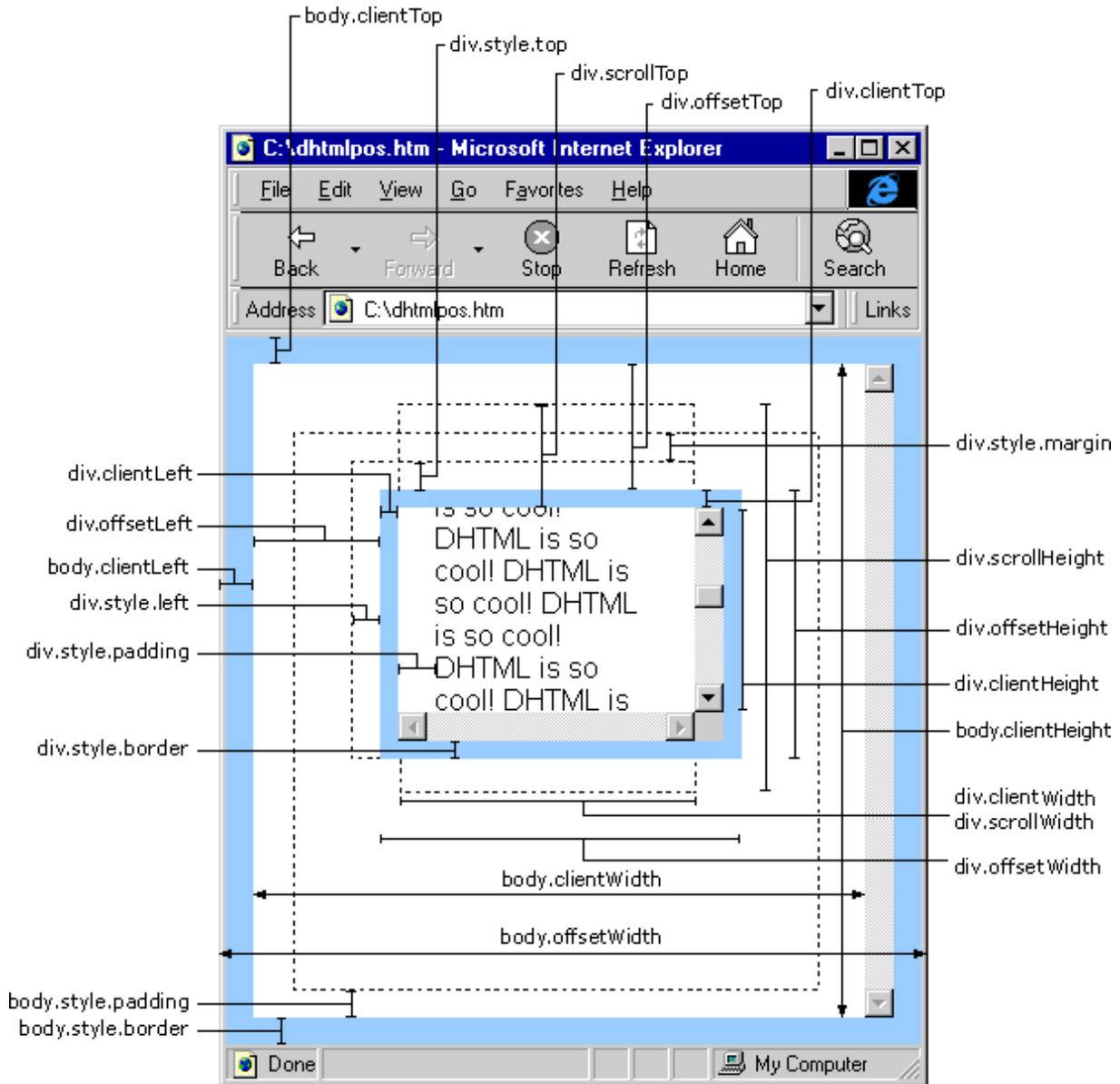
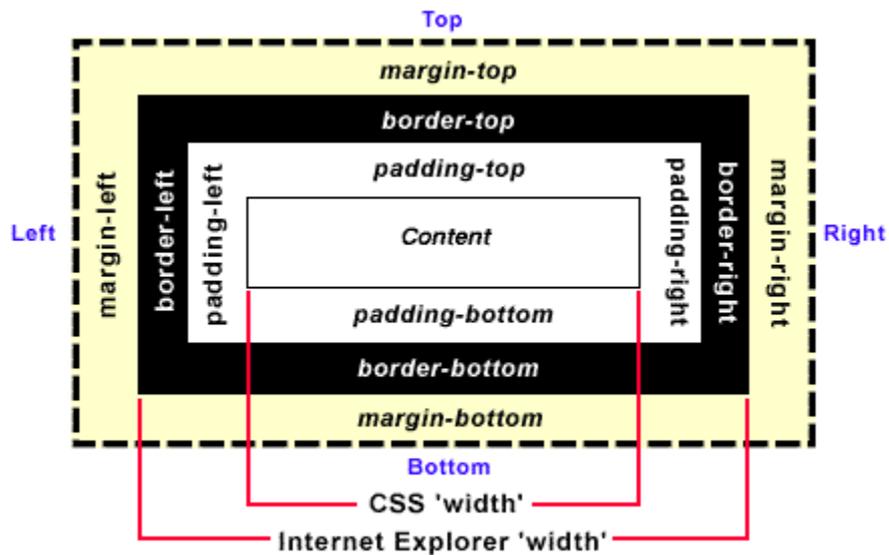
