelle Windows-Lautstärke integer 0 für maximimale, also aktuelle Windows-Lautstärke (höher geht nicht !!!) -10000 für total still

Den ganzen Mediaplayer unsichtbar machen: per STYLE-Attribut im OBJECT-Tag

document.ID-Bezeichner.readyState (Mediaplayer 6.4, funktioniert aber auch mit höheren Versionen)

.readyState hat folgende Werte

0	The FileName property has not been initialized.
1	Windows Media Player control is asynchronously loading a file.
3	Windows Media Player control loaded a file, and downloaded enough data to play the file, but has not yet received all data.
4	All data has been downloaded.

Methoden:

.IsSoundCardEnabled()



liefert boolean
 true Soundkarte erkannt
 false keine Soundkarte erkannt

.Run(); Erzeugt und startet Instanz des Mediaplayers und ermöglicht danach weitere Aktionen wie .Play etc.

.Mute ohne ()

lesen, schreiben
 true Player volume is muted.
 false (Default) Player volume is not muted.

.Play ohne ()

Wiedergabe starten

.AutoStart ohne ()

true (Default) Clip automatically starts.
 false Disables automatic start.

.Stop ohne ()

.Pause ohne ()

Audio wiedergeben

```
<HEAD>
<SCRIPT>
var SoundStarten_TimeoutID=0;

function SoundStarten()
{
  // auf Soundkarte prüfen
  if (ID_PlayerObjekt.IsSoundCardEnabled())
  {
    // Soundkarte vorhanden
    // Musik geladen und abspielfähig ?
    if (ID_PlayerObjekt.readyState == 4)
    {
      // geladen, also Playerinstanz starten
      ID_PlayerObjekt.Run();
      // Musik wiedergeben
      ID_PlayerObjekt.Play();
      alert('Musik ist gestartet !');
    }
    else{ SoundStarten_TimeoutID=window.setTimeout('SoundStarten()', 100);}
  }
  else{alert('Keine Soundkarte gefunden !');}
}
</SCRIPT>
</HEAD>
<BODY>
<OBJECT ID="ID_PlayerObjekt" CLASSID="CLSID:05589FA1-C356-11CE-BF01-00AA0055595A">
  <PARAM NAME="FileName" VALUE="a0.mp3">
  <PARAM NAME="AutoStart" VALUE="0">      <!-- kein Autoplay -->
</OBJECT>
<SCRIPT>
if (ID_PlayerObjekt != null){SoundStarten();}
</SCRIPT>
</BODY>
</HTML>
```

25. Sonstiges

25.1. Kalender

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JScript">
<!--
```



```

// Kalender aktueller Monat
// nur für IE

// ##### änderbare Variablen #####

// +++++ Position des Kalenders +++++
var Kalender_Top=20;
var Kalender_Left=20;

// +++++ Farben und Schrift-Layout +++++

// Schriften: Es sollten nur Fonts verwendet werden, die auf jedem Windows-PC installiert sind

// ---- Kalenderkopf -----

var Kalender_Kopf_SchriftArt='Times New Roman, Arial';
var Kalender_Kopf_SchriftGroesse= 5;
var Kalender_Kopf_SchriftFarbe='#CCEEFF';
var Kalender_Kopf_HintergrundFarbe='#3A6EA5'; // identisch mit BGColor des BODY
var Kalender_SpaltenKopf_SchriftFarbe='#AAFFFF';

// ---- Tag -----

var Tag_SchriftArt='Times New Roman, Verdana,Arial';
var Tag_SchriftGroesse=4;
var Tag_SchriftFarbe='#CCCCCC';
var Tag_HintergrundFarbe=Kalender_Kopf_HintergrundFarbe;
var Tag_Sonntag_SchriftFarbe='#FFDD00';
var Tag_Heute_SchriftFarbe='#FFFFFF';
var Tag_Heute_HintergrundFarbe='#1A4E85';
var Tag_Heute_Sonntag_SchriftFarbe=Tag_Heute_SchriftFarbe;
var Tag_Heute_Sonntag_HintergrundFarbe=Tag_Heute_HintergrundFarbe;

// +++++ Timer +++++

// Timer für Kalender zur Erkennung von Tages-, Monats- und Jahreswechsel verwenden
var Kalender_Timer_Verwenden=false; // true für verwenden, false für nicht verwenden

// Timer in Millisekunden für Aktualisierung des Kalenders
// nur verwendet wenn Kalender_Timer_Verwenden auf true
// Wert so hoch wie möglich setzen damit durch den Timer wenig Ressourcen genutzt werden
var Kalender_TimerWert=5000;

// +++++ Monatsnamen +++++

var MonatsNamenFeld = new Array
(
    'Januar',
    'Februar',
    'M&uuml;r',
    'April',
    'Mai',
    'Juni',
    'Juli',
    'August',
    'September',
    'Oktober',
    'November',
    'Dezember'
);

// +++++ Tagnamen kurz +++++

var TagKurzNamenFeld = new Array // Woche beginnt mit Montag
(
    'Mo',
    'Di',
    'Mi',

```



```

'Do',
'Fr',
'Sa',
'So'
);

// ##### interne Variablen #####
// alle Variablen möglichst Wert-Typ-gerecht initialisieren (bei Zeiger kein "nil" oder
// "null" verwenden, also Zeiger ohne init).

var TagKurzNamenFeld_Laenge=TagKurzNamenFeld.length;

// Zeit- und Datum-Berechnungen
var Zeit_Jetzt;
var Zeit_Jetzt_Kette="";
var Zeit_Jetzt_Tag_Numerisch=0;
var Zeit_Jetzt_Monat_Numerisch=0;
var Zeit_Jetzt_Jahr_Numerisch=0;
var Zeit_Jetzt_Stunden_Numerisch=0;
var Zeit_Jetzt_Minuten_Numerisch=0;
var Zeit_Jetzt_Sekunden_Numerisch=0;
var Zeit_Vormonat;
var Zeit_VormonatErsterTag;

// Kalender-Anzeige: Globale Variablen, da Anzeige per Timer gesteuert werden kann
// und somit nicht pro Aufruf die Variablen neu erzeugt werden müssen.
var Anzeige_Kalender_TagesZahler=0;
var Anzeige_Kalender_HTMLCode="";
var Anzeige_Kalender_Kette="";
var Anzeige_Kalender_Zahler_TR=0;
var Anzeige_Kalender_Zahler_TD=0;
var Anzeige_Kalender_Tag_Kette1="";
var Anzeige_Kalender_Tag_Kette2="";
var Anzeige_Kalender_Tag_Kette3="";
var Anzeige_Kalender_Sonntag_Kette1="";
var Anzeige_Kalender_Sonntag_Kette2="";
var Anzeige_Kalender_Sonntag_Kette3="";

// ##### Funktionen der Zeit- und Datum-Berechnungen #####

function Zeit_AktuelleAngabenHolen()
{
    Zeit_Jetzt          = new Date();
    Zeit_Jetzt_Tag_Numerisch = Zeit_Jetzt.getDate();
    Zeit_Jetzt_Monat_Numerisch = Zeit_Jetzt.getMonth() + 1;
    Zeit_Jetzt_Jahr_Numerisch = Zeit_Jetzt.getYear();
    Zeit_Jetzt_Stunden_Numerisch = Zeit_Jetzt.getHours();
    Zeit_Jetzt_Minuten_Numerisch = Zeit_Jetzt.getMinutes();
    Zeit_Jetzt_Sekunden_Numerisch = Zeit_Jetzt.getSeconds();
    Zeit_Vormonat        = new Date(Zeit_Jetzt_Jahr_Numerisch,Zeit_Jetzt_Monat_Numerisch-1,1);
    Zeit_VormonatErsterTag = Zeit_Vormonat.getDay();

    if(Zeit_Jetzt_Jahr_Numerisch < 100)
    {Zeit_Jetzt_Jahr_Numerisch+=1900;}

    if(Zeit_Jetzt_Jahr_Numerisch < 100)
    {Zeit_Jetzt_Jahr_Numerisch+=1900;}
}

function Zeit_NumerischNachString_MitVornull(NumerischerWert)
{
    Zeit_Jetzt_Kette=NumerischerWert.toString();

    // auf Vornull prüfen
    if (NumerischerWert < 10)
    {Zeit_Jetzt_Kette='0' + Zeit_Jetzt_Kette;}

    return Zeit_Jetzt_Kette;
}

```



```

// ##### Funktionen des Kalenders #####

// +++++ TR erzeugen +++++

function Anzeige_Kalender_HTMLCode_TR_Erzeugen(
    FarbeBackground,
    BreiteAlsString,
    HoeheAlsString,
    Ausrichtung_Horizontal,
    Ausrichtung_Vertikal,
    SpaltenSpannweiteAlsString,
    SchriftGroesse,
    SchriftFarbe,
    SchriftArt,
    Kette
)

// erweitert Anzeige_Kalender_HTMLCode
{
    // +++++ TR erzeugen: Beginn-Tag
    Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + '<TR>';

    // +++++ TD (Zelle) erzeugen

    // ----- Tag-Beginn
    Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + '<TD>';

    // ----- Attribut Background
    if (FarbeBackground != "")
    {Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + ' BGCOLOR="' + FarbeBackground + '"';}

    // ----- Attribut Breite
    if (BreiteAlsString != "")
    {Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + ' WIDTH=' + BreiteAlsString;}

    // ----- Attribut Höhe
    if (HoeheAlsString != "")
    {Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + ' HEIGHT=' + HoeheAlsString;}

    // ----- Attribut Ausrichtung horizontal
    if (Ausrichtung_Horizontal != "")
    {Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + ' ALIGN=' + Ausrichtung_Horizontal;}

    // ----- Attribut Ausrichtung vertikal
    if (Ausrichtung_Vertikal != "")
    {Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + ' VALIGN=' + Ausrichtung_Vertikal;}

    // ----- Attribut Spannweite
    if (SpaltenSpannweiteAlsString != "")
    {Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + ' COLSPAN=' + SpaltenSpannweiteAlsString;}

    // ----- Tag-Ende
    Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + '>';

    // ----- Inhalt der Zelle als Font mit Text
    Anzeige_Kalender_HTMLCode=
        Anzeige_Kalender_HTMLCode
        + '<FONT SIZE=' + SchriftGroesse
        + ' COLOR=' + SchriftFarbe
        + ' FACE="' + SchriftArt + '"'
        + '>'
        + Kette
        + '</FONT>';

    // ----- Ende-Tag
    Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + '</TD>';

    // +++++ TR erzeugen: Ende-Tag
    Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode + '</TR>';
}

```




```
// +++++ TD erzeugen ++++++

function Anzeige_Kalender_HTMLCode_TD_Erzeugen(
    Inhalt,
    HintergrundFarbe,
    SchriftFarbe,
    SchriftGroesse,
    SchriftArt,
    BorderColor
)

// erweitert Anzeige_Kalender_HTMLCode
// alle Variablen alle global, damit nicht pro Funktionsaufruf die Variablen neu lokal erzeugt werden.
{
    // +++++ Inhalt auf FETT setzen
    Inhalt='<B>' + Inhalt + '</B>';

    // +++++ TD erzeugen
    // Inhalt der TD ist eine Tabelle
    Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode
        + '<TD BGCOLOR=' + HintergrundFarbe + ' BORDER=' + BorderColor + '>'
        + '<TABLE BORDER=0'
        + ' CELLSPACING=0'
        + ' CELLPADDING=5'
        + ' WIDTH=100%'
        + ' HEIGHT=100%'
        + '>'

    Anzeige_Kalender_HTMLCode_TR_Erzeugen(HintergrundFarbe,'30','30','center','middle','',
        SchriftGroesse,SchriftFarbe,SchriftArt,Inhalt);

    Anzeige_Kalender_HTMLCode= Anzeige_Kalender_HTMLCode
        + '</TABLE>'
        + '</TD>';
}

// +++++ Tag (nicht Sonntag) TD erzeugen ++++++

function Anzeige_Kalender_Tag_HTMLCode_TD_Erzeugen(Inhalt,Heute)
// erweitert Anzeige_Kalender_HTMLCode
// Heute true, so Tag = heute
// alle Variablen alle global, damit nicht pro Funktionsaufruf die Variablen neu lokal erzeugt werden.
{
    // Annahme: Tag ist nicht heute
    Anzeige_Kalender_Tag_Kette1=Tag_HintergrundFarbe;
    Anzeige_Kalender_Tag_Kette2=Tag_SchriftFarbe;
    Anzeige_Kalender_Tag_Kette3=Kalender_Kopf_HintergrundFarbe;

    // prüfen auf Tag == heute
    if (Heute)
    {
        Anzeige_Kalender_Tag_Kette1=Tag_Heute_HintergrundFarbe;
        Anzeige_Kalender_Tag_Kette2=Tag_Heute_SchriftFarbe;
        Anzeige_Kalender_Tag_Kette3=Tag_SchriftFarbe;
    }

    Anzeige_Kalender_HTMLCode_TD_Erzeugen(
        Inhalt,
        Anzeige_Kalender_Tag_Kette1,
        Anzeige_Kalender_Tag_Kette2,
        Tag_SchriftGroesse,
        Tag_SchriftArt,
        Anzeige_Kalender_Tag_Kette3
    );
}

// +++++ Sonntag TD erzeugen ++++++

function Anzeige_Kalender_SonnTag_HTMLCode_TD_Erzeugen(Inhalt,Sonntag)
```



```

// erweitert Anzeige_Kalender_HTMLCode
// Sonntag true, so Tag = Sonntag
// alle Variablen alle global, damit nicht pro Funktionsaufruf die Variablen neu lokal erzeugt werden.
{
// Annahme: Tag ist nicht Sonntag
Anzeige_Kalender_Sonntag_Kette1=Tag_HintergrundFarbe;
Anzeige_Kalender_Sonntag_Kette2=Tag_Sonntag_SchriftFarbe;
Anzeige_Kalender_Sonntag_Kette3=Kalender_Kopf_HintergrundFarbe;

// prüfen auf Tag == Sonntag
if (Sonntag)
{
Anzeige_Kalender_Sonntag_Kette1=Tag_Heute_Sonntag_HintergrundFarbe;
Anzeige_Kalender_Sonntag_Kette2=Tag_Heute_Sonntag_SchriftFarbe;
Anzeige_Kalender_Sonntag_Kette3=Tag_Sonntag_SchriftFarbe;
}

Anzeige_Kalender_HTMLCode_TD_Erzeugen(
    Inhalt,
    Anzeige_Kalender_Sonntag_Kette1,
    Anzeige_Kalender_Sonntag_Kette2,
    Tag_SchriftGroesse,
    Tag_SchriftArt,
    Anzeige_Kalender_Sonntag_Kette3
);
}

// +++++ Kalender erzeugen und anzeigen ++++++

function Anzeige_Kalender()
// periodischer Aufruf per Timer NUR zur Erkennung des Tages- oder Monats- oder Jahreswechsel, falls
// genau zu diesen Zeitpunkten der Kalender aktiv ist.
// Es spart Ressourcen, wenn Timer nicht benutzt wird !!
// Kalender_Timer_Verwenden auf false für Timer nicht verwenden
//      auf true für Timer verwenden
//
// alle Variablen alle global, damit nicht pro Funktionsaufruf die Variablen neu lokal erzeugt werden.
{
// +++++ aktuelle Zeitangaben holen
Zeit_AktuelleAngabenHolen(); // füllt
    // Zeit_Jetzt
    // Zeit_Jetzt_Tag_Numerisch
    // Zeit_Jetzt_Monat_Numerisch
    // Zeit_Jetzt_Jahr_Numerisch
    // Zeit_Jetzt_Stunden_Numerisch
    // Zeit_Jetzt_Minuten_Numerisch
    // Zeit_Jetzt_Sekunden_Numerisch
    // Zeit_Vormonat
    // Zeit_VormonatErsterTag

// +++++ Start- und Endwert der Tage für Erzeugung des Kalenders holen

// ----- Erster Tag des Vormonats als Startwert
var Start = Zeit_VormonatErsterTag;
if(Start > 0)
{Start--;} // minus 1
else
{Start = 6;} //

// ----- Endwert für Monatstage ermitteln
// Annahme: Monat hat 31 Tage
var Stop = 31;

// für alle Monate mit 30 Tagen korrigieren
if( (Zeit_Jetzt_Monat_Numerisch==4)
    || (Zeit_Jetzt_Monat_Numerisch==6)
    || (Zeit_Jetzt_Monat_Numerisch==9)
    || (Zeit_Jetzt_Monat_Numerisch==11)
)

```



```

{--Stop;}

// für Februar korrigieren: 29 oder 28 Tage, also Schaltjahr
if(Zeit_Jetzt_Monat_Numerisch==2)
{
    // Februar
    // Annahme: Kein Schaltjahr, also 28 Tage
    Stop=28;

    // Schaltjahr mit 29 Tagen ermitteln
    if ((Zeit_Jetzt_Jahr_Numerisch%4) == 0)
    {Stop++;} // 29

    if ( (Zeit_Jetzt_Jahr_Numerisch%100) == 0)
    {Stop--;} // wieder 28

    if ((Zeit_Jetzt_Jahr_Numerisch%400) ==0 )
    {Stop++;} // 29
}

// +++++ HTML-Code des Kalenders als Tabelle erzeugen

// +++++ äusserste Tabelle also die des Kalenders insgesamt
Anzeige_Kalender_HTMLCode=
    '<TABLE ID="TABLE_Kalender"'
    +      ' BORDER=0'
    +      ' CELLPADDING=1'
    +      ' CELLSPACING=0'
    +      ' STYLE="position:absolute;'
    +          'left:' + Kalender_Left + ','
    +          'top:' + Kalender_Top
    +      '""'
    + '>';

// +++++ 1. TR des Kalenders als Anzeige aktueller Monat und aktuelles Jahr in Fettschrift

// ----- Text der 1. TR ermitteln: aktueller Monat und aktuelles Jahr in Fettschrift
Anzeige_Kalender_Kette='<B>';
Anzeige_Kalender_Kette=MonatsNamenFeld[Zeit_Jetzt_Monat_Numerisch-1]; // Monatsname als String
Anzeige_Kalender_Kette+=' ';
Anzeige_Kalender_Kette+=Zeit_Jetzt_Jahr_Numerisch.toString(); // Jahr zu String
Anzeige_Kalender_Kette+='</B>';

// ----- Abstand zum Nachfolger
Anzeige_Kalender_Kette+='<BR>';
Anzeige_Kalender_Kette+='<BR>';

// ----- und erzeugen
Anzeige_Kalender_HTMLCode_TR_Erzeugen(
    Kalender_Kopf_HintergrundFarbe, ",", 'center', 'middle', '7',
    Kalender_Kopf_SchriftGroesse,
    Kalender_Kopf_SchriftFarbe,
    Kalender_Kopf_SchriftArt,
    Anzeige_Kalender_Kette
);

// +++++ 2. TR Spaltenkopf der Monatstage; pro Spalte ein TD
Anzeige_Kalender_HTMLCode+='<TR>';

// ----- Tage der Woche abklappern: Woche beginnt mit Montag
for(Anzeige_Kalender_TagesZahler=0;
    Anzeige_Kalender_TagesZahler < TagKurzNamenFeld_Laenge;
    Anzeige_Kalender_TagesZahler++)
{
    // ----- Text als Kurzname des Tages holen
    Anzeige_Kalender_Kette=TagKurzNamenFeld[Anzeige_Kalender_TagesZahler];

    // ----- und Text erzeugen

```



```

Anzeige_Kalender_HTMLCode_TD_Erzeugen(
    Anzeige_Kalender_Kette,
    Kalender_Kopf_HintergrundFarbe,
    Kalender_SpaltenKopf_SchriftFarbe,
    Tag_SchriftGroesse,
    Kalender_Kopf_SchriftArt,
    Kalender_Kopf_HintergrundFarbe
);
}

Anzeige_Kalender_HTMLCode+='<TR>'; // Ende der Wocheneinteilung in 7 Tage (Spaltenköpfe)

// +++++ ab 3. TR erfolgt die Wochenauflistung
//   pro Woche 1 Zeile
//   pro Zeile 1 TR
//   ab 3. TR im gesamten Kalender

// ----- init des Tageszähler des Monats
Anzeige_Kalender_TagesZahler = 1;

// ----- Wochen abklappen also Zeilen (TR)
for(Anzeige_Kalender_Zahler_TR=0;
    Anzeige_Kalender_Zahler_TR < 6;
    Anzeige_Kalender_Zahler_TR++)
{
    // - - - aktuelle Woche (Zeile) erzeugen
    Anzeige_Kalender_HTMLCode+='<TR>';

    // - - - pro Spalte 1 TD erzeugen: Nur die ersten 6 Tage der Woche, OHNE Sonntag !
    for(Anzeige_Kalender_Zahler_TD=0;
        Anzeige_Kalender_Zahler_TD < 6; // ohne Sonntag !
        Anzeige_Kalender_Zahler_TD++)
    {
        // - - - prüfen ob 1. TR-Zeile noch nicht abgeklappert wurde
        if( (Anzeige_Kalender_Zahler_TR == 0)
            && (Anzeige_Kalender_Zahler_TD < Start)
        )
        {
            // 1. TR und nur 1. TD
            Anzeige_Kalender_Tag_HTMLCode_TD_Erzeugen('&#160;','false');
        }
        else
        {
            // 1. TR ab 2.TD oder ab 2.TR

            // - - - prüfen ob alle Tage bereits abgeklappert wurden
            if (Anzeige_Kalender_TagesZahler > Stop)
            {
                // alle Tage sind abgeklappert
                Anzeige_Kalender_Tag_HTMLCode_TD_Erzeugen('&#160;','false');
            }
            else
            {
                // 1. TR ab 2.TD oder ab 2.TR
                // es sind noch Tage abzuklappen