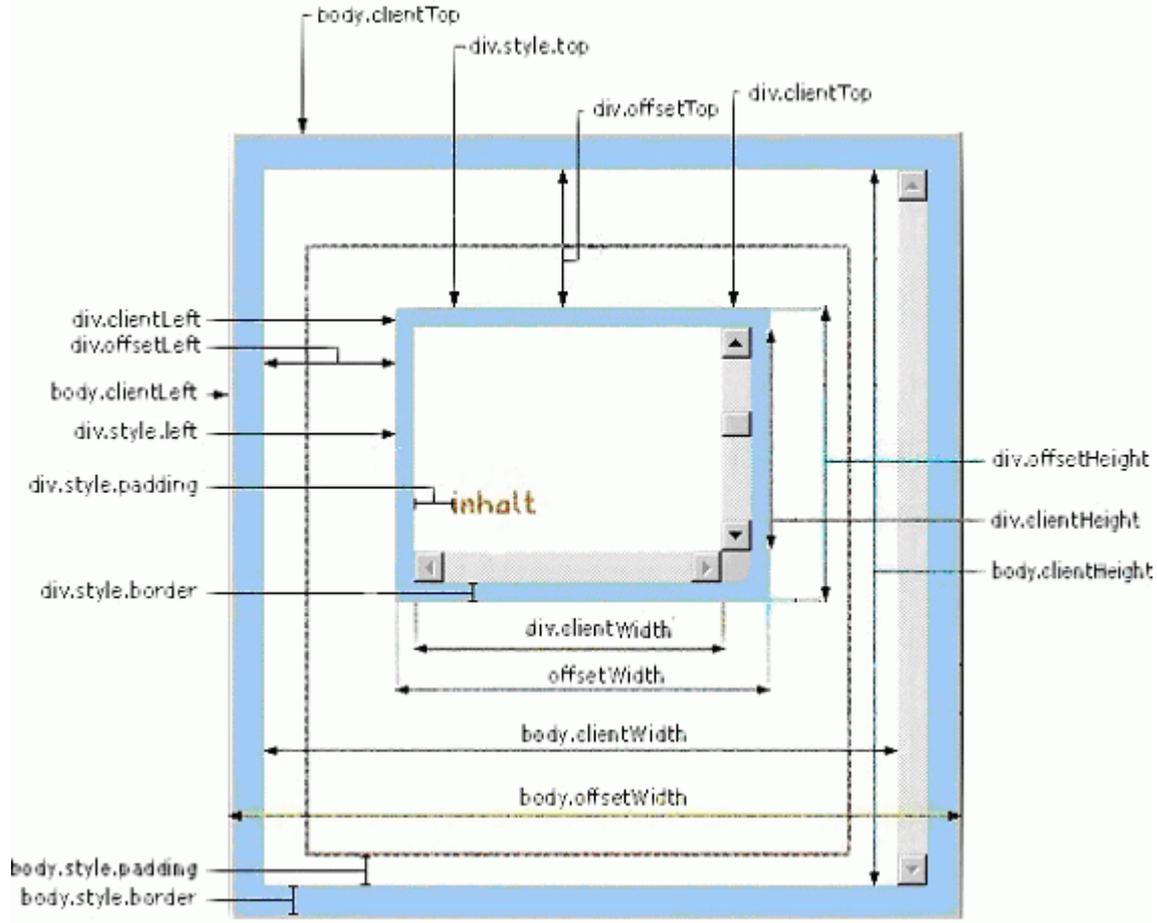


7. Anhang: Styles des Internet Explorer

Der in nachfolgenden Beschreibungen wegen Vereinfachung verwendete Zeiger `object.style.eigenschaft` setzt voraus, dass für `object` eine konkrete Referenz kodiert werden muss.





7.1. Objekte mit STYLE-Attribut

A, ACRONYM, ADDRESS, APPLET, AREA, B, BIG, BLOCKQUOTE, BODY, BR, BUTTON, CAPTION, CENTER, CITE, CODE, COL, COLGROUP, CUSTOM, DD, DEL, DFN, DIR, DIV, DL, DT, EM, EMBED, 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, ISINDEX, KBD, LABEL, LEGEND, LI, LISTING, MAP, 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, XMP



7.2. Style-Eigenschaften - Übersicht

:active Pseudoklasse des a Objektes für aktiven Link (User hat Link aktiviert)
ist nicht per Script ansprechbar

Beispiel:

```
<HEAD>
<STYLE>
A:active { font-weight:bold; color:purple }
</STYLE>
</HEAD>
```

:first Pseudoklasse für @page Regel, siehe auch Objekt page

:first-letter Pseudoklasse für Style des ersten Buchstaben eines HTML-Elementes (Tags)
nicht in Script ansprechbar
nur für Blockelemente wie DIV und SPAN
Elemente mit Eigenschaft .style.display mit Wert "block"
ab IE 5.5

Beispiel:

```
<STYLE>
P.ErsterBuchstabe:first-letter { font-size: 200%; float: left }
</STYLE>
<P CLASS="ErsterBuchstabe">
Testtext
</P>
```

:first-line Pseudoklasse für Style der ersten Zeile eines HTML-Elementes (Tags)
nicht in Script ansprechbar
nur für Blockelemente wie DIV und SPAN
Elemente mit Eigenschaft .style.display mit Wert "block"
ab IE 5.5

Beispiel:

```
<STYLE>
P.ErsteZeile:first-letter { text-transform: uppercase }
</STYLE>
<P CLASS="ErsteZeile">
Testtext Zeile1
<BR>
Testtext Zeile2
</P>
```

:hover Pseudoklasse des a Objektes für Link, der nicht aktiviert wurde
solange Mauszeiger auf dem Link steht, ist die Klasse aktiv
ist nicht per Script ansprechbar

Beispiel:

```
<HEAD>
<STYLE>
A:hover { color:red; text-transform:uppercase; letter-spacing:1cm }
</STYLE>
</HEAD>
```

:left Pseudoklasse für @page Regel, siehe auch Objekt page

:link Pseudoklasse des a Objektes für Link, der nicht kürzlich aktiviert wurde.
ist nicht per Script ansprechbar
ab IE 4.x

Beispiel:

```
<HEAD>
<STYLE>
A:link { color:#FF0000 }
</STYLE>
</HEAD>
```

:right Pseudoklasse für @page Regel, siehe auch Objekt page

:visited Pseudoklasse des a Objektes für Link, der bereits aktiviert wurde und User ist wieder auf das Dokument
zurückgekehrt, das den Link enthält.
ist nicht per Script ansprechbar

Beispiel:

```
<HEAD>
<STYLE>
A:visited { color: blue }
</STYLE>
</HEAD>
```



.style.accelerator ALT+Taste als Kurztastenkombination-Zugriff auf Objekt ein/aus

Beispiel Label mit Tastenkombination

```

<LABEL FOR="ID_InputText">
  <U STYLE="ACCELERATOR:true">
    N
  </U>
  ame
</LABEL>
<INPUT TYPE="text"
  ID="ID_InputText"
  SIZE="25"
  ACCESSKEY="N"
  VALUE="Name"
>

```

.style.background Hintergrund eines Objektes bei P, DIV, SPAN: Eigenschaft nur für eingeschlossenen Text

Beispiel 1

```

<HEAD>
<STYLE>
  .style1 {background:beige url(test.jpg) no-repeat top center}
  .style2 {background:ivory url(test.jpeg) no-repeat bottom right}
</STYLE>
</HEAD>
<BODY>
<SPAN onmouseover="this.className='style1'"
  onmouseout="this.className='style2'"
>
  Testtext
</SPAN>

```

Beispiel 2

```

<SPAN onclick="this.style.background='beige url(test.jpeg) no-repeat top center'">
  Testtext
</SPAN>

```

Beispiel 3 für Hintergrundbild als Wasserzeichen:

```

document.body.style.background="url('test.jpg') white center no-repeat fixed";

```

.style.backgroundAttachment Hintergrundbild nur ab IE 4.x

Beispiel 1:

```

<HEAD>
<STYLE >
  BODY { background-attachment:fixed }
</STYLE>
</HEAD>
<BODY background="test.jpg">
.....