                // - - - prüfen ob aktueller Tag der Tag von JETZT ist, also Heute ist
                if(Anzeige_Kalender_TagesZahler==Zeit_Jetzt_Tag_Numerisch)
                {
                    // 1. TR ab 2.TD oder ab 2.TR
                    // es sind noch Tage abzuklappen
                    // aktueller Tag ist heute

                    Anzeige_Kalender_Tag_HTMLCode_TD_Erzeugen(Anzeige_Kalender_TagesZahler,true);
                }
                else
                {

```



```

// 1. TR ab 2.TD oder ab 2.TR
// es sind noch Tage abzuklappen
// aktueller Tag ist nicht heute
Anzeige_Kalender_Tag_HTMLCode_TD_Erzeugen(Anzeige_Kalender_TagesZahler,false);
}

// - - - Tageszähler erhöhen für nächsten Tag als neuen aktuellen
Anzeige_Kalender_TagesZahler++;
}
}

// - - - ALLE TD der ersten 6 Tage der aktuellen Woche (Zeile) wurden erzeugt
// offen ist der 7. Tag, also der Sonntag (falls vorhanden)

// prüfen ob alle Tage abgeklappt wurden, also ob es keinen Sonntag gibt
if(Anzeige_Kalender_TagesZahler > Stop)
{
// keine Tage mehr abzuklappen, kein Sonntag mehr da
Anzeige_Kalender_Sonntag_HTMLCode_TD_Erzeugen('&#160;';false);
}
else
{
// - - - Es sind noch Tage abzuklappen, also 7. Tag = Sonntag vorhanden

// - - - prüfen ob der aktuelle Tag der von heute ist,
// wenn ja, dann ist heute Sonntag
if(Anzeige_Kalender_TagesZahler==Zeit_Jetzt_Tag_Numerisch)
{
// heute ist Sonntag, also den Tag mit anderem Hintergrund erzeugen
// als alle anderen Sonntage
Anzeige_Kalender_Sonntag_HTMLCode_TD_Erzeugen(Anzeige_Kalender_TagesZahler,true);
}
else
{
// heute ist nicht Sonntag, also den Tag dem normalen Hintergrund für Sonntage erzeugen
Anzeige_Kalender_Sonntag_HTMLCode_TD_Erzeugen(Anzeige_Kalender_TagesZahler,false);
}

// - - - Tageszähler erhöhen für nächsten Tag als neuen aktuellen
Anzeige_Kalender_TagesZahler++;
}

// - - - aktuelle Woche also TD beenden
Anzeige_Kalender_HTMLCode+='</TR>';
}

// +++++ Ende des Kalenders
Anzeige_Kalender_HTMLCode+='</TABLE>';

// +++++ und Kalenderdaten neu rendern
document.all.ID_Kalender.innerHTML = Anzeige_Kalender_HTMLCode;

// +++++ prüfen ob der Timer zur Erkennung der Tages-, Monats- und Jahreswechsel
// verwendet werden soll
if (Kalender_Timer_Verwenden)
{window.setTimeout('Anzeige_Kalender()',5000);}
}

// -->
</SCRIPT>
</HEAD>

<BODY BGCOLOR=#3A6EA5>
<!-- BODY mit identischer Hintergrundfarbe wie laut Kalender_Kopf_HintergrundFarbe -->

<DIV ID="ID_Kalender"></DIV>
<!-- Der DIV wird nach seiner Erzeugung erst gefüllt: per .innerHTML -->

```



```
<SCRIPT LANGUAGE="JScript">
<!--
Anzeige_Kalender();
// -->
</SCRIPT>
</BODY>
</HTML>
```

25.2. Aktive Inhalte im Internet Explorer

Ab IE 6.0 ist das Blockieren aktiver Inhalte möglich, z.B. als Standardeinstellung. Es wird also dem IE verboten, JScript zu nutzen. Daher muss mit Start der Webseite auf das Blockieren von Inhalten der Webseite, die auf JScript basieren, aufmerksam gemacht werden. Bleibt die Blockierung aktiv, so muss die Webseite ALLE Elemente, die per Script angesteuert werden, inaktiv machen: Am besten garnicht erst anzeigen. Oder es wird eine scriptfreie Version der Webseite per <NOSCRIPT> aktiviert, wobei dann Browser vorzuziehbar sind, die z.B. CSS exakter rendern als der IE (will man keine IE-spezifischen HTML-Elemente verwenden).

Wenn der IE 6.x aktive Inhalte blockiert, wird NOSCRIPT-Tag aktiviert, Ausnahme: Frameset

FRAMESET ist ein aktiver Inhalt:

Da der Frameset anstelle <BODY> kodiert sein muss, gilt:

Alle Tags, die für BODY zulässig sind, werden ignoriert, auch NOSCRIPT.

Wird neben Frameset noch BODY kodiert, so wird Frameset ignoriert.

Die Freigabe der Scriptblockierung erzeugt Ausführung aller Script-Teile inklusive der Eventauslösungen

Bsp.: Folgendes funktioniert vom Dokument, das window.open() hat im geöffneten Dokument (Quelltext im Dokument das window.open() verwendet):

```
function Y_unload(X00){X85[X00].close();}

var X85=new Array();var X86=new Array();

X85[0]=window.open(...);

var X87='parent.Y_unload(0)'; X86[0]=new Function(",X87);

X85[0].document.body.onunload=X86[0];
```

Wird die Scriptblockierung im geöffneten Fenster abgeschaltet,
so wird das Fenster geschlossen, weil onunload ausgelöst wird.

Achtung: document.body.onunload funktioniert ev. nicht mehr
wenn z.B. mit attachevent() aktiviert wurde

Folgende Metatags sind für den IE 6.x aktiver Inhalt:

```
<META HTTP-EQUIV="imagetoolbar" CONTENT="no">
    unterdrückt NICHT IE-Kontextmenü rechte Maus auf Bild

<META HTTP-EQUIV="site-enter" CONTENT="revealtrans(duration=0.3, transition=12)">
<META HTTP-EQUIV="site-exit" CONTENT="revealtrans(duration=0.3, transition=12)">
```

25.3. Aktive Inhalte im per .replace() geöffnetem Dokument

Je nach IE-Version kann folgendes gelten:

Dokumente mit replace() im Quelltext

dürfen KEIN Script enthalten: weder im Head noch z.B. als onevent="...."; im HTML-Tag
können also kein Button zum Schließen des Fensters, das im Kiosk-Modus geöffnet wurde, enthalten !
Empfehlung: NICHT im Kioskmodus öffnen !

Dokumente ohne replace() im Quelltext

dürfen Script durch das Dokument, das open() ausführt, untergejubelt bekommen:
z.B. Button im zu öffnenden Dokument ohne onclick-Belegung im Tag
onclick-Handler durch Dokument, das open() aufruft, belegen im geöffneten Fenster

wenn kein replace() im Quelltext des Dokumentes enthalten ist, dann gilt:

Ein per open() geöffnetes Fenster mit blockierten Script im Dokument (blockierter Aktiver Inhalt)
- wobei es wurscht ist, welches Zielfenster angegeben wird -
wird sofort vom Internet Explorer geschlossen, wenn

UND der User das blockierte Script manuell freigibt
UND das Script im Dokument kodiert ist.

Ein unload-Handler ist verwendbar.



wird NICHT vom Internet Explorer geschlossen, wenn

- UND kein replace() enthalten ist, also das Dokument kein anderes lädt
- UND der User das blockierte Script manuell freigibt
- UND das Script von Dokument, das open() aufruft, im geöffneten Dokument nach dem open() erzeugt wird, also unterjubelt.

Ein unload-Handler ist verwendbar.

Warnung: Wenn für das Dokument, das open() aufruft, die Scriptblockade bereits beseitigt wurde, wird der untergejubelte Code im geöffneten Fenster NICHT mehr als blockierter Script erkannt !

wenn replace() im Quelltext des Dokumentes enthalten ist, dann gilt:

Ein per open() geöffnetes Fenster mit blockierten Script im Dokument (blockierter Aktiver Inhalt) - wobei es wurscht ist, welches Zielfenster angegeben wird - wird NICHT vom Internet Explorer geschlossen, wenn

- replace() enthalten ist, also das Dokument ein anderes lädt
- UND der User das blockierte Script manuell freigibt.

Ein unload-Handler ist NICHT verwendbar.

Bsp.: Folgendes funktioniert vom Dokument, das window.open() hat im geöffneten Dokument (Quelltext im Dokument das window.open() verwendet):

```
function Y_unload(X00){X85[X00].close();}

var X85=new Array();var X86=new Array();

X85[0]=window.open(...);

var X87='parent.Y_unload(0)'; X86[0]=new Function(",X87);

X85[0].document.body.onunload=X86[0];
```

Wird die Scriptblockierung im geöffneten Fenster abgeschaltet, so wird das Fenster geschlossen, weil onunload ausgelöst wird.

Achtung: document.body.onunload funktioniert ev. nicht mehr wenn z.B. mit attachevent() aktiviert wurde

25.4. ID-Attribut und .createElement()

Wird im HTML-Code für createElement() das ID-Attribut kodiert, dann gilt folgendes:

Das ID-Attribut wird NUR dann verwendet, wenn KEINE gleichnamige Variable deklariert ist, die zum Zeitpunkt vor createElement() bereits existiert: z.B. keine gleichnamige globale Variable.

Dem ID-Attribut wird der Zeiger aus createElement() bzw. appendChild() zugewiesen (beide Methoden liefern identischen Zeiger)

Es ist eigentlich sinnlos, das ID-Attribut zu kodieren, da createElement() einen Zeiger bereits liefert, der anstelle ID verwendet werden kann (ID ist auch nichts anderes als ein Bezeichner für einen Zeiger).

zu createElement():

Tag auch in der Form '< ...>' mit Attributen möglich

Ende-Tag kann muss aber nicht kodiert werden (falls HTML-Syntax einen Endetag vorschreibt)

Bsp: '<DIV STYLE="....">' oder '<DIV STYLE="...."></DIV>'

möglichst keine Elemente zwischen den Tags mit angeben, sondern diese NACH der Erzeugung und append() erzeugen per

z.B. .innerText bzw. .innerHTML (Achtung: beide Eigenschaften schaltet aktives BGSOUND auf stumm !)

also z.B. nicht '<DIV STYLE="....">TestText</DIV>'

'TestText</DIV>'

es werden z.B. HTML-Code-Fehler ignoriert, aber nicht alle

wenn ignoriert, so Objekt erzeugt aber eben nicht mit vollständigem HTML-Code

Daher auch unbedingt .innerText und .innerHTML nach der Erzeugung und append belegen !

Achtung: Für das Hinzufügen von Elementen in den BODY (document.body) per DOM-Funktion createElement() MUSS der Body komplett geparkt sein (document.body.readyState == 'complete').

Grund: Es wird standargemäß immer am Ende des BODY angefügt.

Für das Hinzufügen nicht an das Ende des BODY muss im HTML-Code ein Platzhalter z.B. DIV kodiert sein, innerhalb dessen dann die neuen HTML-Elemente erzeugt werden.

Nachfolgend ein Beispiel der falschen Brwoser-Implementation vom Objekt OBJECT per appendChild() das sich nur um genau 1 PARAM-Objekt erweitern lässt



kein .innerHTML besitzt

```
<HTML>
<BODY>
<SCRIPT LANGUAGE='JScript'>
<!--
var X24=false;           // Returnwert, Annahme: Objekt nicht installiert
var X28="";
var X30;                 // Arbeitszeiger 1 Zeiger aus createElement()
var X31;                 // Arbeitszeiger 2 Zeiger auf aktuelles PARAM-Objekt
var X32;                 // Zeiger auf Container-DIV
var X33;                 // Zeiger auf WebSpeech-Objekt im DIV

// +++++ Objekt-Container als DIV erzeugen
X30=document.createElement('DIV');X24=(X30!=null);
if(X24){X32=document.body.appendChild(X30);X24=(X32!=null);}
if(X24){X24=(X30==X32);}

if(X24)
{ // +++++ Objekt erzeugen im DIV-Container
  X30=document.createElement('OBJECT');X24=(X30!=null);
  if(X24){X33=X32.appendChild(X30);X24=(X33!=null);}
  if(X24){X24=(X30==X33);}
}

if(X24)
{
  // +++++ Objekt: PARAM's-Objekte im Objekt erzeugen
  // ---- AUTHKEY
  X30=document.createElement('PARAM');X24=(X30!=null);
  if(X24){X31=X33.appendChild(X30);X24=(X31!=null);}
  if(X24){X24=(X30==X31);}
  if(X24){X31.name='AUTHKEY';X31.value='tt';}
}

if(X24)
{ // ---- AUTOSTART
  X30=document.createElement('PARAM');X24=(X30!=null);
  if(X24){X31=X33.appendChild(X30);X24=(X31!=null);}
  if(X24){X24=(X30==X31);}
  if(X24){X28='0';if(X16){X28='1';}X31.name='AUTOSTART';X31.value=X28;}
}
//-->
</SCRIPT>
</BODY>
</HTML>
```

Zeile 37 Zeichen 11 ungültiges Argument: AUTOSTART-Param-Objekt nicht implementierbar per appendChild

OBJECT kann nur im genau 1 PARAM-Objekt erweitert werden per createElement()
wurde falsch implementiert
OBJECT besitzt NICHT .innerHTML

Lösung: DIV per .innerHTML füllen, also HTML-Code des Objektes OBJECT

25.5. *.focus() beim IE 7*

windows.focus()
document.focus()
und body.focus()
funktionieren NICHT zwischen
Register in einem IE-Fenster
zwischen Fensters z.B. in Taskleiste

Hinweis:

.focus() setzt Element aktiv, gibt dem Element den Focus und feuert onfocus
.setActive() ist Teilmenge von .focus(): nur das aktiv setzen
funktioniert nicht mit allen Elementen, mit denen .focus() funktioniert



25.6. *window.status beim IE 7*

Der IE 7 hat ein anderes Layout und erzeugt Zusatzanzeigen, die per JScript für den IE 6 nicht verwaltbar sind.
Der IE 7 ist geschwätziger als der IE 6: Dateiladeinformationen zeigt der IE 7 gern an - natürlich ausser von windows.status.

Apache-HTTP-Server verhält sich bei IE 6 und IE 7 anders bezüglich window.status.

25.7. *Renderqualität und animierte Gif beim IE 7*

Die Renderqualität mit der Installation des IE 7 standardgemäß auf LCD-Monitor eingestellt, und zwar unabhängig davon, welche Monitor-Hardware von Windows bereits erkannt wurde. Der User eines Röhren-Monitors wird dann eine entsetzlich verschwommene Schrift erkennen. Diese nennt sich "ClearType" und dient nur LCD-Monitoren. Das allerdings muss ein User erst mal wissen. Microsoft geht wohl davon aus, dass es Röhrenmonitor-User kaum noch gibt. Per Hilfe im IE 7 ist unter Cleartype ist zu erfahren, wo diese Schriftart im IE abschaltbar ist.

Animierte Gifs (mit Timer), die unter IE 6 korrekt laufen, müssen unter IE 7 im Timer nicht mehr laufen:

z.B. garnicht mehr sichtbar, oder Timer nicht verwendet.

Dann müssen animierte Gif-Bilder nach IE-Version bereitgestellt werden.

25.8. *.createPopup() und Popupblocker beim IE 7*

Die Microsoft Browser-Version IE 7 ist nicht abwärtskompatibel bezüglich Popup per window.createPopup()

Popup per window-Objekt ist ein Markenzeichen des IE, das im IE 7 nicht mehr fehlerfrei nutzbar ist.

Der Fehler liegt in der Popup-Blockerverwaltung des IE und wurde mit dem IE 7 implementiert.

Der Fehler tritt nicht auf, wenn ein Fenster per window.open() erzeugt wurde.

Bedingung:

Scripfehleranzeige ist erlaubt im IE 7

Popupblocker ist im IE abgeschaltet

ein aktives Fenster (Register) mit Dokument, dass fortlaufend (rekursiv) genau 1 window.popup per .show()erzeugt.

ein weiteres Fenster (Register) z.B. leere Seite (about:blank)

beide (Register) liegen in einer gemeinsamen IE-Instanz

Ablauf: Wird Focus auf Register der leeren Seite gehalten und wird parallel das Popup per .show() erzeugt,

bricht der Browser das Dokument mit .show() ab (Scriptfehler).

Der Popupblocker für die leere Seite verursacht den Programmfehler im Dokument mit .show(). Es wird folgende

Meldung angezeigt (in der Informationsleiste):

'Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen.'

Die Bedeutung der Meldung laut Microsoft-Hilfe im IE 7:

Der Popupblocker hat ein Populfenster geblockt. Sie können den Popupblocker deaktivieren

oder Popups temporär zulassen, indem Sie auf die Informationsleiste klicken.

Die Realität zur obigen Meldung ist völlig anders:

Linke oder rechte Maus auf die Meldung liefert z.B. Einstellungen darunter

Popupblocker einschalten

weitere Informationen

jedoch keine Möglichkeit wie laut Bedeutung

Damit gilt: Der abgeschaltete Popupblocker ist in Wirklichkeit aktiv.

Pikant: Ein Popup erscheint normalerweise auch über fremde Fenster, die nicht das Popup erzeugt haben (z.B. Fenster einer Windowsanwendung z.B. einer anderen IE-Instanz)

Der Popupblocker des IE bemerkt aber NUR Webseite, die das Popup erzeugt.

Durch das Abwürgen von Popup wird das Popup natürlich auf und für anderen Seiten nicht relevant; im Falle einer anderen IE-Instanz also auch für diese nicht relevant, obwohl diese Instanz per Popupblocker verwaltet wird.

Der Popupblocker beschneidet die Popup-Reichweite an der Wurzel, ist aber nicht objektorientiert zu den anderen Webseiten (die nicht das Popup erzeugt haben).

Der Popupblocker ist nicht als Filter aufgesetzt sondern reingestrickt worden.

Der Popupblockerfehler verändert die Eventverwaltung:

Es werden u.a. ignoriert

onfocus

onblur

onfocusin

onfocusout

und viele andere, so dass trotz Events z.B. des Body der Popupblockerfehler entsteht.

// nachfolgender Code setzt focus nicht neu: Fenstereintrag in Taskleiste blinkt eventuell

window.focus();

window.document.focus();

if(document.body!=null)

{if(document.body.style!='hidden') // wenn hidden so focus() nicht möglich (Scriptfehler erzeugt)

{document.body.focus();}

}

// wenn paralleles Fenster offen (on oder offline), so Scriptfehler erzeugt

popupzeiger.show(...);

Hinweis: Der Popupfehler ist so elementar, dass die vielen Beta-Testphasen des IE mehr als fragwürdig erscheinen, wie die Angabe von



Microsoft, dass Code neu programmiert wurde, um den IE sicherer zu machen.

25.9. **.style.zIndex nach createElement() des Objektes mit .style im IE 6**

Es wird vorausgesetzt, dass direkt nach createElement() auch appendChild() erfolgt !

für document.body gilt:

für die in document.body per createElement erzeugten Objekte OHNE deren Kinder
ist es nicht möglich, per zIndex die Reihenfolge laut der createElement-Aufrufe zu verändern.
braucht zIndex nicht kodiert werden

Achtung: im Container erzeugte Objekte z.B. DIV (Container in document.body erzeugt)

Wenn sich die Objekte überlappen (durch setzen von .style.left und .style.top):

Das zuerst erzeugte Objekt liegt ganz hinten.

Das zuletzt erzeugte Objekt liegt ganz vorn.

Ein ohne Container erzeugtes Objekt verhält sich anders:

Da zuletzt erzeugte Objekt liegt ganz hinten.

zIndex funktioniert allerdings bezüglich Container-Objekt, das nicht document.body ist:

Dabei spielt es keine Rolle, ob der z-Index > 0 oder < 0 ist.

Entscheidend ist, dass style.zIndex kodiert wurde mit verschiedenen Werten,

z.B. -1 dann -2 etc.

wobei die Wertefolge Ausschlag gebend ist:

Das Objekt mit zIndex als im Absolutbetrag höchsten Wert liegt

ganz oben, also vor allen anderen Objekten.

Das Objekt mit zIndex als im Absolutbetrag niedrigsten Wert liegt

ganz unten, also hinter allen anderen Objekten.

Bsp: Ein Objekt, das als letztes Element in document.body per createElement erzeugt wird,
hat den obersten zIndex.

Die Angabe .style.zIndex=-100; wird zwar ausgeführt, aber ohne Wirkung

Konsequenz: Elemente in einem Container, die sich überlappen sollen,

müssen in der Überlappungsfolge in document.body erzeugt werden.

oder man erzeugt einen Container z.B. einen Rahmen-DIV

Achtung: IMG verfügt über keinen style.zIndex.

Es muss also Container z.B. DIV verwendet werden

25.10. **.eval()**

eval verlangt String innerhalb ()

Variablen, deren Inhalte kein String sind, müssen als 'variablen_bezeichner' kodiert werden.
siehe unten.

Variablen, deren Inhalt String sind, dürfen NICHT als 'variablen_bezeichner' kodiert werden,
sondern nur als variablen_bezeichner.

String wird ausgeführt, wenn ausführbar (sonst passiert nichts, auch kein Scriptfehler)

Ist im String als Wert ein Variablenbezeichner, so ist dieser ja ausführbar.

Beispiel: Variable mit Zeiger --> Zeiger verwenden
mit String --> String ausführen

Ist im String als Wert kein Variablenbezeichner oder kein JScript-Kommando,
so kann nichts ausgeführt werden.

Wird eine Stringvariable erkannt, dann wird der Inhalt der Variablen als String ausgeführt wie oben.

Variablen sind ausführbar, weil sie eine Adresse besitzen, ab der der Inhalt der
Variablen steht, der ausgeführt wird.

Bei einer Stringvariable innerhalb ' ' wird der Variablenbezeichner als String verwendet

und wie wie oben ausgeführt. Es wird also nicht die Adresse der Variablen verwendet.

Da aber Variablenbezeichner NICHT identisch sein dürfen mit JScript-Kommandos,
kann also nichts ausgeführt werden.

Beispiel var drei+10 ist nicht zulässig
var dreipus10 ist ein JScript-Kommando

Ausführbarkeit einer Variablen: Adresse der Variablen ermitteln und den Inhalt ausführen.

Beispiel:

```
var x=0;
```

```
function test()  
{ x++; alert(x);}
```



```

var ZeigerAufFunktion=test;           // ohne () da sonst sofort ausgeführt wird
var Kette='ZeigerAufFunktion()';      // auch als Zeichenkettenvariable
eval(Kette);                          // '1', Inhalt von Kette auswerten
eval('Kette');                        // Zeichenfolge 'Kette' auswerten: Da diese
                                      // weder ein JScript-Kommando ist
                                      // noch ausführbar ist
                                      // passiert nichts.
                                      // Es wird nicht die Variable Kette referenziert !!!
                                      // sondern der String 'Kette' ausgeführt.

eval('test()');                       // '2', String auswerten, der ausführbar ist

```

Funktion als Zeiger starten

Beispiel:

```

var x=0;

function test()
{x++; alert(x);}

var ZeigerAufFunktion=test;           // ohne () da sonst sofort ausgeführt wird
eval('ZeigerAufFunktion'+ '()');      // jetzt mit ()

var Kette='ZeigerAufFunktion()';      // auch als Zeichenkettenvariable
eval(Kette);                          // Achtung: ist der Inhalt von ZeigerAufFunktion kein Zeiger,
                                      // so bringt eval einen Scriptfehler !

eval('test()');                       // identisch mit test();

```

Das Testen eines Zeigers auf != null ist mit eval nicht möglich, wenn
eval den Zeiger erst ermittelt aber der Zeigerwert nicht bildbar ist
wobei eval KEINEN Rückkehrcode liefert !!!

Beispiel:

```

var z;                                // Zeiger
var n='testen';                       // Fehler, Es ist 'test' gemeint, jedoch nicht kodiert
                                      // Funktion testen() gibt es nicht

var k="";
var f=false;
var x=0;

function test()
{x++; alert(x);}

// Funktion testen() ist nicht deklariert

k='z=' + n + ' ';                    // 'z=testen;'

// fehlerhaften Zeiger auf Routine testen() holen --> erzeugt Scriptfehler

eval(k);                             // "testen" wird verwendet auch wenn "test" gemeint ist
                                      // testen() ist nicht deklariert, also Zeigerzuweisung auf z nicht möglich
                                      // eval liefert keinen Rückkehrcode !!!

// wegen Scriptfehler wird nachfolgender Befehl nicht abgearbeitet
if (z != null)
{alert();}

```

Damit gilt: Ein Zeigerwert, der per eval zugewiesen wird, muss korrekt sein.

Also ist die Abfrage auf != null nach eval sinnlos !

Alternative sind nicht:

```

eval('if (k != null){....});          // ergibt true, da Variable k deklariert und gefüllt ist
                                      // k wird also nicht durch 'z=' + n + ' ' ersetzt !

eval('if ( 'z=' + n + ' ');           // anstelle von 'z=' + n + ' '

```



```

    != null
  )
    { .....}'
);

```

// ist auch 'z=testen;' möglich

da zuerst der Ausdruck in der if-Anweisung berechnet wird, ergibt sich also aus 'z=' + n + ';' der Wert 'z=testen;'

wird 'z=testen;' ausgeführt, entsteht der Scriptfehler, so dass der Vergleich != null nicht zum Zuge kommt.

```

eval('alert(' + n + '!= null);'); // --- > Scriptfehler, da n als Zeiger interpretiert wird
                                //                                         wegen Vergleich mit null
                                //                                         alert nicht ausgeführt

```

eval() besitzt eine interne return-Anweisung, die nicht abschaltbar ist

```

var Wert=eval(...);           internes Return wird an Wert geliefert
eval(...);                   internes Return wird abgearbeitet, aber es wird nichts geliefert:
                                Kodierung ist zulässig !

```

Wegen internem Return sollte innerhalb () kein return kodiert sein:

inneres return wird zuerst abgearbeitet, und dann das return aus (), was aber sinnlos ist

Eventhandler mit return-Anweisung:

Variante 1

```
return eval("return window.event.returnValue");
```

eval selbst hat ein internes return, das an return vor eval liefert
innerhalb eval ist ein return kodiert, dass ebenfalls liefert, aber unsinnig ist

```

richtig kodiert:    return eval("window.event.returnValue");
                   oder return(eval("window.event.returnValue"));

```

Variante 2

```
eval("return window.event.returnValue");
```

es muss ein return vor eval() kodiert worden sein, wurde aber vergessen

das interne Return von eval ersetzt NICHT das Return vor eval()
Es kann die Fehlermeldung erscheinen, dass das return innerhalb ()
ausserhalb des Eventhandlers liegt.
Grund: eval() ist eine JScript-Funktion, die im Kontext des
Handlers läuft, aber ein internes return besitzt,
das nichts mit dem Handler zu tun hat

Variablen-Erzeugung innerhalb eval()

```
var variable=eval('...');           eval liefert also return-Wert
```

ist identisch mit eval('var variable=...');

```
z.B.  var X17=eval('bs45.' + bs13 + 'Vector3;');
```

```
oder eval('var X17=bs45.' + bs13 + 'Vector3;');
```

Die var-Erzeugung erfolgt lokal zum Kontext:

wenn innerhalb einer Funktion, dann lokal zur Funktion

wenn ausserhalb einer Funktion, dann lokal zum Dokument mit JScript, also global IM Dokument

25.11. wichtige Events

Events bei DOM-Änderung

Wurden Events für window, document, body zugewiesen z.B. für oncontextmenu
und werden danach DOM-Änderungen vollzogen,
dann können die Events vom Browser während der DOM-Änderungen ignoriert werden,



aber am Ende aller DOM-Änderungen wieder aktiv sein.

Grund könnte Ressourcen-Mangel des Browser sein, wenn viele DOM-Änderungen nacheinander folgen.
Ausserdem ändert sich oft body.

Mousevents

Das Event onmouseover kann NUR durch die Bewegung der Maus per User ausgelöst werden.

Wenn das Objekt, das onmouseover hat, per Script anhand style.top und style.left verschoben wird und dabei UNTER den unbewegten Mauszeiger gelangt, wird KEIN onmouseover ausgelöst.

Das selbe Prinzip gilt für andere onmouse-Events

Wird die Sanduhr angezeigt, also das Betriebssystem wird aktiv,
dann kann Event erzeugt werden, da die Sanduhr ausserhalb des Dokumentes liegt und der Fokus sich ändert auf Sanduhr
z.B. mousemove bei IMG wenn cursor über IMG
mouseout für BODY

Funktion z.B. Eventhandler mit returnValue und mit folgenden Anweisungen im Funktionsrumpf:

Variante 1

return (eval-ausdruck mit return-Anweisung);

eval-ausdruck z.B. eval("return window.event.returnValue");

Es wird innere return-Anweisung von eval abgearbeitet
vom äusseren return entgegengenommen

Variante 2

eval-ausdruck mit return-Anweisung

eval-ausdruck z.B. eval("return window.event.returnValue");

Es wird innere return-Anweisung von eval abgearbeitet

Ist jetzt z.B. in einem Eventhandler eine äussere return-Anweisung
zwar notwendig zu kodieren,
aber nicht kodiert WORDEN,
dann kann eine Fehlermeldung, dass ein return ausserhalb der
Funktion liegt

Übersicht zu wichtigen Objekten und Events

Hinweis: bubbles entspricht durchreichen der Events nach oben entlang der DOM-Hiearchie
cancels entspricht unterbrechbar

.fireEvent()

Fires a specified event on the object.

kann nur innerhalb eines Eventhandlers aktiviert werden, da ansonsten wirkungslos

Syntax

bFired = object.fireEvent(sEvent [, oEventObject])

sEvent Required. String that specifies the name of the event to fire.

oEventObject Optional. Object that specifies the event object from which to obtain event object properties.

Return Value

Boolean. Returns one of the following values:

true Event fired successfully.

false Event was cancelled.

fireEvent() setzt automatisch window.event.cancelBubble auf false, also kein Hochreichen des Events in der DOM-

Hierarchie

fireEvent() liefert

true, wenn

Event erfolgreich gefeuert

oder Event nicht feuerebar aber auch nicht cancelbar



false, wenn
Event nicht erfolgreich gefeuert
UND nicht gecancelt wurde

Beispiel:

```
<HTML>
<HEAD>
<SCRIPT>
function fnFireEvents()
{
    oDiv.innerText = "The cursor has moved over me!";
    oButton.fireEvent("onclick");
}
</SCRIPT>
</HEAD>
<BODY>
<h1>Using the fireEvent method</h1>
By moving the cursor over the DIV below, the button is clicked.
<P>
<DIV ID="oDiv" onmouseover="fnFireEvents();">
Mouse over this!
</DIV>
<p>
<BUTTON ID="oButton" ONCLICK="this.innerText='I have been clicked!'" >
Button</BUTTON>
</BODY>
</HTML>
```

Objekte mit der Methode sind

A, ACRONYM, ADDRESS, APPLET, AREA, B, BASE, BASEFONT, BDO, BGSOUND, BIG, BLOCKQUOTE, BODY, BR, BUTTON, CAPTION, CENTER, CITE, CODE, COL, COLGROUP, COMMENT, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, HEAD, hn, HR, HTML, I, IFRAME,

IMG,

INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LINK, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, NOFRAMES, NOSCRIPT,

OBJECT,

OL, OPTION, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SCRIPT, SELECT, SMALL, SPAN, STRIKE, STRONG, styleSheet, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TITLE, TR, TT, U, UL, VAR, WBR, XML, XMP

window-Events

onactivate Fires when the object is set as the active element.
onafterprint Fires on the object immediately after its associated document prints or previews for printing.
onbeforedeactivate Fires immediately before the activeElement is changed from the current object to another object in the parent

document.

onbeforeprint Fires on the object before its associated document prints or previews for printing.
onbeforeunload Fires prior to a page being unloaded.
onblur Fires when the object loses the input focus.
oncontrolselect Fires when the user is about to make a control selection of the object.
ondeactivate Fires when the activeElement is changed from the current object to another object in the parent document.
onerror Fires when an error occurs during object loading.
onfocus Fires when the object receives focus.
onhelp Fires when the user presses the F1 key while the browser is the active window.
onload Fires immediately after the browser loads the object.
onmove Fires when the object moves.
onmoveend Fires when the object stops moving.
onmovestart Fires when the object starts to move.
onresize Fires when the size of the object is about to change.
onresizeend Fires when the user finishes changing the dimensions of the object in a control selection.
onresizestart Fires when the user begins to change the dimensions of the object in a control selection.
onscroll Fires when the user repositions the scroll box in the scroll bar on the object.
onunload Fires immediately before the object is unloaded.

document-Events

onactivate Fires when the object is set as the active element.
onbeforeactivate Fires immediately before the object is set as the active element.



onbeforecut Fires on the source object before the selection is deleted from the document.

onbeforedeactivate Fires immediately before the activeElement is changed from the current object to another object in the parent document.

onbeforeeditfocus Fires before an object contained in an editable element enters a UI-activated state or when an editable container object is control selected.

onbeforepaste Fires on the target object before the selection is pasted from the system clipboard to the document.

onclick Fires when the user clicks the left mouse button on the object.

oncontextmenu Fires when the user clicks the right mouse button in the client area, opening the context menu.

oncontrolselect Fires when the user is about to make a control selection of the object.

oncut Fires on the source element when the object or selection is removed from the document and added to the system clipboard.

ondblclick Fires when the user double-clicks the object.

ondeactivate Fires when the activeElement is changed from the current object to another object in the parent document.

ondrag Fires on the source object continuously during a drag operation.

ondragend Fires on the source object when the user releases the mouse at the close of a drag operation.

ondragenter Fires on the target element when the user drags the object to a valid drop target.

ondragleave Fires on the target object when the user moves the mouse out of a valid drop target during a drag operation.

ondragover Fires on the target element continuously while the user drags the object over a valid drop target.

ondragstart Fires on the source object when the user starts to drag a text selection or selected object.

ondrop Fires on the target object when the mouse button is released during a drag-and-drop operation.

onfocusin Fires for an element just prior to setting focus on that element.

onfocusout Fires for the current element with focus immediately after moving focus to another element.

onhelp Fires when the user presses the F1 key while the browser is the active window.

onkeydown Fires when the user presses a key.

onkeypress Fires when the user presses an alphanumeric key.

onkeyup Fires when the user releases a key.

onmousedown Fires when the user clicks the object with either mouse button.

onmousemove Fires when the user moves the mouse over the object.

onmouseout Fires when the user moves the mouse pointer outside the boundaries of the object.

onmouseover Fires when the user moves the mouse pointer into the object.

onmouseup Fires when the user releases a mouse button while the mouse is over the object.

onmousewheel Fires when the wheel button is rotated.

onmove Fires when the object moves.

onmoveend Fires when the object stops moving.

onmovestart Fires when the object starts to move.

onpaste Fires on the target object when the user pastes data, transferring the data from the system clipboard to the document.

onpropertychange Fires when a property changes on the object.

onreadystatechange Fires when the state of the object has changed.

onresizeend Fires when the user finishes changing the dimensions of the object in a control selection.

onresizestart Fires when the user begins to change the dimensions of the object in a control selection.

onselectionchange Fires when the selection state of a document changes.

onstop Fires when the user clicks the Stop button or leaves the Web page.

body-Events

onactivate Fires when the object is set as the active element.

onafterprint Fires on the object immediately after its associated document prints or previews for printing.

onbeforeactivate Fires immediately before the object is set as the active element.

onbeforecut Fires on the source object before the selection is deleted from the document.

onbeforedeactivate Fires immediately before the activeElement is changed from the current object to another object in the parent document.

onbeforeeditfocus Fires before an object contained in an editable element enters a UI-activated state or when an editable container object is control selected.

onbeforepaste Fires on the target object before the selection is pasted from the system clipboard to the document.

onbeforeprint Fires on the object before its associated document prints or previews for printing.

onbeforeunload Fires prior to a page being unloaded.

onclick Fires when the user clicks the left mouse button on the object.

oncontextmenu Fires when the user clicks the right mouse button in the client area, opening the context menu.

oncontrolselect Fires when the user is about to make a control selection of the object.

oncut Fires on the source element when the object or selection is removed from the document and added to the system clipboard.

ondblclick Fires when the user double-clicks the object.

ondeactivate Fires when the activeElement is changed from the current object to another object in the parent document.

ondrag Fires on the source object continuously during a drag operation.

ondragend Fires on the source object when the user releases the mouse at the close of a drag operation.

ondragenter Fires on the target element when the user drags the object to a valid drop target.

ondragleave Fires on the target object when the user moves the mouse out of a valid drop target during a drag operation.

ondragover Fires on the target element continuously while the user drags the object over a valid drop target.

ondragstart Fires on the source object when the user starts to drag a text selection or selected object.



ondrop Fires on the target object when the mouse button is released during a drag-and-drop operation.
 onfilterchange Fires when a visual filter changes state or completes a transition.
 onfocusin Fires for an element just prior to setting focus on that element.
 onfocusout Fires for the current element with focus immediately after moving focus to another element.
 onkeydown Fires when the user presses a key.
 onkeypress Fires when the user presses an alphanumeric key.
 onkeyup Fires when the user releases a key.
 onload Fires immediately after the browser loads the object.
 onlosecapture Fires when the object loses the mouse capture.
 onmousedown Fires when the user clicks the object with either mouse button.
 onmouseenter Fires when the user moves the mouse pointer into the object.
 onmouseleave Fires when the user moves the mouse pointer outside the boundaries of the object.
 onmousemove Fires when the user moves the mouse over the object.
 onmouseout Fires when the user moves the mouse pointer outside the boundaries of the object.
 onmouseover Fires when the user moves the mouse pointer into the object.
 onmouseup Fires when the user releases a mouse button while the mouse is over the object.
 onmousewheel Fires when the wheel button is rotated.
 onmove Fires when the object moves.
 onmoveend Fires when the object stops moving.
 onmovestart Fires when the object starts to move.
 onpaste Fires on the target object when the user pastes data, transferring the data from the system clipboard to the document.
 onpropertychange Fires when a property changes on the object.
 onreadystatechange Fires when the state of the object has changed.
 onresizeend Fires when the user finishes changing the dimensions of the object in a control selection.
 onresizestart Fires when the user begins to change the dimensions of the object in a control selection.
 onscroll Fires when the user repositions the scroll box in the scroll bar on the object.
 onselect Fires when the current selection changes.
 onselectstart Fires when the object is being selected.
 onunload Fires immediately before the object is unloaded.

onabort

Fires when the user aborts the download of an image.
 Bubbles No
 Cancels Yes
 nur Objekt IMG

onactivate

Fires when the object is set as the active element.
 Bubbles Yes
 Cancels No
 Default action Change activation from the event.fromElement to the event.srcElement.
 A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onafterprint

Fires on the object immediately after its associated document prints or previews for printing.
 Bubbles No
 Cancels No
 window, BODY, FRAMESET

onafterupdate

Fires on a databound object after successfully updating the associated data in the data source object.
 Bubbles Yes
 Cancels No
 A, BDO, BUTTON, CUSTOM, DIV, FRAME, IFRAME, IMG, INPUT type=checkbox, INPUT type=hidden, INPUT type=password, INPUT type=radio, INPUT type=text, LABEL, LEGEND, MARQUEE, RT, RUBY, SELECT, SPAN, TEXTAREA

onbeforeactivate

Fires immediately before the object is set as the active element.
 Bubbles Yes
 Cancels Yes
 A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=reset, INPUT type=password, INPUT type=radio, INPUT



type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onbeforecopy

Fires on the source object before the selection is copied to the system clipboard.

Bubbles Yes

Cancels Yes

A, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL,

DT,

EM, FIELDSET, FORM, hn, I, IMG, LABEL, LEGEND, LI, LISTING, MENU, NOBR, OL, P, PLAINTEXT, PRE, S, SAMP,

SMALL,

SPAN, STRIKE, STRONG, SUB, SUP, TD, TEXTAREA, TH, TR, TT, U, UL

onbeforecut

Fires on the source object before the selection is deleted from the document.

Bubbles Yes

Cancels Yes

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onbeforedeactivate

Fires immediately before the activeElement is changed from the current object to another object in the parent document.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onbeforeeditfocus

Fires before an object contained in an editable element enters a UI-activated state or when an editable container object is control

selected.

Bubbles Yes

Cancels Yes

DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, I, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, NOBR, OBJECT, OL,

P,

PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TD, TEXTAREA, TR, TT, U, UL, VAR, XMP

onbeforepaste

Fires on the target object before the selection is pasted from the system clipboard to the document.

Bubbles Yes

Cancels Yes

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onbeforeprint

Fires on the object before its associated document prints or previews for printing.

Bubbles No

Cancels No

window, BODY, FRAMESET

onbeforeunload

Fires prior to a page being unloaded.

Bubbles No



Cancels Yes
 BODY, FRAMESET, window

onbeforeupdate
 Fires on a databound object before updating the associated data in the data source object.
 Bubbles Yes
 Cancels Yes
 A, BUTTON, DIV, FRAME, IFRAME, IMG, INPUT type=checkbox, INPUT type=hidden, INPUT type=password, INPUT type=radio,
 INPUT type=text, TEXTAREA, LABEL, LEGEND, MARQUEE, SELECT, SPAN, BDO, CUSTOM, RT, RUBY

onblur
 Fires when the object loses the input focus.
 Bubbles No
 Cancels No
 A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onbounce
 Fires when the behavior property of the marquee object is set to "alternate" and the contents of the marquee reach one side of the window.
 Bubbles No
 Cancels Yes
 MARQUEE

oncellchange
 Fires when data changes in the data provider.
 Bubbles Yes
 Cancels No
 APPLET, BDO, OBJECT

onchange
 Fires when the contents of the object or selection have changed.
 Bubbles No
 Cancels Yes
 INPUT type=text, SELECT, TEXTAREA

onclick
 Fires when the user clicks the left mouse button on the object.
 Bubbles Yes
 Cancels Yes
 A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OBJECT, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

oncontextmenu
 Fires when the user clicks the right mouse button in the client area, opening the context menu.
 Bubbles Yes
 Cancels Yes
 A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

oncontrolselect
 Fires when the user is about to make a control selection of the object.
 Bubbles Yes



	<p>Cancels Yes</p> <p>A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP</p>
oncopy	<p>Fires on the source element when the user copies the object or selection, adding it to the system clipboard.</p> <p>Bubbles Yes</p> <p>Cancels Yes</p> <p>A, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, DT,</p>
	<p>EM, FIELDSET, FORM, hn, HR, I, IMG, LEGEND, LI, LISTING, MENU, NOBR, OL, P, PLAINTEXT, PRE, S, SAMP, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TD, TH, TR, TT, U, UL</p>
oncut	<p>Fires on the source element when the object or selection is removed from the document and added to the system clipboard.</p> <p>Bubbles Yes</p> <p>Cancels Yes</p> <p>A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP</p>
ondataavailable	<p>Fires periodically as data arrives from data source objects that asynchronously transmit their data.</p> <p>Bubbles Yes</p> <p>Cancels No</p> <p>APPLET, OBJECT, XML</p>
ondatachanged	<p>Fires when the data set exposed by a data source object changes.</p> <p>Bubbles Yes</p> <p>Cancels No</p> <p>APPLET, OBJECT, XML</p>
ondatacomplete	<p>Fires to indicate that all data is available from the data source object.</p> <p>Bubbles Yes</p> <p>Cancels No</p> <p>APPLET, OBJECT, XML</p>
ondblclick	<p>Fires when the user double-clicks the object.</p> <p>Bubbles Yes</p> <p>Cancels Yes</p> <p>A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OBJECT, OL,</p>
P,	<p>PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP</p>
ondeactivate	<p>Fires when the activeElement is changed from the current object to another object in the parent document.</p> <p>Bubbles Yes</p> <p>Cancels No</p> <p>A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP</p>



XMP

ondrag

Fires on the source object continuously during a drag operation.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, P, PLAINTEXT,

PRE,

Q, S, SAMP, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TR, TT, U, UL, VAR, XMP

ondragend

Fires on the source object when the user releases the mouse at the close of a drag operation.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, P, PLAINTEXT,

PRE,

Q, S, SAMP, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TR, TT, U, UL, VAR, XMP

ondragenter

Fires on the target element when the user drags the object to a valid drop target.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, P, PLAINTEXT,

PRE,

Q, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

ondragleave

Fires on the target object when the user moves the mouse out of a valid drop target during a drag operation.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, P, PLAINTEXT,

PRE,

Q, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TR, TT, U, UL, VAR, XMP

ondragover

Fires on the target element continuously while the user drags the object over a valid drop target.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, P, PLAINTEXT,

PRE,

Q, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TR, TT, U, UL, VAR, XMP

ondragstart

Fires on the source object when the user starts to drag a text selection or selected object.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA,



	TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP
ondrop	<p>Fires on the target object when the mouse button is released during a drag-and-drop operation.</p> <p>Bubbles Yes</p> <p>Cancels Yes</p> <p>A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, P, PLAINTEXT, PRE,</p> <p>Q, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TR, TT, U, UL, VAR, XMP</p>
onerror	<p>Fires when an error occurs during object loading.</p> <p>Bubbles No</p> <p>Cancels Yes</p> <p>IMG, OBJECT, STYLE, window</p>
onerrorupdate	<p>Fires on a databound object when an error occurs while updating the associated data in the data source object.</p> <p>Bubbles Yes</p> <p>Cancels No</p> <p>A, BUTTON, DIV, FRAME, IFRAME, IMG, INPUT type=checkbox, INPUT type=hidden, INPUT type=password, INPUT type=radio, INPUT type=text, TEXTAREA, LABEL, LEGEND, MARQUEE, SELECT, SPAN, BDO, CUSTOM, RT, RUBY</p>
onfilterchange	<p>Fires when a visual filter changes state or completes a transition.</p> <p>Bubbles No</p> <p>Cancels No</p> <p>BDO, BODY, BUTTON, CUSTOM, DIV, FIELDSET, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, MARQUEE,</p>
nextID,	RT, RUBY, SPAN, TABLE, TD, TEXTAREA, TH, TR
onfinish	<p>Fires when marquee looping is complete.</p> <p>Bubbles No</p> <p>Cancels Yes</p> <p>MARQUEE</p>
onfocus	<p>Fires when the object receives focus.</p> <p>Bubbles No</p> <p>Cancels No</p> <p>A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP</p>
onfocusin	<p>Fires for an element just prior to setting focus on that element.</p> <p>Bubbles Yes</p> <p>Cancels No</p> <p>A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=reset, INPUT type=password, INPUT type=radio, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP</p>
onfocusout	<p>Fires for the current element with focus immediately after moving focus to another element.</p> <p>Bubbles Yes</p>



Cancels No
A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=reset, INPUT type=password, INPUT type=radio, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onhelp
Fires when the user presses the F1 key while the browser is the active window.
Bubbles Yes
Cancels Yes
A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onkeydown
Fires when the user presses a key.
Bubbles Yes
Cancels Yes
A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onkeypress
Fires when the user presses an alphanumeric key.
Bubbles Yes
Cancels Yes
A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onkeyup
Fires when the user releases a key.
Bubbles Yes
Cancels No
A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, document, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onlayoutcomplete
Fires when the print or print preview layout process finishes filling the current LayoutRect object with content from the source document.
Bubbles Yes
Cancels Yes
BASE, BASEFONT, BGSOUND, BR, COL, DD, DIV, DL, DT, FONT, HEAD, HR, HTML, HTML Comment, LAYOUTRECT, LI, META, OL, OPTION, P, TITLE, UL

onload
Fires immediately after the browser loads the object.
Bubbles No



Cancels No
 APPLET, BODY, EMBED, FRAME, FRAMESET, IFRAME, IMG, LINK, SCRIPT, window

onlosecapture

Fires when the object loses the mouse capture.

Bubbles No

Cancels No

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BR, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, OPTION, P, PLAINTEXT, PRE, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmousedown

Bubbles Yes

Cancels Yes

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmouseenter

Fires when the user moves the mouse pointer into the object.

Bubbles No

Cancels No

A, ACRONYM, ADDRESS, APPLET, AREA, B, BASE, BASEFONT, BDO, BGSOUND, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, HTML, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB,

SUP,

TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmouseleave

Fires when the user moves the mouse pointer outside the boundaries of the object.

Bubbles No

Cancels No

A, ACRONYM, ADDRESS, APPLET, AREA, B, BASE, BASEFONT, BDO, BGSOUND, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, HTML, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB,

SUP,

TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmousemove

Fires when the user moves the mouse over the object.

Bubbles Yes

Cancels No

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=reset, INPUT type=password, INPUT type=radio, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmouseout

Fires when the user moves the mouse pointer outside the boundaries of the object.

Bubbles Yes

Cancels No

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP



onmouseover

Fires when the user moves the mouse pointer into the object.

Bubbles Yes

Cancels Yes

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmouseup

Fires when the user releases a mouse button while the mouse is over the object.

Bubbles Yes

Cancels Yes

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmousewheel

Fires when the wheel button is rotated.

Bubbles Yes

Cancels Yes

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=reset, INPUT type=password, INPUT type=radio, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onmove

Fires when the object moves.

Bubbles Yes

Cancels No

A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onmoveend

Fires when the object stops moving.

Bubbles Yes

Cancels No

A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onmovestart

Fires when the object starts to move.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP



onpaste

Fires on the target object when the user pastes data, transferring the data from the system clipboard to the document.

Bubbles Yes

Cancels Yes

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OL, P, PLAINTEXT, PRE, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onpropertychange

Fires when a property changes on the object.

Bubbles No

Cancels No

A, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, COMMENT, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LEGEND, LI, LISTING, MAP, MARQUEE, MENU, NOBR, OBJECT, OL, OPTION, P, PLAINTEXT, PRE, S, SAMP, SCRIPT, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onreadystatechange

Fires when the state of the object has changed.

Bubbles No

Cancels No

A, ACRONYM, ADDRESS, APPLET, AREA, B, BASE, BASEFONT, BDO, BGSOUND, BIG, BLOCKQUOTE, BODY, BR, BUTTON, CAPTION, CENTER, CITE, CODE, COL, COLGROUP, COMMENT, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, HEAD, hn, HR, HTML, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LINK, LISTING, MAP,

MARQUEE,

MENU, namespace, nextID, NOBR, NOFRAMES, NOSCRIPT, OBJECT, OL, OPTION, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SCRIPT, SELECT, SMALL, SPAN, STRIKE, STRONG, STYLE, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TITLE, TR, TT, U, UL, VAR, XML, XMP

onreset

Fires when the user resets a form.

Bubbles No

Cancels Yes

FORM

onresize

Fires when the size of the object is about to change.

Bubbles No

Cancels No

A, ADDRESS, APPLET, B, BIG, BLOCKQUOTE, BUTTON, CENTER, CITE, CODE, CUSTOM, DD, DFN, DIR, DIV, DL, DT,

EM,

EMBED, FIELDSET, FORM, FRAME, hn, HR, I, IMG, INPUT type=button, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=reset, INPUT type=submit, INPUT type=text, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PRE, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TEXTAREA, TT, U, UL, VAR, window, XMP

onresizeend

Fires when the user finishes changing the dimensions of the object in a control selection.

Bubbles Yes

Cancels No

A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onresizestart

Fires when the user begins to change the dimensions of the object in a control selection.

Bubbles Yes



Cancels Yes

A, ACRONYM, ADDRESS, APPLET, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CUSTOM, DD, DFN, DIR, DIV, DL, document, DT, EM, EMBED, FIELDSET, FONT, FORM, FRAME, FRAMESET, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MARQUEE, MENU, OBJECT, OL, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, window, XMP

onrowenter

Fires to indicate that the current row has changed in the data source and new data values are available on the object.

Bubbles Yes

Cancels No

APPLET, OBJECT, XML

onrowexit

Fires just before the data source control changes the current row in the object.

Bubbles No

Cancels Yes

APPLET, OBJECT, XML

onrowsdelete

Fires when rows are about to be deleted from the recordset.

Bubbles Yes

Cancels No

APPLET, OBJECT, XML

onrowsinserted

Fires just after new rows are inserted in the current recordset.

Bubbles Yes

Cancels No

APPLET, OBJECT, XML

onscroll

Fires when the user repositions the scroll box in the scroll bar on the object.

Bubbles No

Cancels No

APPLET, BDO, BODY, CUSTOM, DIV, EMBED, MAP, MARQUEE, OBJECT, TABLE, TEXTAREA, window

onselect

Fires when the current selection changes.

Bubbles No

Cancels Yes

BODY, INPUT type=text, TEXTAREA

onselectionchange

Fires when the selection state of a document changes.

Bubbles No

Cancels No

document

onselectstart

Fires when the object is being selected.

Bubbles Yes

Cancels Yes

A, ACRONYM, ADDRESS, AREA, B, BDO, BIG, BLOCKQUOTE, BODY, BUTTON, CAPTION, CENTER, CITE, CODE, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, KBD, LABEL, LI, LISTING, MAP, MARQUEE, MENU, nextID, NOBR, OBJECT, OL, OPTION, P, PLAINTEXT, PRE, Q, RT, RUBY, S, SAMP, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onstart

Fires at the beginning of every loop of the marquee object.

Bubbles No

Cancels No

MARQUEE



onstop

Fires when the user clicks the Stop button or leaves the Web page.
 Bubbles No
 Cancels No
 document

onsubmit

Fires when a FORM is about to be submitted.
 Bubbles No
 Cancels Yes
 FORM

ontimeerror

Fires whenever a time-specific error occurs, usually as a result of setting a property to an invalid value.
 Bubbles No
 Cancels No
 A, ACRONYM, ADDRESS, AREA, B, BIG, BLOCKQUOTE, BUTTON, CAPTION, CENTER, CITE, CODE, DD, DEL, DIR, DIV, DL, DT, EM, FIELDSET, FONT, FORM, hn, HR, I, IFRAME, IMG, INPUT type=button, INPUT type=checkbox, INPUT type=file, INPUT type=hidden, INPUT type=image, INPUT type=password, INPUT type=radio, INPUT type=reset, INPUT type=submit, INPUT type=text, INS, KBD, LEGEND, LI, LISTING, MARQUEE, MENU, OL, OPTION, P, PLAINTEXT, PRE, Q, S, SAMP, selection, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TABLE, TBODY, TD, TEXTAREA, TFOOT, TH, THEAD, TR, TT, U, UL, VAR, XMP

onunload

Fires immediately before the object is unloaded.
 Bubbles No
 Cancels No
 BODY, FRAMESET, window

25.12. *for-Schleife - Schleifenzähler manipulieren*

Der Zähler der for-Schleife ist eine Variable, die - wie andere Variablen auch - innerhalb der for-Schleife manipuliert werden kann: Z.B. Vorzeitiges Setzen des Schleifenzählers auf den Endwert, also vorzeitiges Ende der Schleife.

Egal was innerhalb der Schleife nach for(...) passiert:

for(...) wird immer betreten zum Zweck der Abbruchbedingung. Dabei wird auch der Teil nach der Prüfung der Abbruchbedingung abgearbeitet:

for(...;X00<X01;X00++) also X00++

Damit gilt: Wird innerhalb der Schleife der Zähler auf den Endwert gesetzt, dann wird for(...) aktiviert und somit auch X00++ im obigen und untigen Beispiel
 Der Zähler ist mit Verlassen der Schleife also Endwert +1