```

Beispiel 2:

```

<BODY ID="ID_Body"
  STYLE="background='marble05.jpg'"
  onload="ID_Body.style.backgroundAttachment = 'fixed'"
>

```

.style.backgroundColor Hintergrundfarbe nur ab IE 4.x
 Achtung: Für das body Objekt gilt:
 BGCOLOR Attribut und Eigenschaft .bgColor für die Hintergrundfarbe im BODY sind deprecated und durch
 <BODY STYLE="background-color:.....">
 zu ersetzen.

Beispiel 1:

```

<SPAN STYLE="font-size:14; background-color:beige">
  Testtext
</SPAN>

```

Beispiel 2:

```

<SPAN onmouseover="this.style.backgroundColor='beige'">

```



```

Testtext
</SPAN>

```

`.style.backgroundImage` Hintergrundbild, das sich über die Hintergrundfarbe legt nur ab IE 4.x

Beispiel 1:

```

<HEAD>
<STYLE>
.setUrl { background-image: url(test.jpg) }
.loseUrl { background-image: url(none) }
</STYLE>
</HEAD>
<BODY>
<SPAN STYLE="font-size:14"
onmouseover="this.className='setUrl'"
onmouseout="this.className='loseUrl'"
>
Testtext
</SPAN>

```

Beispiel 2:

```

<SPAN onmouseover="this.style.backgroundImage='url(test.jpg)'">
Testtext
</SPAN>

```

`.style.backgroundPosition` Hintergrund-Position für Hintergrund z.B. Image (linke obere Ecke)

Beispiel 1:

```

<HEAD>
<STYLE>
.style1 { background-position:top center }
.style2 { background-position:bottom right }
</STYLE>
</HEAD>
<BODY onload=" ID_Span.className='style1'">
<SPAN ID="ID_Span"
STYLE="font-size:14; width:250;"
onmouseover="this.className='style2'"
onmouseout="this.className='style1'"
>
Testtext
</SPAN>

```

Beispiel 2:

```

<SPAN onmouseover="this.style.backgroundPosition='bottom right'">
Testtext
</SPAN>

```

`.style.backgroundPositionX` Hintergrund-Position X für Hintergrund z.B. Image (linke obere Ecke)

`.style.backgroundPositionY` Hintergrund-Position Y für Hintergrund z.B. Image (linke obere Ecke)

`.style.backgroundRepeat` Kachelung des Hintergrundbildes

Beispiel 1:

```

<HEAD>
<STYLE>
.style1 { background-image:url(test.jpg); background-repeat:repeat }
.style2 { background-image:url(test.jpeg); background-repeat:no-repeat }
</STYLE>
</HEAD>
<BODY>
<SPAN onmouseover="this.className='style1'"
onmouseout="this.className='style2'"
onclick="this.className=''"
>
Testtext
</SPAN>

```

Beispiel 2:

```

<SPAN onmouseover="this.style.backgroundImage='url(test.jpeg)'; this.style.backgroundRepeat='repeat'">
Testtext
</SPAN>

```

`.style.behavior` Ort der Behaviors-Datei *.htc (Verhaltensweise eines Elementes) bzw. Standard-Behavior des IE



Beispiel 1:

```
<ELEMENT STYLE="behavior:url(a1.htc) url(a2.htc) ..." >
```

Element steht hier als Platzhalter für ein Tag

Beispiel 2:

```
<UL>
  <LI STYLE="behavior:url(ul.htc) url(hilite.htc)">
    HTML
  </LI>
</UL>
```

Beispiel 3:

```
<HEAD>
<STYLE>
.test_style_klasse {behavior:url(ul.htc) url(hilite.htc)}
</STYLE>
</HEAD>
<BODY>
  <UL>
    <LI CLASS="test_style_klasse">
      HTML
    </LI>
  </UL>
</BODY>
```