```
<html>
<head>
<SCRIPT LANGUAGE='JScript'>
<!--
var X00=0;
var X01=5;
for(X00=0;X00<X01;X00++) // wegen Prüfung                      X00 von 0 bis 4
                        // wegen X00++ nach der Prüfung                      X00 von 0 bis 5
{
    alert('innerhalb for: ' + X00);
    if (X00 ==3)
    {
        X00=X01;
        X00--;
        alert('innerhalb for: Zähler == 3 also Zähler auf Endwert -1 also auf ' + X00);
    }
    alert('nächster Durchlauf for-Schleife: Zähler erhöhen auf ' + (X00+1)
        + ' dann ' + (X00+1) + ' < ' + X01 + ' ergibt ' + ((X00+1)<X01));
}

alert('nach for ' + X00); // 5 angezeigt also wie bei Schleife ohne vorzeitiges Ende
-->
</SCRIPT>
</BODY>
</html>
```



Obige for-Schleife als while, die 1:1 aus for-Schleife umgesetzt wird

```

var X00=0;                // X00=0 Teil im Kopf von for(...)
var X01=5;
while (X00 < X01) // X00 < X01 Teil im Kopf von for(...) wegen Prüfung X00 von 0 bis 4
{
    alert('innerhalb while: ' + X00);
    if (X00 ==3)
    {
        X00=X01;
        X00--;
        alert('innerhalb while: Zähler == 3 also Zähler auf Endwert X01-1 also auf ' + X00);
    }
    alert('nächster Durchlauf while-Schleife: Zähler erhöhen auf ' + (X00+1)
        + ' dann ' + (X00+1) + ' < ' + X01 + ' ergibt ' + ((X00+1)<X01));

    // erst jetzt erhöhen also wie bei for-Schleife
    X00++;                // X00++ Teil im Kopf von for(...)
                        // wegen X00++ X00 von 0 bis 5
}

alert('nach while ' + X00); // 5 angezeigt

```

Obige for-Schleife als optimierte while-Schleife (keine 1:1 Umsetzung)

```

var X00=-1;                // Teil im Kopf von for(...) aber anderer Init-Wert
var X01=5;
while (X00 < X01) // X00 < X01 Teil im Kopf von for(...) wegen Prüfung X00 von 0 bis 4
{
    X00++;                // Teil im Kopf von for(...) X00 von 0 bis 5

    alert('innerhalb while: ' + X00);
    if (X00 ==3)
    {
        X00=X01;
        alert('innerhalb while: Zähler == 3 also Zähler auf Endwert X01 also auf ' + X00);
    }
    alert('nächster Durchlauf while-Schleife: KEIN Zähler erhöhen auf ' + (X00+1)
        + ' sondern nur ' + X00 + ' < ' + X01 + ' ergibt ' + (X00<X01));
}

alert('nach while ' + X00); // 5 angezeigt

```

for-und while-Schleifen ohne alerts:

```

// Deklaration der Variablen
var X00=0;
var X01=5;

// for
for(X00=0;X00<X01;X00++) // wegen Prüfung X00 von 0 bis 4
                        // wegen X00++ nach der Prüfung X00 von 0 bis 5
{
    if (X00 ==3){X00=X01-1;}
}

// while optimiert
X00=-1;
while (X00 < X01) // wegen Prüfung X00 von 0 bis 4
{
    X00++;                // X00 von 0 bis 5
    if (X00 ==3){X00=X01;}
}

```



Damit ist klar: for etwas erspart Schreibarbeit

ABER while hat andere Programmlogik als for !!!

25.13. Funktion

Eine per var Zeiger=new Function(...) erzeugte Funktion

liegt geparkt im Hauptspeicher
wird aus Hauptspeicher aktiviert
dürfte schneller ablaufen als eine im Quelltext codierte und daher mit jedem
Funktionsaufruf zu parsende Funktion

```
function Y1(X0)
{ alert(X0);
  var X1=new Function(",alert('neue Funktion');");
}
```

```
Y1('test');
alert(X1); // X1 nicht definiert, da lokal zu Y1()
```

```
var X00='Y1=new Function("X0","alert(X0 + Y1);");';
// alert(Y1); zeigt Quelltext der Funktion an
// Wichtig: identischer Funktionsbezeichner
// Parameterliste kann abweichen
```

```
function Y1(X0)
{ alert('Y1 alte Funktion ' + Y1);
  eval(X00); // Y1 überschreiben also neu erzeugen
  Y1('Y1 neue Funktion '); // keine Rekursion, sondern neu erzeugte Y1 genau 1x aufrufen
// alte Funktion wird nach Aufruf nie wieder erreicht
}
```

```
Y1('test');
```

Funktion als Zeiger starten

Beispiel 1:

```
function Y1(X0)
{ alert(X0);
  var X1=new Function(",alert('neue Funktion');");
}
```

```
Y1('test');
alert(X1); // X1 nicht definiert, da lokal zu Y1()
```

Beispiel 2:

```
var X00='Y1=new Function("X0","alert(X0 + Y1);");';
// alert(Y1); zeigt Quelltext der Funktion an
// Wichtig: identischer Funktionsbezeichner
// Parameterliste kann abweichen
```

```
function Y1(X0)
{ alert('Y1 alte Funktion ' + Y1);
  eval(X00); // Y1 überschreiben also neu erzeugen
  Y1('Y1 neue Funktion '); // keine Rekursion, sondern neu erzeugte Y1 genau 1x aufrufen
// alte Funktion wird nach Aufruf nie wieder erreicht
}
```

```
Y1('test');
```

Beispiel 3:

```
var x=0;
```

```
function test()
{ x++; alert(x);}
```



```

var ZeigerAufFunktion=test;           // ohne () da sonst sofort ausgeführt wird
eval('ZeigerAufFunktion'+ '()'); // jetzt mit ()

var Kette='ZeigerAufFunktion()';      // auch als Zeichenkettenvariable
eval(Kette);                         // Achtung: ist der Inhalt von ZeigerAufFunktion kein Zeiger,
                                   // so bringt eval einen Scriptfehler !

eval('test()');                      // identisch mit test();

```

Das Testen eines Zeigers auf != null ist mit eval nicht möglich, wenn
eval den Zeiger erst ermittelt aber der Zeigerwert nicht bildbar ist
wobei eval KEINEN Rückkehrcode liefert !!!

Beispiel:

```

var z;                               // Zeiger
var n='testen';                      // Fehler, Es ist 'test' gemeint, jedoch nicht kodiert
                                   // Funktion testen() gibt es nicht

var k="";
var f=false;
var x=0;

function test()
{ x++; alert(x); }

// Funktion testen() ist nicht deklariert

k='z=' + n + ' ';                    // 'z=testen;'

// fehlerhaften Zeiger auf Routine testen() holen --> erzeugt Scriptfehler

eval(k);                             // "testen" wird verwendet auch wenn "test" gemeint ist
                                   // testen() ist nicht deklariert, also Zeigerzuweisung auf z nicht möglich
                                   // eval liefert keinen Rückkehrcode !!!

// wegen Scriptfehler wird nachfolgender Befehl nicht abgearbeitet
if (z != null)
{ alert(); }

```

Damit gilt: Ein Zeigerwert, der per eval zugewiesen wird, muss korrekt sein.
Also ist die Abfrage auf != null nach eval sinnlos !

Alternative sind nicht:

```

eval('if (k != null){ ....});        // ergibt true, da Variable k deklariert und gefüllt ist
                                   // k wird also nicht durch 'z=' + n + ' '; ersetzt !

eval('if ( ('z=' + n + ' '); // anstelle von 'z=' + n + ' ';
                                   // ist auch 'z=testen;' möglich

!= null
)
{ .....}'
);

```

da zuerst der Ausdruck in der if-Anweisung berechnet wird, ergibt sich also
aus 'z=' + n + ' '; der Wert 'z=testen;'

wird 'z=testen;' ausgeführt, entsteht der Scriptfehler, so dass der
Vergleich != null nicht zum Zuge kommt.

```

eval('alert(' + n + '!= null);');    // --- > Scriptfehler, da n als Zeiger interpretiert wird
                                   // wegen Vergleich mit null
                                   // alert nicht ausgeführt

```

25.14. js-Datei dynmatisch einbinden

Der Inhalt der JS-Datei muss Script-konform sein.



```

<html>
<head>
<SCRIPT TYPE="text/javascript" LANGUAGE="JScript" ID="ID_JS"></SCRIPT>

<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
<!--
// Variable oder Funktion aus der JS-Datei hier deklarieren, damit nach fehlerhaftem Laden abfragbar ist
// Es darf keine Variable vor-deklariert werden, die in einer per Mocrsoft Encoder kodierten JS-Datei deklariert wird.
    (im Gegensatz zur nicht kodierten JS-Datei).

var X_Variable_Aus_JS_Datei=null;      // nur zulässig, wenn JS-Datei t.js nicht kodiert ist

var X_Kette="t.js";
ID_JS.src=X_Kette;

// Achtung: JS-Datei erst geladen, wenn dieser Scriptblock komplett abgearbeitet wurde !!
//          Wenn dieser Script-Block eine Referenz auf die JS-Datei besitzt, so wird ein
//          Fehler erzeugt !!
// -->
</script>

<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
<!--
// Extra-Block wegen JS-Datei

// Auf Darstellung der Rekursion in der Variablenprüfung wird hier verzichtet.
// Grössere JS-Dateien benötigen mehr Ladezeit
if (X_Variable_Aus_JS_Datei != null)
{
    alert('JS-Datei ist geladen !');
}
// -->
</script>
</head>
<body>
</body>
</html>

```

25.15. Anzahl der Tage zu einem Monat ermitteln

```

<HTML><HEAD>
<SCRIPT LANGUAGE="JScript">
// ---- Anzahl der Tage zu einem Monat in einem bestimmten Jahr ermitteln
function AnzahlMonatsTageErmitteln(X00,X01)
// X00 Jahr ab 1 und 4-stellig
// X01 Monat ab 0 bis 11
// liefert Anzahl der Tage ab 1
// 0 bei Prameterfehler
{var X02=0;
var X03=0;
var X04=0;
if(X00!=null)
{if(X01!=null)
{// +++++ Annahme: 31 Tage im Monat (alle Monate ausser Februar, April, Juni, September, November)
X02=31;
// +++++ Monate mit 30 Tagen ermitteln
if( (X01==3) // April
||(X01==5) // Juni
||(X01==8) // September
||(X01==10) // November
)
{X02--;}
// +++++ auf Februar wegen Schaltjahr prüfen
if(X01==1)
{// - - - Annahme 28 Tage also kein Schaltjahr
X02=28;
// - - - prüfen auf Schaltjahr
X03=(X00%4); // Modulo
X04=(X00%400); // Modulo
if( (X03==0)

```



```

        ||(X04==0)
        )
        {X02++;}                // Schaltjahr also 29 Tage
    }
}
return X02;
}
alert(AnzahlMonatsTageErmitteln(2007,1));
</SCRIPT>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JScript">
</SCRIPT>
</BODY>
</HTML>

```

25.16. Zeichen kodieren zur Form &#xx; erzeugen

Bsp.: a entspricht a

```

<HTML>
<HEAD>
<SCRIPT language='JAVASCRIPT'>
<!--
var ActionButton_label1='Code dekodiert anzeigen';
var ActionButton_label2='Eingabe nach Code umwandeln';
var code="";
var kette="";
var kette_laenge=0;
var zahler=0;

function KodierenDekodieren()
{
    // prüfen auf leere Eingabe
    if(TextEingabe.value != "")
    {
        // Eingabe nicht leer

        // prüfen auf Dekodierung
        if (ActionButton.value==ActionButton_label1)
        {

            // erst Label ändern
            ActionButton.value=ActionButton_label2;

            // dann TextAusgabe.value in einem neuen Fenster per document.write() dekodiert anzeigen
            Dekodieren();
        }
        else
        {
            // TextEingabe.value kodiert anzeigen
            TextAusgabe.value = Kodieren();

            // Button-Label ändern
            ActionButton.value=ActionButton_label1;

            // Reset-Button unsichtbar machen
            ResetButton.style.visibility='hidden';
        }
    }
    else
    {alert('Eingabe ist leer !');}
}

function Kodieren()
{
    code = "";
    kette = TextEingabe.value;
    kette_laenge = kette.length

```




```

for (zahler=0; zahler<kette_laenge; zahler++)
{
    code = code.concat('&#');
    code = code.concat(kette.charCodeAt(zahler));
    code = code.concat(';');
}

return(code);
}

function Dekodieren()
{
    // Es muss mit document.write() angezeigt werden
    // alert() zeigt nicht dekodiert an !
    document.writeln( '<HTML><BODY>'
        + TextAusgabe.value
        + '<BR>'
        + '<BR>'
        + '<A HREF=javascript:history.back()>zurück</A>'
        + '</BODY></HTML>'
    );
}

function init()
{
    TextAusgabe.value = "";
    TextEingabe.value = "";

    ActionButton.value=ActionButton_label2;

    // Reset-Button sichtbar machen
    ResetButton.style.visibility='visible';
}

<!-->
</SCRIPT>
</HEAD>
<BODY>
<!-- nicht onload='init();' wegen Dekodieren(), das neues Fenster öffnet-->
<!-- und mit Rückkehr dann onload auslöst-->
<H4>Text nach Code</H4>
Code wird per document.write() bzw. vom Browser als normaler Text angezeigt.
<BR>
Dagegen von document.alert() <B>nicht !</B>
<BR>
<BR>
<BR>
<B>Text (beliebig lang):</B>
<INPUT TYPE='Text' ID='TextEingabe' SIZE='120'>
<BR>
<INPUT TYPE='button' ID='ActionButton' VALUE="" onclick='KodierenDekodieren();'>
<INPUT TYPE='button' ID='ResetButton' VALUE='Reset' onclick='init();'>
<br>
<B>in Code:</B>
<br>
<TEXTAREA ID="TextAusgabe" ROWS='40' COLS='120' READONLY></TEXTAREA>
<SCRIPT language='JAVASCRIPT'>
<!--
init();
// nicht BODY onload='init();' wegen Dekodieren(), das neues Fenster öffnet
// und mit Rückkehr dann onload auslöst
-->
</SCRIPT>
</BODY>
</HTML>

xanti

```



25.17. ASCII-Tabelle anzeigen

gepackter Quellcode:

```
<BODY>
<a href="javascript:function C(v){return '<td>'
+v+'</td><td>' + ((v>>4).toString(16)+(v&15).toString(16)).toUpperCase()+'</td><td bgcolor=DDDDDD><b>&'+'#'
+v+'</b></td>';} var c=4,b=Math.ceil(224/c),a='<table border=0><tr>';for(j=0;j<c;j++){
a+='<td>DEC</td><td>HEX</td><td><b>ASC</b></td>';} a+='</tr>';for(i=33;i<33+b;i++){ a+='<tr>';
for(j=0;j<c;j++){ t=i+(j*b);if(t<=255)a+=C(t);} a+='</tr>';} a+='</table>';var W=open("",'width=500,height=600,
left=0,top=0,resizable,scrollbars');W.document.writeln(a);">Klick</a>
<BODY>
```

aufbereiteter Quellcode

```
<BODY>
<a href="javascript: function C(v)
{
return
'<td>'
+v+'</td><td>'
+((v>>4).toString(16)+(v&15).toString(16)).toUpperCase()
+'</td><td bgcolor=DDDDDD><b>&'
+'#'
+v
+'</b></td>';
}
var c=4,b=Math.ceil(224/c),a='<table border=0><tr>';
for(j=0;j<c;j++)
{
a+='<td>DEC</td><td>HEX</td><td><b>ASC</b></td>';
}
a+='</tr>';
for(i=33;i<33+b;i++)
{
a+='<tr>';
for(j=0;j<c;j++)
{
t=i+(j*b);if(t<=255)a+=C(t);
}
a+='</tr>';
}
a+='</table>';
var W=open("",'width=500,height=600,left=0,top=0,resizable,scrollbars');
W.document.writeln(a);">Klick</a>
<BODY>
```

25.17. Primzahlen ermitteln

Die Primzahlenermittlung kann z.B. als Leistungsmesser einer CPU unter z.B. 32-Bit- Windows benutzt werden (Anzahl der Primzahlen pro Zeiteinheit).

Achtung: Wenn der Parser der Script-Maschine und diese selbst nicht Prozesse auf vorhandene Mehrkern-CPU verteilen kann, wird diese wohl ähnlich

schlecht abschneiden, wie Einkern-CPU bei gleichem Motherboard und CPU-Takt etc..

Die Primzahlenermittlung benötigt sehr intensiv CPU-Ressourcen.

25.17.1 Primzahlen ermitteln langsame Variante

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
// Primzahlen LANGSAM berechnen ab 2 bis inklusive AnzahlAb3 und Primzahlen anzeigen per alert
// Berechnung anhand des Sieb des Eratosthenes (276-195 vor Christus)
// +++++ interne Variablen
var ErsteZahl=0; // erste Zahl > 1
// ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
// 2 ist die erste Primzahl
var AnzahlAb3=0; // letzte Zahl >=ErsteZahl
// ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
// ist die Anzahl der Zahlen ab 3 ganz positiv, aus denen die Primzahlen ermittelt werden
```



```

var SiebAlsFeld;           // Sieb als Feld, hat Index ab 0 bis inklusive AnzahlAb3
var StartIndex=0;         // NaechstesAbKlappernDesSiebes() und SiebAbKlappern(): Startindex der aktuellen Primzahlermittlung
var Index=0;              // SiebAbKlappern(): Hilfsindex
var IndexQuadrat=0;       // SiebAbKlappern(): Arbeitswert
var Flag1=false;         // NaechstesAbKlappernDesSiebes() und SiebAbKlappern(): Arbeitsflag
var Flag2=false;         // SiebAbKlappern(): Arbeitsflag
var Flag3=false;         // SiebAbKlappern(): Arbeitsflag
var TimeoutID=0;          // SiebAbKlappern(): Timeout-ID für Rekursion, muss 0 sein
var TimeoutWert=0;        // SiebAbKlappern(): Millisekunden >=200 der Rekursion
var PrimZahlenErmittlungAktiv=false; // true so Primzahlen-Ermittlung aktiv, muss false sein

// ----- Sieb abklappern
function SiebAbKlappern() // interne Routine, rekursiv
                        // wird aufgerufen von NaechstesAbKlappernDesSiebes()
{
    // ++++ prüfen ob Primzahlen-Ermittlung aktiv ist
    if(PrimZahlenErmittlungAktiv)
    {
        // ++++ Feldelement an Position laut StartIndex immer als gerahmt markieren
        SiebAlsFeld[StartIndex]=1;
        // ++++ jeden Nachfolger im Abstand laut StartIndex zum Vorgänger als gestrichen markieren
        Index=StartIndex; // StartIndex
        Index+=StartIndex; // nächsten Nachfolger
        while(Index<=AnzahlAb3) // solange wie Nachfolger existent
        {
            SiebAlsFeld[Index]=2; // Nachfolger als gestrichen markieren
            Index+=StartIndex; // nächsten Nachfolger
        }
        // ++++ SiebAlsFeld abklappern und 1. Feldelement mit Wert == 0 suchen als nächsten StartIndex
        StartIndex=-1;
        do
        {
            // ----- nächstes Feldelement
            StartIndex++;
            // ----- merken ob Feldelement mit Wert == 0
            Flag1=(SiebAlsFeld[StartIndex]==0);
            // ----- merken ob letzter Index erreicht worden ist
            Flag2=(StartIndex==AnzahlAb3);
            // ----- Abbruch wenn
            //         Feldelement mit Wert == 0 gefunden
            //         und/oder StartIndex == AnzahlAb3
            Flag3=(Flag1||Flag2); // true so Abbruch
        }
        while(!Flag3); // solange suchen wie kein Abbruch erkannt
        // ++++ ermitteln ob nächster gefundener StartIndex gültig ist
        // ----- Quadrat des gefundenen StartIndex prüfen
        if(Flag1)
        {
            IndexQuadrat=StartIndex*StartIndex; // Quadrat
            Flag1=(IndexQuadrat<=AnzahlAb3); // true so gültig
        }
        // ----- ermitteln ob Feldelement an StartIndex mit Wert 0 ist
        Flag1=(SiebAlsFeld[StartIndex]==0);
        // ++++ IMMER rekursieren werden, da alle Schleifen sonst die Systemuhr nachgehen lassen !
        TimeoutID=window.setTimeout('NaechstesAbKlappernDesSiebes()',TimeoutWert);
    }
}

// ----- Nächstes Abklappern des Siebes bzw. Primzahlen per alert Anzeigen
function NaechstesAbKlappernDesSiebes() // interne Routine, rekursiv
                                        // wird durch SiebAbKlappern()
                                        // aber Erster Aufruf durch InitUndStartDerPrimzahlErmittlung()

{
    // ++++ prüfen ob Primzahlen-Ermittlung aktiv ist
    if(PrimZahlenErmittlungAktiv)
    {
        // ++++ wenn nächster StartIndex gefunden so Primzahl ermitteln
        if(Flag1){SiebAbKlappern();} // SiebAbKlappern() ruft NaechstesAbKlappernDesSiebes() per Rekursion auf
                                    // belegt Flag1
    }
    else{PrimzahlenAnzeigen(true);} // alle Primzahlen wurden ermittelt, also anzeigen
                                    // true oder false ist egal da bereits alle Primzahlen
    ermittelt wurden
}

// ----- Anzeige der ermittelten Primzahlen

```



```
// optional bereits aktive Primzahlen-Ermittlung stoppen
function PrimzahlenAnzeigen(X00) // auch vom Programmierer aufrufbar, wiederholbar
// X00 true so eine bereits aktive Primzahl-Ermittlung stoppen
// false so confirm-Abfrage, ob eine bereits aktive Primzahl-Ermittlung gestoppt werden soll
// nur ausgewertet wenn in bereits aktiver Primzahl-Ermittlung noch nicht alle Primzahlen ermittelt wurden
{var X01=0;
var X02="";
var X03=0;
var X04=false;
// +++++ prüfen ob Primzahlen-Ermittlung aktiv ist
if(PrimZahlenErmittlungAktiv)
{ // +++++ prüfen ob alle Primzahlen ermittelt wurden
if(!Flag1)
{ // +++++ Primzahlen-Ermittlung als inaktiv setzen
PrimZahlenErmittlungAktiv=false; // true so Primzahlen-Ermittlung aktiv
// +++++ Timeout-ID löschen
TimeoutID=0;
// +++++ Alle Feldelemente mit Wert <=1 (also weder gerahmt noch gestrichen): deren Indexe sind Primzahlen
for(X01=0;X01<=AnzahlAb3;X01++)
{ // ----- Indexbereich prüfen
if(X01>=ErsteZahl)
{ if(X01<=AnzahlAb3)
{ // ----- Feldelement-Wert prüfen
if(SiebAlsFeld[X01]<=1)
{ X03++;
if(X03>1){ X02+=", ";
X02+=X01.toString();
}
}
}
}
}
}
}
alert( 'Es wurden alle Primzahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3 + ' ermittelt !\n\n'
+ 'Primzahlen-Anzahl = ' + X03 + '\n'
+ 'Primzahlen sind:\n'
+ X02
);
}
else // in der aktiven Primzahlen-Ermittlung wurden noch nicht alle Primzahlen ermittelt
{ alert( 'Es ist gerade im Hintergrund eine Primzahlen-Ermittlung aktiv (' + ErsteZahl + ' bis ' + AnzahlAb3 + ') !\n'
+ 'Daher können keine Ergebnisse angezeigt werden !'
);
// +++++ Annahme: aktive Primzahl-Ermittlung stoppen
X04=true;
// +++++ prüfen ob confirm-Abfrage erfolgen soll
if(!X00){ X04=confirm( 'Im Hintergrund aktive Primzahlen-Ermittlung von ' + ErsteZahl + ' bis ' + AnzahlAb3
+ ' stoppen ?'); }
// +++++ aktive Primzahl-Ermittlung stoppen
if(X04){ PrimzahlenErmittlungStoppen(); }
}
}
else{ alert('Es ist keine Primzahlen-Ermittlung im Hintergrund aktiv !'); }
}

// ----- Primzahlen-Ermittlung stoppen
function PrimzahlenErmittlungStoppen() // vom Programmierer aufrufbar
{ // +++++ prüfen ob bereits eine Primzahlen-Ermittlung läuft
if(PrimZahlenErmittlungAktiv)
{ alert( 'Die gerade im Hintergrund aktive Primzahlen-Ermittlung von ' + ErsteZahl + ' bis ' + AnzahlAb3
+ ' wurde gestoppt !\n'
+ 'Daher können keine Ergebnisse angezeigt werden !'
);
}
// +++++ Primzahlen-Ermittlung als inaktiv setzen
PrimZahlenErmittlungAktiv=false; // true so Primzahlen-Ermittlung aktiv
// +++++ Rekursion der aktuellen Primzahlen-Ermittlung stoppen
if(TimeoutID!=0){ window.clearTimeout(TimeoutID); }
TimeoutID=0;
// +++++ simulieren dass nicht alle Primzahlen ermittelt wurden
Flag1=true;
}
```



```

}

// ----- init und Start der Primzahlen-Ermittlung
function InitUndStartDerPrimzahlErmittlung(X00,X01,X02)

// vom Programmierer aufrufbar
// wiederholbar aber eine aktive Primzahlen-
Ermittlung wird gestoppt
// X00   erste Zahl > 1  ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
//                               wenn < 2 so automatisch 2
// X01   letzte Zahl >=X00 ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
//                               wenn < 2 so automatisch 2
//                               wenn < X00 so automatisch X00
// X02   Millisekunden >=200 der Rekursion,
//                               je höher um so langsamer aber um so ressourcen-schonender ist die Primzahlen-Ermittlung
{var X03=0;
var X04=false;
var X05=0;
// +++++ prüfen ob bereits eine Primzahlen-Ermittlung läuft, wenn ja dann stoppen
if(PrimZahlenErmittlungAktiv){PrimZahlenErmittlungStoppen();}
// +++++ Parameter prüfen
if(X00!=null)
if(X02!=null)
{
// +++++ erste Zahl prüfen
ErsteZahl=Math.abs(X00);
if(ErsteZahl<2){ErsteZahl=2;}
// +++++ letzte Zahl prüfen
AnzahlAb3=Math.abs(X01);
if(AnzahlAb3<2){AnzahlAb3=2;}
if(AnzahlAb3<ErsteZahl){AnzahlAb3=ErsteZahl;}
// +++++ Timeoutwert prüfen
TimeoutWert=X02;
if(TimeoutWert<1){TimeoutWert=1;}
// +++++ Anzahl der Primzahlen aus dem Bereich von 2 bis AnzahlAb3 näherungsweise ermitteln
// ----- Logarithmus zur Basis e (Eulersche Zahl)
X03=Math.log(AnzahlAb3);           // ln AnzahlAb3
// ----- wenn AnzahlAb3 <= 55
if(AnzahlAb3<=55)
{X05=AnzahlAb3;
X05/=X03;
}
else // > 55
{X05=4;X05/=X03;           // 4 / ln AnzahlAb3
X03=1;X03=X05;           // 1 - (4 / ln AnzahlAb3)
X05=Math.sqrt(X03);       // Quadratwurzel aus (1 - (4 / ln AnzahlAb3))
X03=1;X03=X05;           // 1 - (Quadratwurzel aus (1 - (4 / ln AnzahlAb3)))
X05=AnzahlAb3;
X05/=2;
X05*=X03;
}
X03=Math.floor(X05);
// +++++ Abfrage ob Primzahlen-Ermittlung starten soll
X04=confirm( 'Primzahlen-Ermittlung aus den natürlichen Zahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3 + ':\n\n'
+ ' - Es sind ca. ' + X03 + ' Primzahlen zu ermitteln !\n\n'
+ ' - wird ressourcen-schonend im Hintergrund (parallel) vollzogen!\n'
+ ' - dauert um so länger,\n'
+ ' * je grösser der Bereich der natürlichen Zahlen ist,\n'
+ '   aus denen die Primzahlen ermittelt werden sollen,\n'
+ ' * und je geringer die CPU-Geschwindigkeit ist.\n'
+ ' - Die Dauer der Ermittlung im Hintergrund ist nicht die optimal kürzeste,\n'
+ '   da mit Timer gearbeitet wird (ohne Timer würde die Systemuhr falsch gehen !).\n'
+ ' - Am Ende der Ermittlung werden die Primzahlen per alert() angezeigt.\n\n\n'
+ 'Primzahlen-Ermittlung aus den natürlichen Zahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3
+ ' + ' aktivieren ?'
);
if(X04)
{
// +++++ Feldinit
SiebAlsFeld=new Array();
SiebAlsFeld[0]=2;           // SiebAlsFeld[0] immer gestrichen
SiebAlsFeld[1]=2;           // SiebAlsFeld[1] immer gestrichen

```



```

SiebAlsFeld[2]=1; // SiebAlsFeld[2] immer gerahmt
// alle anderen mit Wert 0 belegen, also markieren als weder gerahmt noch gestrichen
for(X03=3;X03<=AnzahlAb3;X03++){SiebAlsFeld[X03]=0;}
// +++++ Primzahlen ermitteln
// ----- erste Primzahl ist 2 in SiebAlsFeld[StartIndex] also ist 2 der erste gültige StartIndex
StartIndex=2;
Flag1=true; // StartIndex 2 ist gültig
PrimZahlenErmittlungAktiv=true; // true so Primzahlen-Ermittlung aktiv
NaechstesAbKlappernDesSiebes();
alert( 'Die Primzahlen-Ermittlung aus den natürlichen Zahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3
+ ' wurde aktiviert\n'
+ ' und erfolgt ressourcen-schonend im Hintergrund !\n\n'
+ 'Am Ende der Ermittlung werden die Primzahlen per alert() angezeigt.'
);
}
}
}

// Start der Primzahlermittlung, wiederholbar
InitUndStartDerPrimzahlErmittlung(-1,-12000,20); // wiederholbar (aktuelle Ermittlung wird abgebrochen)
// X00 erste Zahl > 1 ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// negative Zahl wird automatisch positiv
// X01 letzte Zahl >=X00 ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// wenn < X00 so automatisch X00
// negative Zahl wird automatisch positiv
// X02 Millisekunden >=1 der Rekursion,
// je höher um so langsamer aber um so ressourcen-schonender ist die Primzahlen-Ermittlung
//alert();
//InitUndStartDerPrimzahlErmittlung(-1,10,20); // wiederholbar (aktuelle Ermittlung wird abgebrochen)
// X00 erste Zahl > 1 ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// negative Zahl wird automatisch positiv
// X01 letzte Zahl >=X00 ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// wenn < X00 so automatisch X00
// negative Zahl wird automatisch positiv
// X02 Millisekunden >=1 der Rekursion,
// je höher um so langsamer aber um so ressourcen-schonender ist die Primzahlen-Ermittlung
</SCRIPT>
</BODY>
</HTML>

```

25.17.1 Primzahlen ermitteln schnelle Variante

```

<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
// Primzahlen SCHNELL berechnen ab 2 bis inklusive AnzahlAb3 und Primzahlen anzeigen per alert
// verkürzte Berechnung anhand des Sieb des Eratosthenes (276-195 vor Christus)
// Berechnung:
// Man schreibe die Zahlen ab 2 bis gewünschten Endwert auf
// dann pro Zahl ab 2
// alle ganzzahligen Vielfachen der Zahlen streichen
// nach allen Streichungen bleiben die Primzahlen übrig
// Der Trick:
// Die natürlichen Zahlen bauen sich auf aus der Vorgängererhöhung um 1 als Nachfolger, also lückenlos
// und damit auch alle Primzahlen erfassend (Alle Zahlen sind also lückenlos aufsteigend
// sortiert).
// Mit dem Streichen aller Vielfachen fallen also alle ungestrichenen Zahlen heraus (Sieb),
// also alle Zahlen, die selbst kein Vielfaches sind (also nicht durch 2, 3, 4 etc. teilbar)
// damit Primzahl sein müssen
// 1 ist per Definition keine Primzahl
// Erste Primzahl ist 2
//
// Bsp: 2 3 4 5 6 7 8 9 10 11