Beispiel 4:

```
<SCRIPT>
function window.onload()
{
  ID_Li.style.behavior = "url(ul.htc) url(hilite.htc)";
}
</SCRIPT>
<UL>
  <LI ID="ID_Li" >
    HTML
  </LI>
</UL>
```

Beispiel 5:

```
<HEAD>
<STYLE>
.test_style_klasse { behavior:url(#ID_Object) }
</STYLE>
</HEAD>
<BODY>
  <OBJECT ID="ID_Object" ... ></OBJECT>
  <UL>
    <LI CLASS="test_style_klasse">
      HTML
    </LI>
  </UL>
</BODY>
```

.style.border

Rahmen alle 4 Seiten: Art und Dicke
vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
Objektes besitzen

Beispiel 1:

```
<HEAD>
<STYLE>
.applyBorder { border:0.2cm groove orange }
.removeBorder { border:none }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD
      onmouseover="this.className='applyBorder'"
      onmouseout="this.className='removeBorder'"
    >
      <IMG SRC="test.jpg">
    </TD>
  </TR>
```



</TABLE>

Beispiel 2:

```
<TD onmouseover="this.style.border=0.2cm groove pink">
```

.style.borderBottom

Rahmen unten: Art und Dicke
vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```
<HEAD>
<STYLE>
  TD { border-bottom:0.5cm solid yellow }
  .change { border-bottom:0.5cm groove pink }
</STYLE>
</HEAD>
<BODY>
  <TABLE>
  <TR>
    <TD
      onmouseover="this.className='change'"
      onmouseout="this.className="
    >
      <IMG SRC="test.jpg">
    </TD>
  </TR>
</TABLE>
</BODY>
```

Beispiel 2:

```
<TD onmouseover="this.style.borderBottom=0.3cm groove yellow">
```

.style.borderBottomColor

Rahmen unten: Farbe
vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```
<HEAD>
<STYLE>
  TD { border-bottom-color: red; border-width: 0.5cm; border-style: groove; }
  .blue { border-bottom-color: blue; }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD
      onmouseover="this.className='blue'"
      onmouseout="this.className="
    >
      <IMG SRC="test.jpg">
    </TD>
  </TR>
</TABLE>
</BODY>
```

Beispiel 2:

```
<TD onmouseover="this.style.borderWidth=0.5cm; this.style.borderBottomColor='blue'">
```

.style.borderBottomStyle

Rahmen unten: Art
vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```
<HEAD>
<STYLE>
  TD { border-bottom-style:solid; border-width=0.3cm }
  .change { border-bottom-style:groove }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD
      onmouseover="this.className='change'"
      onmouseout="this.className="
    >
      <IMG SRC="test.jpg">
    </TD>
  </TR>
```



</TABLE>

Beispiel 2:

<TD onmouseover="this.style.borderWidth='0.5cm'; this.style.borderBottomStyle='groove'">

.style.borderBottomWidth Rahmen unten: Dicke
vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-bottom-width:3mm }
  .changeborder1 { border-bottom-width:1cm }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
    <TR>
      <TD
        onclick="this.className='changeborder1'"
        ondblclick="this.className=''"
      >
        <IMG SRC="test.jpg">
      </TD>
    </TR>
  </TABLE>
</BODY>

```

Beispiel 2:

<TD onclick="this.style.borderBottomWidth='1cm'">

.style.borderCollapse Rahmen alle 4 Seiten: Verschmelzung zum Single Border

Beispiel:

```

<TABLE ID="ID_Tabelle" STYLE="border-collapse:collapse">
.....
</TABLE>
<INPUT TYPE=button VALUE="separate" onclick="ID_Tabelle.style.borderCollapse='separate'">
<INPUT TYPE=button VALUE="collapse" onclick="ID_Tabelle.style.borderCollapse='collapse'">

```

.style.borderColor Rahmen alle 4 Seiten: Farbe
vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-color: red; border-width: 0.5cm }
  .blue { border-color: blue }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
    <TR>
      <TD
        onmouseover="this.className='blue'"
        onmouseout="this.className=''"
      >
        <IMG SRC="test.jpg">
      </TD>
    </TR>
  </TABLE>
</BODY>

```

Beispiel 2:

<TD onmouseover="this.style.borderWidth='0.5cm'; this.style.borderColor='blue'; this.style.borderStyle='solid'">

.style.borderLeft Rahmen links: Art und Dicke
vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-left:0.5cm solid yellow }
  .change { border-left:0.5cm groove pink }
</STYLE>
</HEAD>
<BODY>