```



```

//          12 13 14 15 16 17 18 19 20 21
//          für 2 streichen 2*2 also 4
//
//          2*3 also 6
//          2*4 also 8
//          2*5 also 10
//          2*6 also 12
//
//          ....
//          3 streichen 2*3 also 6          ist schon gestrichen
//          3*3 also 9
//          3*4 also 12 ist schon gestrichen
//          3*5 also 15
//          ....
//          4 streichen          ist schon gestrichen
//          da 4 ein Vielfaches von 2
//          und für 2 schon alle Vielfachen gestrichen sind
//          sind also für 4 auch alle Vielfachen gestrichen, die auch Vielfache von 2 sind
//          5 streichen 5*2 also 10          ist schon gestrichen
//          5*3 also 15 ist schon gestrichen
//          ....
//          ....
//          6 streichen          ist schon gestrichen
//          da 6 ein Vielfaches von 3
//          und für 3 schon alle Vielfachen gestrichen sind
//          sind also für 6 auch alle Vielfachen gestrichen, die auch Vielfache von 3 sind
//          ....
//          Wegen aufsteigender Sortierung der Zahlen sind also alle Vielfache, die selbst Vielfaches
//          eines kleineren Zahl oder eines kleineren Vielfaches sind, bereits gestrichen,
//          da die kleinere Zahl bzw. das kleinere Vielfache vorher alle Vielfache erwischte
//          und gestrichen hat.
//          Falls also eine Zahl schon gestrichen ist, dann nicht nochmal streichen.
//
//          Die Berechnung entspricht verkürzt dem Sieb des Eratosthenes (276-195 vor Christus)

// +++++ interne Variablen
var ErsteZahl=0;          // erste Zahl > 1
//                          // ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
//                          // 2 ist die erste Primzahl
var AnzahlAb3=0;          // letzte Zahl >=ErsteZahl
//                          // ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
//                          // ist die Anzahl der Zahlen ab 3 ganz positiv, aus denen die Primzahlen ermittelt werden
var SiebAlsFeld;          // Sieb als Feld, hat Index ab 0 bis inklusive AnzahlAb3
var X_IndexAussen=0;      // Index 1 der Rekursion, Wert egal
var X_IndexInnen=0;      // Index 2 der Rekursion, Wert egal
var AlleStreichungenErfolgt=false; // true so alle Streichungen vollzogen
var TimeoutID=0;          // Timeout-ID der Rekursion, muss 0 sein
var TimeoutWert=0;        // Millisekunden >=200 der Rekursion, Wert egal
var PrimZahlenErmittlungAktiv=false; // true so Primzahlen-Ermittlung aktiv, muss false sein

// ----- Sieb abklappern
function AlleVielFachenStreichen()          // interne Routine, rekursiv
//                                          // wird aufgerufen von StartAlleVielFachenStreichen()

{
    // +++++ prüfen ob Primzahlen-Ermittlung aktiv ist
    if(PrimZahlenErmittlungAktiv)
    {
        for(X_IndexInnen=2*X_IndexAussen;X_IndexInnen<=AnzahlAb3;X_IndexInnen+=X_IndexAussen) // streichen von 4, 6, 8 ...
        {
            if(SiebAlsFeld[X_IndexInnen]){SiebAlsFeld[X_IndexInnen]=false;}
            // wenn nicht gestrichen, dann als gestrichen markieren
            // true so nicht gestrichen
        }
        // +++++ IMMER rekursieren werden, da alle Schleifen sonst die Systemuhr nachgehen lassen !
        TimeoutID=window.setTimeout('StartAlleVielFachenStreichen()',TimeoutWert);
    }
}

// ----- Nächstes Abklappern des Siebes bzw. Primzahlen per alert Anzeigen
function StartAlleVielFachenStreichen()     // interne Routine, rekursiv
//                                          // wird durch AlleVielFachenStreichen()
//                                          // aber Erster Aufruf durch InitUndStartDerPrimzahlErmittlung()

{
    // +++++ prüfen ob Primzahlen-Ermittlung aktiv ist
    if(PrimZahlenErmittlungAktiv)
    {
        // +++++ nächster Index also Zahl, deren Vielfache zu streichen sind
    }
}

```



```

X_IndexAussen++;
// +++++ merken ob alle Streichungen vollzogen wurden
AlleStreichungenErfolgt=(X_IndexAussen>AnzahlAb3)
// +++++ wenn alle Streichungen vollzogen so Primzahlen anzeigen; sonst weiterstreichen
if(AlleStreichungenErfolgt){PrimzahlenAnzeigen(true);}

// alle Primzahlen wurden ermittelt, also anzeigen
// true oder false ist egal da bereits alle Primzahlen
ermittelt wurden
else // Streichungen noch offen
{
  // +++++ prüfen ob Zahl, deren Vielfaches gestrichen werden soll, nicht bereits gestrichen ist
  if(SiebAlsFeld[X_IndexAussen]){AlleVielFachenStreichen();} // true so nicht gestrichen
  else{StartAlleVielFachenStreichen();} // nächste Zahl, deren Vielfache gestrichen werden sollen, verarbeiten
  // Rekursion ohne Timer
}
}
}

// ----- Anzeige der ermittelten Primzahlen
// optional bereits aktive Primzahlen-Ermittlung stoppen
function PrimzahlenAnzeigen(X00) // auch vom Programmierer aufrufbar, wiederholbar
// X00 true so eine bereits aktive Primzahl-Ermittlung stoppen
// false so confirm-Abfrage, ob eine bereits aktive Primzahl-Ermittlung gestoppt werden soll
// nur ausgewertet wenn in bereits aktiver Primzahl-Ermittlung noch nicht alle Primzahlen ermittelt wurden
{
  var X01=0;
  var X02="";
  var X03=0;
  var X04=false;
  // +++++ prüfen ob Primzahlen-Ermittlung aktiv ist
  if(PrimZahlenErmittlungAktiv)
  {
    // +++++ prüfen ob alle Primzahlen ermittelt wurden
    if(AlleStreichungenErfolgt)
    {
      // +++++ Primzahlen-Ermittlung als inaktiv setzen
      PrimZahlenErmittlungAktiv=false; // true so Primzahlen-Ermittlung aktiv
      // +++++ Timeout-ID löschen
      TimeoutID=0;
      // +++++ Alle Feldelemente mit Wert <=1 (also weder gerahmt noch gestrichen): deren Indexe sind Primzahlen
      for(X01=0;X01<=AnzahlAb3;X01++)
      {
        // ----- Indexbereich prüfen
        if(X01>=ErsteZahl)
        {
          if(X01<=AnzahlAb3)
          {
            // ----- Feldelement-Wert prüfen auf ungestrichen
            if(SiebAlsFeld[X01]) // true so nicht gestrichen
            {
              X03++;
              if(X03>1){X02+=',';}
              X02+=X01.toString();
            }
          }
        }
      }
    }
    alert( 'Es wurden alle Primzahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3 + ' ermittelt !\n\n'
      + 'Primzahlen-Anzahl = ' + X03 + '\n'
      + 'Primzahlen sind\n'
      + X02
    );
  }
  else // in der aktiven Primzahlen-Ermittlung wurden noch nicht alle Primzahlen ermittelt
  {
    alert( 'Es ist gerade im Hintergrund eine Primzahlen-Ermittlung aktiv ( ' + ErsteZahl + ' bis ' + AnzahlAb3 + ' ) !\n'
      + 'Daher können keine Ergebnisse angezeigt werden !'
    );
    // +++++ Annahme: aktive Primzahl-Ermittlung stoppen
    X04=true;
    // +++++ prüfen ob confirm-Abfrage erfolgen soll
    if(!X00){X04=confirm( 'Im Hintergrund aktive Primzahlen-Ermittlung von ' + ErsteZahl + ' bis ' + AnzahlAb3
      + ' stoppen ?');}
    // +++++ aktive Primzahl-Ermittlung stoppen
    if(X04){PrimzahlenErmittlungStoppen();}
  }
  else{alert('Es ist keine Primzahlen-Ermittlung im Hintergrund aktiv !');}
}

```




```

}

// ----- Primzahlen-Ermittlung stoppen
function PrimzahlenErmittlungStoppen() // vom Programmierer aufrufbar
{
  // +++++ prüfen ob bereits eine Primzahlen-Ermittlung läuft
  if(PrimZahlenErmittlungAktiv)
  {
    alert( 'Die gerade im Hintergrund aktive Primzahlen-Ermittlung von ' + ErsteZahl + ' bis ' + AnzahlAb3
      + ' wurde gestoppt !\n'
      + 'Daher können keine Ergebnisse angezeigt werden !'
    );
  }
  // +++++ Primzahlen-Ermittlung als inaktiv setzen
  PrimZahlenErmittlungAktiv=false; // true so Primzahlen-Ermittlung aktiv
  // +++++ Rekursion der aktuellen Primzahlen-Ermittlung stoppen
  if(TimeoutID!=0){window.clearTimeout(TimeoutID);}
  TimeoutID=0;
  // +++++ simulieren dass nicht alle Primzahlen ermittelt wurden
  AlleStreichungenErfolgt=false;
}

// ----- init und Start der Primzahlen-Ermittlung
function InitUndStartDerPrimzahlErmittlung(X00,X01,X02)

// vom Programmierer aufrufbar
// wiederholbar aber eine aktive Primzahlen-
Ermittlung wird gestoppt
// X00 erste Zahl > 1 ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// X01 letzte Zahl >=X00 ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// wenn < X00 so automatisch X00
// X02 Millisekunden >=200 der Rekursion,
// je höher um so langsamer aber um so ressourcen-schonender ist die Primzahlen-Ermittlung
{
  var X03=0;
  var X04=false;
  var X05=0;
  // +++++ prüfen ob bereits eine Primzahlen-Ermittlung läuft, wenn ja dann stoppen
  if(PrimZahlenErmittlungAktiv){PrimzahlenErmittlungStoppen();}
  // +++++ Parameter prüfen
  if(X00!=null)
  {
    if(X02!=null)
    {
      // +++++ erste Zahl prüfen
      ErsteZahl=Math.abs(X00);
      if(ErsteZahl<2){ErsteZahl=2;}
      // +++++ letzte Zahl prüfen
      AnzahlAb3=Math.abs(X01);
      if(AnzahlAb3<2){AnzahlAb3=2;}
      if(AnzahlAb3<ErsteZahl){AnzahlAb3=ErsteZahl;}
      // +++++ Timeoutwert prüfen
      TimeoutWert=X02;
      if(TimeoutWert<1){TimeoutWert=1;}
      // +++++ Anzahl der Primzahlen aus dem Bereich von 2 bis AnzahlAb3 näherungsweise ermitteln
      // ----- Logarithmus zur Basis e (Eulersche Zahl)
      X03=Math.log(AnzahlAb3); // ln AnzahlAb3
      // ----- wenn AnzahlAb3 <= 55
      if(AnzahlAb3<=55)
      {
        X05=AnzahlAb3;
        X05/=X03;
      }
      else // > 55
      {
        X05=4;X05/=X03; // 4 / ln AnzahlAb3
        X03=1;X03-=X05; // 1 - (4 / ln AnzahlAb3)
        X05=Math.sqrt(X03); // Quadratwurzel aus (1 - (4 / ln AnzahlAb3))
        X03=1;X03-=X05; // 1 - (Quadratwurzel aus (1 - (4 / ln AnzahlAb3)))
        X05=AnzahlAb3;
        X05/=2;
        X05*=X03;
      }
      X03=Math.floor(X05);
      // +++++ Abfrage ob Primzahlen-Ermittlung starten soll

```



```

X04=confirm( 'Primzahlen-Ermittlung aus den natürlichen Zahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3 + ':\n\n'
+ ' - Es sind ca. ' + X03 + ' Primzahlen zu ermitteln !\n\n'
+ ' - wird ressourcen-schonend im Hintergrund (parallel) vollzogen!\n'
+ ' - dauert um so länger,\n'
+ ' * je grösser der Bereich der natürlichen Zahlen ist,\n'
+ ' * aus denen die Primzahlen ermittelt werden sollen,\n'
+ ' * und je geringer die CPU-Geschwindigkeit ist.\n'
+ ' - Die Dauer der Ermittlung im Hintergrund ist nicht die optimal kürzeste !\n\n'
+ ' - Am Ende der Ermittlung werden die Primzahlen per alert() angezeigt.\n\n'
+ 'Primzahlen-Ermittlung aus den natürlichen Zahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3
+ ' aktivieren ?'
);
if(X04)
{
+++++ Feldinit
SiebAlsFeld=new Array();
SiebAlsFeld[0]=false; // SiebAlsFeld[0] immer gestrichen, falso so gestrichen
SiebAlsFeld[1]=false; // SiebAlsFeld[1] immer gestrichen
// alle anderen mit Wert 0 also als ungestrichen belegen
for(X03=2;X03<=AnzahlAb3;X03++){SiebAlsFeld[X03]=true;}
+++++ Primzahlen ermitteln
// ----- erste Zahl für die Streichungen der vielfachen erfolgen soll ist 2
X_IndexAussen=1; // 2-1
PrimZahlenErmittlungAktiv=true; // true so Primzahlen-Ermittlung aktiv
StartAlleVielFachenStreichen();
alert( 'Die Primzahlen-Ermittlung aus den natürlichen Zahlen von ' + ErsteZahl + ' bis ' + AnzahlAb3
+ ' wurde aktiviert!\n'
+ ' und erfolgt ressourcen-schonend im Hintergrund !\n\n'
+ 'Am Ende der Ermittlung werden die Primzahlen per alert() angezeigt.'
);
}
}
}
}

// Start der Primzahlermittlung, wiederholbar
InitUndStartDerPrimzahlErmittlung(-1,-12000,20); // wiederholbar (aktuelle Ermittlung wird abgebrochen)
// X00 erste Zahl > 1 ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// negative Zahl wird automatisch positiv
// X01 letzte Zahl >=X00 ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// wenn < X00 so automatisch X00
// negative Zahl wird automatisch positiv
// X02 Millisekunden >=1 der Rekursion,
// je höher um so langsamer aber um so ressourcen-schonender ist die Primzahlen-Ermittlung
//alert();
//InitUndStartDerPrimzahlErmittlung(-1,10,20); // wiederholbar (aktuelle Ermittlung wird abgebrochen)
// X00 erste Zahl > 1 ganz und positiv, ab inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// negative Zahl wird automatisch positiv
// X01 letzte Zahl >=X00 ganz und positiv, bis zu inklusive der die Primzahlen ermittelt werden sollen
// wenn < 2 so automatisch 2
// wenn < X00 so automatisch X00
// negative Zahl wird automatisch positiv
// X02 Millisekunden >=1 der Rekursion,
// je höher um so langsamer aber um so ressourcen-schonender ist die Primzahlen-Ermittlung
</SCRIPT>
</BODY>
</HTML>

```

25.18. *Formatierte HTML-Text-Erzeugung*

```

<HTML>
<HEAD>
<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
// ++++++ Routinen zur formatierten HTML-Texterzeugung ++++++
// per FONT-Tag oder DIV-Tag
// Routinen dienen z.B. als Ersatz für HTML-Tags wie <Hx>, da z.B. <Hx> unter Windows 98 und
// unter Windows XP verschieden dargestellt wird bei
gleicher

```



```

//
var d0139a=15;      // Standard-FontSize in pt
var d0139b='black'; // Standard-Fontfarbe
var d0139c='Times'; // Standard-Font-Family
var d0139d=624;     // Fenster Standardhöhe bei 1024x768
                    // in Pixel, wenn Taskleiste am unteren Bildrand
                    //   Windows 98  633
                    //   Windows XP  624 für IE 6.0 SP2
                    // Die Standardwerte ändern sich je nach Browserversion. Daher wurde der BODY mit
STYLE="overflow:visible"
                    //   instanziert, da damit bei zu grossem Bild des Hintergrundes die Scrollleisten ausgeblendet werden.
var d0139e=1004;    // Fenster Standardbreite bei 1024x768
                    // in Pixel, wenn Taskleiste am unteren Bildrand
                    //   identisch bei Windows 98 und Windows XP
                    //   1004 für IE 6.0 SP2
                    // Die Standardwerte ändern sich je nach Browserversion. Daher wurde der BODY mit
STYLE="overflow:visible"
                    //   instanziert, da damit bei zu grossem Bild des Hintergrundes die Scrollleisten ausgeblendet werden.

// +++++ Konvertiere zu ganzer Zahl
function Y1(X00) // interne Routine
// X00 ist Gleitkomma und mit bliebigem Vorzeichen
// Beispiel zur Konvertierung:          22.5 wird zu 22
//                                     -22.5 wird zu -22
// Achtung: Math.floor() zum obigen Beispiel: 22.5 wird zu 22
//                                     -22.5 wird zu -23 und nicht -22 !
{var X01=0;      // Returnwert
 var X02=0;      // Faktor
 if(X00!=null)
 {if(X00!=0)
 {X01=Math.floor(X00); // als ganze Zahl
                    //   Bsp.: 22.5 wird zu 22 da 22 die nächst kleinere ganze Zahl ist
                    //   -22.5 wird zu -23 da -23 die nächst kleinere ganze Zahl ist
                    //   22 wird zu 22
                    //   -22 wird zu -22

 if(X00!=X01)
 {// Gleitkomma, also konvertieren
 // Regel der Konvertierung: 22.5 wird zu 22
 //   -22.5 wird zu -22 und nicht -23 !
 // Vorzeichen ermitteln
 X02=1; // Faktor, Annahme: X00 ist positiv
 // X00 prüfen auf negatives Vorzeichen
 if(X00<0)
 {X02=-1; // Vorzeichen ist Minus
  X00=-1*X00; // X00 nur als positiven Wert verwenden
 }
 X01=X02*Math.floor(X00);
 }
 }
 }
 return X01;
 }

// +++++ Routine zur Anpassung an die aktuelle Fensterdimension
function Y2(X00,X01) // interne Routine
// X00 anzupassender Wert > 0
// X01 Anpassungsart: true, so Höhenanpassung, sonst Breitenanpassung
// liefert angepassten Wert als ganze Zahl > 0
//   1 wenn Anpassung einen Wert von 0 liefert
//   0 wenn Parameterfehler
{var X02=0; // Returnwert
 var X03=0; // Faktor
 var X04=false;
 var X05=0;
 var X06=0;
 var X07=0;
// ++++++ Parameter prüfen

```



```

X04=(X00!=null);
if(X04){X04=(X01!=null);}
if(X04)
{ // +++++ Fenster-Dimension holen
  X05=document.body.clientHeight;
  X06=document.body.clientWidth;
  // +++++ Returnwert belegen: Annahme: keine Anpassung nötig
  X02=X00;
  // +++++ Prüfung auf Anpassung je nach verfügbarer Fensterdimension
  X04=((X05!=d0139d)||(X06!=d0139e));
}
if(X04)
{ // +++++ Faktor berechnen als Gleitkomma !
  if(X01) // Höhenanpassung
  { X03=X05; // aktuelle Fensterhöhe
    X07=d0139d; // Vergleichswert der Fensterhöhe
  }
  else // Breitenanpassung
  { X03=X06; // aktuelle Fensterbreite
    X07=d0139e; // Vergleichswert der Fensterbreite
  }
  X03/=X07; // Faktor als Gleitkomma
  // +++++ anpassen per Faktor als Gleitkomma
  X02=X03*X00;
  // +++++ und zur ganzen Zahl
  X02=Y1(X02);
  // +++++ wenn 0 so 1 liefern
  if(X02==0){X02=1;}
}
return X02;
}

// ----- Fontsize an Fensterdimension anpassen
function Y3(X00) // interne Routine
// X00 Fontsize >0 in pt
// liefert angepassten Fontsize in pt
// 0 wenn X00 <=0
{ var X01=0;
  var X02=0;
  var X03=0;
  if(X00!=null)
  {if(X00>0)
  { // +++++ an Fensterbreite anpassen
    X01=Y2(X00,false);
    // +++++ an Fensterhöhe anpassen
    X02=Y2(X00,true);
    // +++++ Minimalwert als Returnwert
    X03=Math.min(X01,X02);
  }
  }
  return X03;
}

// ----- formatierten HTML-Text mit freien Angaben ermitteln und nach .innerHTML eines FONT- oder DIV
function Y4(X00,X01,X02,X03,X04,X05,X06,X07) // auch vom Programmierer aufrufbar
// X00 Zeiger auf Objekt mit folgenden Eigenschaften
// .innerHTML z.B. FONT, DIV, SPAN
// .style.fontSize z.B. FONT, DIV, SPAN
// .style.color z.B. FONT, DIV, SPAN
// .style.fontFamily z.B. FONT, DIV, SPAN
// .style.visibility z.B. FONT, DIV, SPAN
// wenn X06 auf true so .style.fontWeight z.B. FONT, DIV, SPAN
// wenn X07 auf true so .align z.B. DIV jedoch nicht FONT und SPAN
// Die Eigenschaften MÜSSEN vorhanden sein !
// X01 Wert von .innerHTML, kann leer sein, wenn leer so auch innerHTML zugewiesen (also Löschung)
// Es darf kein <B> </B> verwendet werden, da dafür X03 zuständig ist
// X02 Font-Size in pt
// wenn <= 0, so auf d0139a gesetzt

```



```

// X03 Font-Size-Abweichung in pt von X02
//      > 0, so Erhöhung des X02 um X03
//      < 0, so Erniedrigung des X02 um X03
//      = 0, so StandardText-FontSize verwendet
//      wenn (Math.abs(X03) >= X02) so wird X03 auf 0 gesetzt
// X04 Font-Farbe (Hexa-String oder vordefinierter Farbbezeichner), wenn Leerkette so keine Farbe erzeugen
// X05 Font-Family der Schriftart als String
// X06 true so Fettdarstellung des Plaintextes (also nicht Text laut .innerHTML) per .style.fontWeight
// X07 true so Zentrierung
// liefert true, wenn Formatierung erfolgreich
{ var X08=false;
  var X09=0;
  var X10="";
    X08=(X00!=null);
  if(X08){X08=(X01!=null);}
  if(X08){X08=(X02!=null);}
  if(X08){X08=(X03!=null);}
  if(X08){X08=(X04!=null);}
  if(X08){X08=(X05!=null);}
  if(X08){X08=(X06!=null);}
  if(X08){X08=(X07!=null);}
  if(X08)
  { // +++++ ERST Objekt ausblenden
    X00.style.visibility='hidden';
    // +++++ Font-Size prüfen
    if(X02<=0){X02=d0139a;}
    // +++++ ERST Font-Size um Abweichung anpassen wenn Abweichung != 0
    if(X03!=0)
    { // ----- prüfen auf negative Fontsizeabweichung
      //      wenn ja, prüfen ob sie im Absolutbetrag nicht größer als der aktuelle Fontsize ist,
      //      also prüfen ob mit Umsetzung der Abweichung ein Fontsize <= 0 entstehen würde
      if(X03<0) // negative Abweichung
      { X09=Math.abs(X03); // Absolutbetrag der Abweichung bilden
        if(X09>=X02) // Absolutbetrag >= Fontsize
        { X03=0; } // es würde negativer Fontsize entstehen, also Abweichung zu 0
      }
      // ----- Font-Size verändern
      X02+=X03; // wenn X03 > 0 so addieren; sonst subtrahieren
      // ----- sicherheitshalber nochmal prüfen
      if(X02<=0){X02=d0139a;}
    }
    // +++++ DANN Fontsize anpassen an aktuelle Fensterhöhe und -breite
    X02=Y3(X02);
    // +++++ ERST Styles setzen
    X00.style.fontSize=X02;
    if(X04!=""){X00.style.color=X04;}
    if(X05!=""){X00.style.fontFamily=X05;}
    X10='normal';if(X06){X10='bold';}X00.style.fontWeight=X10;
    // +++++ DANN .align
    if(X07){X00.align='center';}
    // +++++ DANN innerHTML ändern, auch wenn Leerkette
    X00.innerHTML=X01;
    // +++++ DANN Objekt einblenden
    X00.style.visibility='visible';
  }
  return X08;
}

// ----- formatierten HTML-Text mit Standard-Font-Angaben laut d0139a, d0139b, d0139c ermitteln
//      und nach .innerHTML eines FONT- oder DIV
function Y5(X00,X01,X02,X03,X04) // auch vom Programmierer aufrufbar
// X00 Zeiger auf Objekt mit folgenden Eigenschaften
//      .innerHTML z.B. FONT, DIV, SPAN
//      .style.fontSize z.B. FONT, DIV, SPAN
//      .style.color z.B. FONT, DIV, SPAN
//      .style.fontFamily z.B. FONT, DIV, SPAN
//      .style.visibility z.B. FONT, DIV, SPAN
//      wenn X03 auf true so style.fontWeight z.B. FONT, DIV, SPAN

```



```

//      wenn X04 auf true so .align                z.B. DIV jedoch nicht FONT und SPAN
//      Die Eigenschaften MÜSSEN vorhanden sein !
// X01 Wert von .innerHTML, kann leer sein, wenn leer so auch innerHTML zugewiesen (also Löschung)
//      Es darf kein <B> </B> verwendet werden, da dafür X03 zuständig ist
// X02 Font-Size-Abweichung in pt von d0139a
//      > 0, so Erhöhung des d0139a um X02
//      < 0, so Erniedrigung des d0139a um X02
//      = 0, so StandardText-FontSize verwendet
//      wenn (Math.abs(X02) >= X02) so wird X02 auf 0 gesetzt
// X03 true so Fettdarstellung des Plaintextes (also nicht Text laut .innerHTML) per .style.fontWeight
// X04 true so Zentrierung
// verwendet
//      d0139a   Standard-FontSize in pt
//      d0139b   Standard-Fontfarbe
//      d0139c   Standard-Font-Family (Schriftart)
// liefert true, wenn Formatierung erfolgreich
{var X05=false;
X05=Y4(X00,X01,d0139a,X02,d0139b,d0139c,X03,X04);
// X00 Zeiger auf Objekt mit folgenden Eigenschaften
//      .innerHTML                z.B. FONT, DIV, SPAN
//      .style.fontSize           z.B. FONT, DIV, SPAN
//      .style.color              z.B. FONT, DIV, SPAN
//      .style.fontFamily         z.B. FONT, DIV, SPAN
//      .style.visibility         z.B. FONT, DIV, SPAN
//      wenn X06 auf true so style.fontWeight z.B. FONT, DIV, SPAN
//      wenn X07 auf true so .align                z.B. DIV jedoch nicht FONT und SPAN
//      Die Eigenschaften MÜSSEN vorhanden sein !
// X01 Wert von .innerHTML, kann leer sein, wenn leer so auch innerHTML zugewiesen (also Löschung)
//      Es darf kein <B> </B> verwendet werden, da dafür X03 zuständig ist
// X02 Font-Size in pt
//      wenn <= 0, so auf d0139a gesetzt
// X03 Font-Size-Abweichung in pt von X02
//      > 0, so Erhöhung des X02 um X03
//      < 0, so Erniedrigung des X02 um X03
//      = 0, so StandardText-FontSize verwendet
//      wenn (Math.abs(X03) >= X02) so wird X03 auf 0 gesetzt
// X04 Font-Farbe (Hexa-String oder vordefinierter Farbbezeichner), wenn Leerkette so keine Farbe erzeugen
// X05 Font-Family der Schriftart als String
// X06 true so Fettdarstellung des Plaintextes (also nicht Text laut .innerHTML) per .style.fontWeight
// X07 true so Zentrierung
// liefert true, wenn Formatierung erfolgreich
return X05;
}
</SCRIPT>
</HEAD>
<BODY>
<DIV ID='XDIV'>Das ist der Testtext</DIV>
<SCRIPT TYPE="text/javascript" LANGUAGE="JScript">
alert();
alert(Y5(XDIV,'Neuer Text',50,false,false));
// X00 Zeiger auf Objekt mit folgenden Eigenschaften
//      .innerHTML                z.B. FONT, DIV, SPAN
//      .style.fontSize           z.B. FONT, DIV, SPAN
//      .style.color              z.B. FONT, DIV, SPAN
//      .style.fontFamily         z.B. FONT, DIV, SPAN
//      wenn X03 auf true so style.fontWeight z.B. FONT, DIV, SPAN
//      wenn X04 auf true so .align                z.B. DIV jedoch nicht FONT und SPAN
//      Die Eigenschaften MÜSSEN vorhanden sein !
// X01 Wert von .innerHTML, kann leer sein, wenn leer so auch innerHTML zugewiesen (also Löschung)
//      Es darf kein <B> </B> verwendet werden, da dafür X03 zuständig ist
// X02 Font-Size-Abweichung in pt von d0139a
//      > 0, so Erhöhung des d0139a um X02
//      < 0, so Erniedrigung des d0139a um X02
//      = 0, so StandardText-FontSize verwendet
//      wenn (Math.abs(X02) >= X02) so wird X02 auf 0 gesetzt
// X03 true so Fettdarstellung des Plaintextes (also nicht Text laut .innerHTML) per .style.fontWeight
// X04 true so Zentrierung
// verwendet
//      d0139a   Standard-FontSize in pt

```



```
//      d0139b   Standard-Fontfarbe
//      d0139c   Standard-Font-Family (Schriftart)
// liefert true, wenn Formatierung erfolgreich
</SCRIPT>
</BODY>
</HTML>
```



26. Index

		91	Anzeige Datum	154

	91	Anzeige Informationen zu Javaklassen	206
:	67	Anzeige Informationen zum Betriebssystem	206
@	67	Anzeige Uhren	156
&#xx;	288	Anzeige Uhrzeit und Datum	154
&#xxx	67	Apache-HTTP-Server	13
*.js	138	Applet-Darstellungsmöglichkeit	51
.Apply();	75	Apply();	75
.createPopup()	90	Argumente	46
.decodeURI()	68	Array anlegen und initialisieren	34
.encodeURI()	68	Array eindimensional	34
.filters[0].Apply();	75	Array Größe	35
.filters[0].Play();	76	Array Größe nachträglich ändern	35
.htaccess	19	Array Hinzufügung von neuen Elementen	35
.ico	92	Array Index	34
.innerHTML	90	Array Konstruktor	34
.overflowX	109	Array mehrdimensional	34
.overflowY	109	Array Sortieren	40
.Play();	76	Arrayelemente nach deren Initialisierung erweitern	39
.show()	90	Array-Elemente Reihenfolge ändern	39
.style.visibility	75	Array-Variable	39
.style.zoom	79	Array-Variablen	34
.visibility	75	Auf ActiveX prüfen beim IE	91
.zoom	79	auf IE 5.5 prüfen	27
@ 67		auf IE-Version 5.5 prüfen	27
<DIV>	188	auf Integerwert aus einem Gültigkeitsbereich prüfen	127
<Hx>	188	auf vorgegebene Zeichenanzahl prüfen	127
<INPUT	169	auf Ziffern prüfen	127
<LAYER>	101, 199	Auflösung Dokument	52
<LINK>	194	Auflösung Fenster	52
<NOSCRIPT>	20	Auflösung Fenster ändern	112
<SCRIPT LANGUAGE="JavaScript">	20	Auflösung optimal	52
<SCRIPT>	20	Aus- und Einblende von Bildern mit Bildwechsel	76
	187	ausblenden	98
<STYLE>	73	auslesen	124
1 Seite vorwärts gehen	94	Auslesen einer Auswahlliste	125
1 Seite zurück gehen	94	Auslesen einer Checkbox	124
24-Stunden Digitaluhr	92	Auslesen eines Eingabefeldes	124
3D-Effekt eines Bildes	79	Ausschnitt erzeugen	190
68K	51	Aussehen eine Button mit Mausklick verändern	131, 132
Abmahnungsgefahr	53	Aus- und Einblende eines Bildes	75
Abschaltung der automatischen Umrandung von angeklickten Objekten IE	71	Auswahlliste auslesen	125
absolute Pfade	13	Auswahlliste mit einer Auswahlmöglichkeit	125
Acrobat-Reader-Plugin	61	Auswahlliste mit mehreren Auswahlmöglichkeiten	125
ActiveX prüfen	91	Auswahlmöglichkeit	125
AddFavorite()	71	Auswahlmöglichkeiten	125
Adobe Acrobat-Reader-Plugin	61	automatisch scrollen	97
aktuellen Tag prüfen	153	Bad Request	19
aktuelles Fenster	100	Beispiel für 3D-Effekt eines Bildes	79
alert()	101	Beispiel für Aus- und Einblende von Bildern mit Bildwechsel	76
Alert-Fenster	101	Beispiel für Schatten eines Bildes	78
Alle Anker eines Dokumentes anzeigen	94	Beispiel für transparente Überlagerung zweier Bilder	75
Alle Textmarken (Anker) eines Dokumentes anzeigen	94	Beispiel für transparentes INPUT-Element	74
Alpha	51	Beispiel für transparentes Vordergrund-Bild auf dem Hintergrundbild	75
Alpha-Filter für HTML-Grafik-Elemente ab IE 5.5	74	Beispiel: Uhr in der Statuszeile	158
alphanumerisches Zeichen	68	Beispiele für Aus- und Einblende eines Bildes	75
als oberstes Fenster anzeigen	96	berechnen	35
ALT-Kode	67	Besondere Zeichen suchen	151
an Url angehängte Textdaten	68	Betreff einer Email	62
anchors[]	95	Betriebssystem unter dem der Browser läuft	51
anchors[].name	95	Bild in einem neuen Fenster zentriert öffnen	165
Anker eines Dokumentes anzeigen	94	Bild positionieren	188
Anker suchen	153	Bild preloaden	158
Anzahl immer ab 1	35	Bild verschwinden lassen	167
Anzeige Bilderwechsel	159	Bild vorladen	161
Anzeige Bildfolge	160	Bildauflösung	79



Bilder mit Bildwechsel	76	CSS Text positionieren	188
Bilder und Animationen.....	158	CSS Überschrift mit permanentem Farbwechsel.....	188
Bilderwechsel.....	159	CSS-Layer	101
Bildfolge	160	cur-Datei.....	177
Bildschirmauflösung des Users ermitteln.....	52	Dateipfad	13
Bildwechsel.....	76	Dateipfad absolut	13
Blanks nicht in Zeichenkettenparameter	101	Dateipfad relativ.....	13
Blanks und Parameter	101	Dateiverzeichnis lesen	98
blind copy.....	62	Daten aus TEXTAREA.....	68
blinken.....	119	Daten im RAM.....	71
blur().....	91	Datenaustausch zwischen Frames.....	57
Body Scrolleisen abschalten.....	109	Datensätze als Teil einer Javascript Datei	138
body.clientHeight.....	96	Datenstrukturen	136
body.clientWidth.....	95	Datenübergabe	68
body.focus()	71	Datum.....	153
Bookmark per Script erzeugen IE.....	71	Datum der letzten Änderung.....	53
break-Anweisung für Schleifen	44	Datumsangaben.....	155
Browser	20	Datumsanzeige in deutscher Norm	155
Browser Betriebssystem	51	Deckkraft.....	74
Browser Objektfähigkeit.....	27	Deckungsstärke	74
Browser Sicherheitseinstellungen.....	49	Decoding und Encoding von URI's	68
Browser- und System-Informationen	48	Dezimal zu Hexa	99
Browsererkennung.....	21	Dezimalkomma	127, 128
Browsererkennung durch Ermittlung der Leistungsunterschiede21		Dezimalkomma zu Dezimalpunkt.....	127
Browsererkennung zur Unterscheidung browserinterner Objekte27		Dezimalpunkt	127, 128
Browserfenster	111	Dezimalpunkt zu Dezimalkomma.....	127
Browserfenster rausschieben und danach neu anzeigen	101	Dezimalwert zu Hexaziffern-Zeichenkette konvertieren	38
Browser-Informationen.....	48	DHTML.....	186
browserinterne Objekte.....	27	Digital.....	51
Browsersname	48	Digitaluhr	92
Browserplattform	51	direction.....	78
Browserprüfung	27	Directory lesen	98
Browserüberprüfung auf Netscape	49	display:none;	97
Browserüberprüfung auf Microsoft Explorer ab 4.x.....	49	DIV	101
Browserversion: Hauptversion vor dem Punkt.....	49	Division ganzzahlig	136
Browserversion: Updatenummer nach dem Punkt	49	document.body.style.overflowX.....	109
Browserversionen	27	document.body.style.overflowY.....	109
Bubble-up Event	150	document.referrer	52
Buttons	131	document.write()	100
Buttons Aussehen eines Button mit Mausklick verändern131, 132		Dokument automatisch scrollen.....	97
cancelBubble Event	150	Dokument Datum der letzten Änderung.....	53
captureEvents()	148	Dokument drucken	96
carbon copy	62	Dokument Eigenes Dokument wird durch fremde Webseite geladen	53
Cascading Style Sheets (CSS)	186	Dokument Frame-Dokument mit zugehörigem Frameset	56
CGI-Script.....	171	Dokument fremdes Dokument laden ohne Framedarstellung	53
Checkbox auslesen.....	124	Dokument Laden eines fremden Dokumentes ohne Framedarstellung	53
CLASS	73	Dokument mit klingendem Sound laden	167
CLASSID	91	Dokument nach dem Laden als oberstes Fenster anzeigen	96
clientHeight.....	96	Dokument Reload mit allen seinen FRAMES.....	120
clientWidth.....	95	Dokumenten-Hintergrund.....	98
Clipping.....	190	Dokumenten-Hintergrund auswählen	100
close().....	111	Dokumentinhalt als zusätzlichen Frame anzeigen.....	95
Codename Engine	49	Doppelpunkt.....	67
Communicator.....	147, 194	do-while-Schleife	44
CONTENT=.....	59	drucken	96
continue-Anweisung für Schleifen	45	durch eine Funktion definieren	33
Cookies	171	DXImageTransform.Microsoft.Fade().....	74
CopyRight-Meldung.....	53	Dynamisches Erzeugen von Layer-Objekten	198
Cowntdown-Uhr	156	Eieruhr.....	156
CPU-Typ.....	51	Eigenes Dokument wird durch fremde Webseite geladen.....	53
CSS.....	186	Eigenschaften eines Browserfensters	111
CSS Ausschnitt erzeugen.....	190	ein- bzw. ausblenden.....	98
CSS Bild positionieren	188	Ein FRAME für mehrere HTML-Dokumente	123
CSS Clipping	190	Einblende eines Bildes	75
CSS Definition direkt in einem Tag	194	Einblende von Bildern mit Bildwechsel.....	76
CSS Grafik-Thumbnails	190	eindimensionales Array.....	34
CSS Klasse Tag-abhängige Kodierung ohne ID.....	194	Eingabe überprüfen	152
CSS Klasse Tag-unabhängige Kodierung mit ID.....	187		
CSS Klassen erzeugen	186		



Eingabefeld auslesen	124
Eingabefeld Wert zuweisen	126
Einrichten eines virtuellen Hosts per Apache-HTTP-Server	13
Einrichtung einer Webseite per Apache-HTTP-Server	14
E-Mail	62
Email senden per Formular	62
E-Mail Spam-Schutz	67
Email und Stringkonvertierung ab Internet Explorer 5.5	68
Email-Absendung mit Bestätigung für User	68
E-Mail-Adresse überprüfen	150
Encoding und Decoding von URI's	68
Ereignisse	140
Ermittlung einer Taste	173
Error 404	19
ErrorDocument	19
Errormeldungen des HTTP-Servers	19
escape()	68
Event-Handler onAbort	140
Event-Handler onBlur	140
Event-Handler onChange	141
Event-Handler onClick	141
Event-Handler onError	143
Event-Handler onFocus	144
Event-Handler onKeyPress	145
Event-Handler onLoad	146
Event-Handler onMouseout	146
Event-Handler onMouseover	146
Event-Handler onReset	146
Event-Handler onSelect	147
Event-Handler zusätzliche	149
event-Objekt	148, 149
Events	140
Events im Microsoft Internet Explorer	149
Events im Netscape Communicator	147
external.AddFavorite()	71
Fade	74
Fakultät	44
Favorit per Script erzeugen IE	71
Fehler umleiten	143
Fehlerbehandlung durch browsereigene Routinen	143
Fehlermeldung in einem neuen Fenster	143
Fehlermeldung per alert()	143
Fehlermeldungen	143
Fenster	96, 100
Fenster aktuelles	100
Fenster des Browser rausschieben	101
Fenster öffnen	110
Fenster öffnen ohne die Eigenschaften eines Browserfensters	111
Fenster schliessen	111
Fenster schließen, das nicht FRAME ist	111
Fenster schliesst sich selbst nach Wartezeit	112
Fenster Standardgröße	110
Fenster Statuszeile	115
Fenster Statuszeile Hinweise Formularfelder anzeigen	118
Fenster Statuszeile Hyperlink-Text anzeigen	117
Fenster Statuszeile Scroller	116
Fenster Statuszeile Standardtext	115
Fenster Statuszeile Text in fester Zeitspanne blinkend anzeigen	119
Fenster Statuszeile Text Buchstabe für Buchstabe anzeigen	115
Fenster vordefinierte Größe	110
Fenster, das andere Fenster aufruft, als oberstes anzeigen	111
Fenster, das Frame ist: Oberstes Fenster schliessen	112
Fenster aufblasen bei Auflösung von 640x480	112
Fenster auflösung ändern	112
Fenster-Handle	69
Fenstertitel nachträglich ändern	113
Fenstertitel permanet wechseln	113
Fenstertitelzeile mit scrollendem Text	113
feste, nicht mitscrollende Objekte	161

Festes, nicht mitscrollendes Bild	161
Festplattenordner der Webseite	13
Festplattenverzeichnis unter Windows auslesen	98
Feststellung des Dialektes zum aktiven Browser	21
file:///C:/	98
Filter für HTML-Elemente ab IE 5.5	72
Filter kodieren	73
filter: alpha(opacity=deck_kraft)	72
filter: blur(add= ..strength=..., direction=...)	72
filter: chroma(color=...)	72
filter: dropShadow(color=schatten_farbe, offX=..., offY=...)	72
filter: flipH	72
filter: flipV	72
filter: glow(color=..., strength=...)	72
filter: gray	72
filter: invert	72
filter: mask(color=...)	72
filter: progid:DXImageTransform.Microsoft.Emboss	73
filter: progid:DXImageTransform.Microsoft.MotionBlur(strength=..., direction=...)	73
filter: progid:DXImageTransform.Microsoft.Wheel(duration=...)	73
filter: shadow(color=schatten_farbe, direction=...)	73
filter: wave(strength=..., freq=...)	73
filter: xray	73
filter:Alpha()	74
filter:progid:DXImageTransform.Microsoft.Fade()	74
filter:revealTrans(duration=..., transition=...)	72
filter:shadow()	78
Filterarten	72
filters[0].Apply()	75
filters[0].Play()	76
focus()	111
for -Schleife	42
Formular Laufschrift	129
Formular Uhr anzeigen	157
Formulare	124
Formulare Auslesen einer Auswahlliste	125
Formulare Auslesen einer Checkbox	124
Formulare Auslesen eines Eingabefeldes	124
Formulare Wert einem Eingabefeld zuweisen	126
Formulare Werte auslesen	124
Formulare Werte konvertieren	127
Formulare Werte überprüfen	126
Formulare Werte zuweisen	126
Formularfelder-Hinweise in Statuszeile anzeigen	118
Formular-Option-Menü aus Hyperlinks	131
Formularprüfung für Submit	128
Formular-Text-Eingabe überprüfen	152
Formular-Text-Pulldown-Menü	130
Formular-Text-Pulldown-Menü mit Button	130
Frame	
Oberstes Fenster schliessen	112
Frame anzeigen	95
FRAME Ein FRAME für mehrere HTML-Dokumente	123
FRAME HTML-Inhalte von FRAMES tauschen	121
FRAME Reload eines Dokumentes mit allen seinen FRAMES	120
Frame und Datenaustausch	57
FRAME Werbefenster als FRAME alle 3 Sekunden einblenden	120
Framedarstellung und Abmahnungsgefahr	53
Framedarstellung und rechtliche Zulässigkeit	53
Frame-Dokument mit zugehörigem Frameset laden	56
Frameinhalte	56
FRAMES	119
FRAMES als Menüleiste	129
FRAMES füllen	119
frames[]	120
Frameset	56
freie Suche nach Stichworten	139



fremde Homepage laden	110
fremde Webseite	53
Fremdes Dokument laden ohne Framedarstellung	53
Funktion	34
Funktion rekursiv	44
Funktion Übergabewerte	46
Funktionen	45
Funktionen mit Konstruktor	46
Funktionen ohne Konstruktor	45
Funktionen Rückgabewert undefined	48
Funktionen Rückgabewerte	47
ganzahlige Division	136
ganzzahlig konvertieren	135
ganzzahlig teilbar konvertieren	135
gemeinsamer Server	123
geschützte Homepage	171
GET	98
getProperty()	206
getTime()	94
global definieren	33
Globale Variable	33
Grafik vorladen	161
Grafik zoomen ab IE 5.5	79
Grafiken	74
Grafik-Thumbnails	190
Größe eines Arrays	35
Größter gemeinsamer Teiler zweier ganzer Zahlen	135
Gültigkeitsbereich	127
Handle	69
hash	53
Hauptversion vor dem Punkt	49
height=	111
Hexa	99
HexaKette	99
Hexaziffern	38
Hintergrund auswählen	100
Hintergrund ein- bzw. ausblenden	98
Hintergrundbild	75
Hintergrundbild beim IE als Wasserzeichen	92
Hinweise zu Formularfelder in Statuszeile anzeigen	118
Hinzufügung von neuen Elementen	35
History anwenden	94
history.go()	94
history.go(+1)	94
history.go(0)	72
history.go(-1)	94
History-Eintrag ersetzen	57, 94
Homepage laden	110
host	53
Host der Webseite	13
hostname	13, 53
href	53
HREF	117, 170
HTML-Code ausführen	57
HTML-Code in einem Fenster erzeugen und zugleich ausführen	57
HTML-Datenstrukturen	136
HTML-Dokument	52
HTML-Dokument Informationen ermitteln	52
HTML-Dokumente	123
HTML-Dokumente Textdaten-Übergabe	68
HTML-Elemente	72
HTML-Inhalte von FRAMES tauschen	121
HTTP.EQUIV="Refresh"	59
HTTP-Server	13
Hyperlink	117, 170
ICO-Datei	92
ID 187, 194	
IE 101	
IE 5.5	27, 72

IE 5.5 Stackoperationen ab für das Array-Objekt	91
IE 5.x Webseite mit eigenem Icon	92
IE Abschaltung der automatischen Umrandung von angeklickten Objekten	71
IE Alpha-Filter für HTML-Grafik-Eelmente ab IE 5.5	74
IE auf ActiveX prüfen	91
IE Bookmark per Script erzeugen	71
IE erkannt	48
IE Favorit per Script erzeugen	71
IE Filter für HTML-Eelmente ab IE 5.5	72
IE Filter kodieren	73
IE Filterarten	72
IE Grafik zoomen ab IE 5.5	79
IE Hintergrundbild als Wasserzeichen	92
IE Kodierung von Filtern	73
IE Layertechnik	76
IE Linkrahmen unterdrücken	91
IE PopUp-Fenster ab IE 5.5	90
IE Reload des Dokumentes nach Resize des Browserfenster	72
IE Shadow-Filter für HTML-Grafik-Elemente ab IE 5.5	78
IE Uhr	92
IE und Tooltip	91
if (document.all)	21
if (document.layers)	21
if-else-Verzweigung	41
Index	34
Index immer ab 0	34, 35
indexOf()	37
Informationen ermitteln	52
Informationen zu Javaklassen	206
Informationen zum Betriebssystem	206
Inhalt des aktuellen Dokumentes als zusätzlichen Frame anzeigen	95
innerHeight	96
innerWidth	96
INPUT-Element	74
Instant Page Update	202
Integerwert	127
Intel	51
Internal Server Error	19
Internet Explorer	71, 149, 201
Internet Explorer 5.5	68
Internet Explorer erkannt	48
Internet-Provider	13
isNaN()	36
ISO Latin-1	68
Java Konsole	205
java.class.version	206
java.lang.System	206
java.util.Stack()	71
java.vendor	206
java.vendor.url	206
java.version	206
Java-Applet-Darstellungsmöglichkeit	51
javaEnabled()	51
Java-JavaScript Kommunikation	205
Java-Klassen	206
JavaScript	27
Javascript 1.1	20
Javascript 1.2	135
Javascript Datei	138
JavaScript-Bibliothek	20
Javascript-Datei	20
Javascript-Version	19
Java-Stack	71
join()	39
Kapitalchen	207
Kodierung von Filtern	73
Konstruktor	34, 46



konvertieren	127
konvertieren Dezimalwert zu Hexaziffern	38
konvertieren Zeichenkette zu numerisch	37
konvertieren zu ganzzahlig	135
konvertieren zu ganzzahlig teilbar	135
Konvertierung von String zu URI	68
Konvertierung von URI zu String	68
Laden eines fremden Dokumentes ohne Framedarstellung	53
Länge immer ab 1	34, 35
Last in First Out	71
Laufschrift in einem Formular	129
Layer und Style Unterscheidung je nach Browser	203
Layer()-Konstruktor	199
Layer-Eigenschaft document	196
Layer-Objekt	101
Layer-Objekt bewegen	196
Layer-Objekt clipping	198
Layer-Objekt dynamisch erzeugen	198
Layer-Objekt erzeugen	194
Layer-Objekt im Netscape Communicator	194
Layer-Objekt Laden einer neuen HTML-Seite	197
Layer-Objekt und JavaScript	197
Layer-Objekte verschachtelt	199
Layers	21
Layertechnik des IE	76
Leerkette	126
Leerzeichen	127
LiFo	71
Link	207
Linkrahmen unterdrücken beim IE	91
Linux	51
Liste mit einer Auswahlmöglichkeit	125
Liste mit mehreren Auswahlmöglichkeiten	125
location.hash	53
location.host	53
location.hostname	13, 53
location.href	53, 59
location.pathname	53
location.port	53
location.protocol	53
location.search	53
location-Objekt	52
lokal definiert	33
lokal zu einer Funktion	34
Lokale Variable	34
mac	48
Mailtext einer Email	62
mailto	62
major	48
Math.floor()	38, 94
Math.PI	134
Math.pow()	52
Mathematische Berechnungen	134
math-Objekt	134
Matrix	35
Matrix zweidimensional	35
Maus	173
Maustaste belegen	174
MEDIA="print"	97
mehrdimensionales Array	34
mehrere Auswahlmöglichkeiten	125
Mehrere Frameinhalte gleichzeitig ändern	56
mehrere HTML-Dokumente	123
mehrere Seiten vorwärts oder rückwärts gegen	94
mehrere Seiten wahlweise vorwärts oder rückwärts gegen	94
Menü	129
Menü aus Hyperlinks	131
menubar=	111
Menüleiste	129

Menüleiste in einem separaten Fenster	129
Menüleiste mittels FRAMES	129
Microsoft	21, 27, 74, 101
Microsoft Explorer ab 4.x	49
Microsoft Instant Page Update	202
Microsoft Internet Explorer	61, 71, 149, 201
Microsoft Internet Explorer erkannt	48
Microsoft spezifisch	52
Microsoft Style und Netscape Layer Unterscheidung	203
Microsoft style-Objekt bewegen	201
Microsoft style-Objekt clipping	202
Microsoft style-Objekt erzeugen	201
MIME-Type	20
minor	48
Minuszeichen	128
Minuten-Eieruhr	156
Motorola	51
Moved Permanently	19
Moved Temporarily	19
msie	48
nach dem Alphabet sortieren	40
nach Zahlenwerten sortieren	40
Nachkommastellen	134
NAME	169
Name des Browsers	48
navigator.appCodeName	49
navigator.appMinorVersion	49
navigator.appName	48
navigator.appName.subString()	111
navigator.appUserAgent	49
navigator.appVersion	48, 49
navigator.appVersion.indexOf()	49
navigator.cpuClass	51
navigator.javaEnabled()	51
navigator.onLine	51
navigator.platform	51
navigator.plugins.length	61
navigator.plugins[]	61
navigator.plugins[].description	62
navigator.plugins[].filename	62
navigator.plugins[].name	61
navigator.securityPolicy	50
navigator.userAgent	48, 51
navigator.userAgent.indexOf	51
navigator-Objekt	48
Netscape	21, 27, 49, 61, 101
Netscape Communicator	147, 194
Netscape erkannt	48
Netscape Layer und Microsoft Style Unterscheidung	203
Netscape Layer()-Konstruktor	199
Netscape Layer-Eigenschaft document	196
Netscape Layer-Objekt bewegen	196
Netscape Layer-Objekt clipping	198
Netscape Layer-Objekt dynamisch erzeugen	198
Netscape Layer-Objekt erzeugen	194
Netscape Layer-Objekt Laden einer neuen HTML-Seite	197
Netscape Layer-Objekt und JavaScript	197
Netscape Layer-Objekte verschachtelt	199
new Array()	34
new()	45
nicht mitscrollende Objekte	161
Nicht-Popup-Fenster	110
No Content	19
Not Found	19
Not Implemented	19
numerische Operationen	35
numerische Variable	35
Numerischen Wert als Zeichenkette liefern	127
Nummernkreuz	68



nur beim Druck anzeigen	97
oberstes Fenster	96, 111
Oberstes Fenster schliessen	112
Objekt anlegen	46
Objekt nachträglich erweitern	46
Objekte browserinterne	27
Objektfähigkeit des Browsers	27
objekt-kompatibel	27
oCliCap.connectionType	52
Offline-Zustand	51
öffnen Fenster	110
onAbort	140
onBlur	140
onChange	141
onClick	141, 147
onError	143
onFocus	144
onKeyPress	145
Online- oder Offline-Zustand	51
Online-Zustand	51
onLoad	146, 147
onMouseout	146
onMouseover	146
onReset	146
onResize	72, 98
onSelect	147
opacity	74
open()	110
opera	48
Operationen mit Variablen	35
Operationen mit Variablenwerten	35
Operationen numerisch	35
optimale Auflösung	52
Option-Menü aus Hyperlinks	131
os.arch	206
os.name	206
os.version	206
overflowX	109
overflowY	109
Parameter als Zeichenkette	101
parseFloat()	37, 128
parseInt()	36, 101
password	169
Password	169
Password Eingabe per Formular	169
Password Eingabe per Hyperlink (HREF) und prompt	170
Password in einer Variablen als Auswahlkriterium für eine HTML-Seite	169
Password zufällig erzeugen	170
pathname	53
permanenter Fenstertitel-Wechsel	113
Pfade für Dateien	13
PI 134	
Pixelgröße 1x1	158
Play();	76
Plugin-Prüfung	61
Plugins auflisten	61
plugins.length	61
plugins[].description	62
plugins[].filename	62
plugins[].name	61
pop()	71, 91
Popup-Fenster	110
PopUp-Fenster ab IE 5.5	90
Popup-Fenster nach 10 Sekunden erzeugen	153
port	53
Positionierung eines Bildes	188
Positionierung eines Textes	188
PPC	51

preloaden	158
print()	96
protocol	53
Prototyping	46, 137, 161, 207
Prozessortyp	51
prüfen aktuellen Tag	153
prüfen auf Integerwert aus einem Gültigkeitsbereich	127
prüfen auf Leerkette	126
prüfen auf Leerzeichen	127
prüfen auf vorgegebene Zeichenanzahl	127
prüfen auf Ziffern	127
prüfen nicht numerisch	36
Pulldown-Menü	130
Pulldown-Menü mit Button	130
push()	71, 91
RAM	71
Realisierung einer Menüleiste in einem separaten Fenster	129
Realisierung einer Menüleiste mittels FRAMES	129
Rechte Maustaste belegen	174
reg-Methode	36, 37, 150
reguläre Ausdrücke	36
Reguläre Ausdrücke	150
Reihenfolge von Array-Elementen ändern	39
Rekursion	44, 135
rekursive Funktion	44
REL=	97
relative Pfade	13
releaseEvents()	148
Reload des Dokumentes nach Resize des Browserfenster IE	72
Reload des Dokumentes nach Resize des Browserfensters	98
Reload eines Dokumentes mit allen seinen FRAMES	120
Reload mit allen FRAMES	120
reload()	120
Request Time Out	19
resizeable=	111
return	47
reverse()	39
Root der Webseite	13
Rückgabe durch return	47
Rückgabewert einer Funktionen	47
Rückgabewert undefined	48
Runden auf n Nachkommastellen	134
Schatten eines Bildes	78
Schattenbreite	79
Schattenrichtung	78
schliessen Fenster	111
schliessen nach Wartezeit	112
schwebender Text von rechts nach links bei 640 Pixel Bildschirmbreite	101
screen	48
screen.colorDepth	52
screen.height	52
screen.pixelDepth	52
screen.width	52
screenX=	111
screenY=	111
scrollbars=	111
Scrolleleisten des Body abschalten	109
scrollen	97
scrollender Text	113
Scroller in der Statuszeile	116
scrollTo()	98
search	53
Seiten wahlweise vorwärts oder rückwärts gehen	94
Seitenelemente nur beim Druck anzeigen	97
Seitenelemente vom Druck ausschließen	97
Server	123
Service Unavailable	19
shadow()	78



Shadow-Filter für HTML-Grafik-Elemente ab IE 5.5	78	Textdaten-Übergabe zwischen zwei HTML-Dokumenten.....	68
sich auflasendes Fenster bei Auflösung von 640x480	112	Text-Eingabe überprüfen	152
sich selbst schliessen nach Wartezeit	112	Texteinzug	91
Sicherheitseinstellungen	49	Textmeldung in Statuszeile	207
SIZE	169	Text-Pulldown-Menü	130
Slide-Show-Effekt	76	Text-Pulldown-Menü mit Button.....	130
sort()	40	Textticker	208
Sortieren	40	Thumbnails	190
sortieren alphabetisch	40	Timer, Datum, Uhrzeit	153
sortieren mit Sortierfunktion	40	TITLE	91
sortieren nach Zahlenwerten	40	toolbar=	111
sortieren ohne Sortierfunktion	40	Tooltip beim IE	91
Sound	167	transparent	74
Sound für IE und NS	168	transparente Überlagerung zweier Bilder	75
Spam-Schutz	67	transparentes INPUT-Element	74
Stack	71	transparentes Vordergrund-Bild auf dem Hintergrundbild	75
Stackoperationen ab IE 5.5 für das Array-Objekt	91	TYPE	169
Standardgröße	110	Übergabewerte an Funktion	46
Standardtext in der Statuszeile	115	Überlagerung zweier Bilder	75
status	115	Überprüfen einer E-Mail-Adresse	150
Statuszeile	115, 158, 207	Überprüfung einer Formular-Text-Eingabe	152
Statuszeile Hinweise Formularfelder anzeigen	118	Überschrift mit permanentem Farbwechsel	188
Statuszeile Hyperlink-Text anzeigen	117	Uhr im Formular anzeigen	157
Statuszeile Scroller	116	Uhr in der Statuszeile	158
Statuszeile Standardtext	115	Uhr mit IE-Filter	92
Statuszeile Text in fester Zeitspanne blinkend anzeigen	119	Uhren anzeigen	156
Statuszeile wechselnder Text	115	Uhrzeit	153
Statuszeile Text Buchstabe für Buchstabe anzeigen	115	um 1 Seite vorwärts gehen	94
Stichworte freie Suche	139	um 1 Seite zurück gehen	94
Stichworte Suche	139	um mehrere Seiten wahlweise vorwärts oder rückwärts gegen	94
Stichwortliste	139	Umleiten auf andere Url	59
strength	79	Umrechnung screen.colorDepth in Farben	52
style	101	Umrechnung screen.pixelDepth in Farben	52
Style Sheet Definition direkt in einem Tag	194	Umrechnung Zoll nach Zentimeter	35
Style Sheets	186	Unauthorized	19
Style und Layer Unterscheidung je nach Browser	203	undefined	48
style.visibility	75	unescape()	69
STYLE=	101	Universal Resource Index	68
style-Objekt bewegen	201	Unterscheidung der Browser	27
style-Objekt clipping	202	Unterscheidung Layer und Style je nach Browser	203
style-Objekt erzeugen	201	Unterscheidung von Browsern	20
style-Objekt im Microsoft Internet Explorer	201	Update-Datum anzeigen	60
Submit	128	Updatenummer nach dem Punkt	49
Submit mit Bestätigung für User	68	URI	68
substring()	128	Url	59
Suchalgorithmen	150	Url ermitteln von der der User herkommt	52
Suche nach Stichworten	139	Url mit angehängten Textdaten	68
Suchmaschine	56	URL=	59
Suchmaschine einbinden	61	User Bildschirmauflösung	52
switch-Verzweigung	42	User Url ermitteln	52
System-Informationen	48, 51	Useragent	49
Tagebrechnungen	153	Userbestätigung für E-Mail-Absendung	68
Tastatur und Maus	173	User-Hinweis auf optimale Auflösung	52
Taste ermitteln	173	User-Informationen	52
teilbar konvertieren	135	User-Verweilzeit in Sekunden ermitteln	94
Teilketten suchen	152	VALUE	169
Teilstring suchen	37	var ie = document.all ? true : false;	27
temporäres Zwischenspeichern von Daten	71	var ns = document.layers ? true : false;	27
Text blinkend anzeigen	119	Variable Array	34, 39
Text Buchstabe für Buchstabe in Statuszeile anzeigen	115	Variable definieren	33
Text in Statuszeile in fester Zeitspanne blinkend anzeigen	119	Variable definieren durch eine Funktion	33
Text- Kapitälchen	207	Variable global gültig	33
Text positionieren	188	Variable lokal	34
Text scrollend	113	Variable numerisch	35
Text zu einem Hyperlink (HREF) für eine feste Zeispanne in Statuszeile anzeigen	117	Variable Zeichenkette	36
Textausgaben im aktuellen Fenster	100	Variablen	33
Textdaten-Übergabe durch an Url angehängte Textdaten	68	Variablen und ihre Gültigkeit	33
Textdaten-Übergabe durch Fenster-Handle	69	Verkettung von Arrayelementen des Zeichenkettentyps	39
		Verschachtelte Layer-Objekte	199



Verweilzeit in Sekunden des Users ermitteln	94	Windows 95	51
Verweis	207	Windows 98	51
Verzeichnis lesen	98	with=	111
Verzweigungen und Schleifen	41	write()	100
virtueller Host	13	WWW-Seite nach 10 Sekunden laden	153
visibility	75	x11	48
void()	48	x86	51
vom Druck ausschließen	97	Zeichen suchen	151
Vordergrund-Bild auf dem Hintergrundbild	75	Zeichenanzahl	127
vorladen Bild	161	Zeicheneingabe prüfen	36
vorladen Grafik	161	Zeichenkette	127
Vornull bei Ziffern-Zeichenketten	128	Zeichenkette zu numerisch konvertieren	37
vorwärts gehen	94	Zeichenketten	128
vorwärts oder rückwärts gehen	94	Zeichenketten verknüpfen	37
wahlweise vorwärts oder rückwärts gehen	94	Zeichenkettenparameter	101
Webseite Festplattenordner	13	Zeichenketten-Variable	36
Webseite Host	13	zeigt die Uhrzeit und das Datum an	154
Webseite mit eigenem Icon ab IE 5.x	92	zeigt eine Bildfolge	160
Webseite per Apache-HTTP-Server	14	zeigt zwei verschiedene Bilder abwechselnd an	159
Webseite Root	13	Zeilenumbruch	91
Wechselnder Text in Statuszeile	115	Zentimeter	35
Werbefenster als FRAME alle 3 Sekunden einblenden	120	Ziffern	127
Wert einem Eingabefeld zuweisen	126	Ziffern-Zeichenketten	128
Wert nicht numerisch	36	Ziffern-Zeichenkettenwert nach numerisch	127
Werte auslesen	124	z-Index	193
Werte konvertieren	127	Zoll	35
Werte überprüfen	126	Zoom	79
Werte zuweisen	126	zu ganzzahlig konvertieren	135
while-Schleife	44	zu ganzzahlig teilbar konvertieren	135
Wieviel Tage sind es noch bis Weihnachten?	154	Zufallspasswort erzeugen	170
win	48	Zufallszahl	134
Win16	51	Zufallszahl erzeugen	104
Win32	51	zurück gehen	94
Win95	51	zusätzliche Event-Handler	149
Win98	51	zusätzlichen Frame anzeigen	95
WinCE	51	zweidimensionale Matrix	35
window.location.hostname	13	Zwischenspeichern von Daten	71
window.status	115		