```



```

<TABLE>
<TR>
  <TD   onmouseover="this.className='change'"
        onmouseout="this.className=''"
        >
        <IMG SRC="test.jpg">
  </TD>
</TR>
</TABLE>
</BODY>

```

Beispiel 2:

```

<TD onmouseover="this.style.borderLeft=0.3cm groove yellow">

```

.style.borderLeftColor Rahmen links: Farbe
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
 Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-left-color: red; border-width: 0.5cm; border-style: groove; }
  .blue { border-left-color: blue; }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD   onmouseover="this.className='blue'"
          onmouseout="this.className=''"
          >
          <IMG SRC="test.jpg">
    </TD>
  </TR>
  </TABLE>
</BODY>

```

Beispiel 2:

```

<TD onmouseover="this.style.borderWidth=0.5cm'; this.style.borderLeftColor='blue'">

```

.style.borderLeftStyle Rahmen links: Art
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
 Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-left-style:solid; border-width=0.3cm }
  .change { border-left-style:groove; }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD   onmouseover="this.className='change'"
          onmouseout="this.className=''"
          >
          <IMG SRC="test.jpg">
    </TD>
  </TR>
  </TABLE>

```

Beispiel 2:

```

<TD onmouseover="this.style.borderWidth=0.5cm'; this.style.borderLeftStyle='groove'">

```

.style.borderLeftWidth Rahmen links: Dicke
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
 Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-left-width:3mm }
  .changeborder1 { border-left-width:1cm }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>

```



```

        <TR>
            <TD
                onclick="this.className='changeborder1'"
                ondblclick="this.className=''"
            >
                <IMG SRC="test.jpg">
            </TD>
        </TR>
    </TABLE>
</BODY>

```

Beispiel 2:

```
<TD onclick="this.style.borderLeftWidth='1cm'">
```

.style.borderRight Rahmen rechts: Art und Dicke
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
 Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
    TD { border-right:0.5cm solid yellow }
    .change { border-right:0.5cm groove pink }
</STYLE>
</HEAD>
<BODY>
    <TABLE>
    <TR>
        <TD
            onmouseover="this.className='change'"
            onmouseout="this.className=''"
        >
            <IMG SRC="test.jpg">
        </TD>
    </TR>
    </TABLE>
</BODY>

```

Beispiel 2:

```
<TD onmouseover="this.style.borderRight='0.3cm groove yellow'">
```

.style.borderRightColor Rahmen rechts: Farbe
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
 Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
    TD { border-right-color: red; border-width: 0.5cm; border-style: groove; }
    .blue { border-right-color: blue; }
</STYLE>
</HEAD>
<BODY>
    <TABLE BORDER>
    <TR>
        <TD
            onmouseover="this.className='blue'"
            onmouseout="this.className=''"
        >
            <IMG SRC="test.jpg">
        </TD>
    </TR>
    </TABLE>
</BODY>

```

Beispiel 2:

```
<TD onmouseover="this.style.borderWidth='0.5cm'; this.style.borderRightColor='blue'">
```

.style.borderRightStyle Rahmen rechts: Art
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
 Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
    TD { border-right-style:solid; border-width=0.3cm }
    .change { border-right-style:groove; }
</STYLE>
</HEAD>
<BODY>

```



```

<TABLE BORDER>
  <TR>
    <TD
      onmouseover="this.className='change'"
      onmouseout="this.className=''"
    >
      <IMG SRC="test.jpg">
    </TD>
  </TR>
</TABLE>

```

Beispiel 2:

```

<TD onmouseover="this.style.borderWidth='0.5cm'; this.style.borderRightStyle='groove'">

```

.style.borderRightWidth Rahmen rechts: Dicke
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-right-width:3mm }
  .changeborder1 { border-right-width:1cm }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
    <TR>
      <TD
        onclick="this.className='changeborder1'"
        ondblclick="this.className=''"
      >
        <IMG SRC="test.jpg">
      </TD>
    </TR>
  </TABLE>
</BODY>

```

Beispiel 2:

```

<TD onclick="this.style.borderRightWidth='1cm'">

```

.style.borderStyle Rahmen alle 4 Seiten: Art
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-style:solid; border-width:0.3cm }
  .change { border-style:groove }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
    <TR>
      <TD
        onmouseover="this.className='change'"
        onmouseout="this.className=''"
      >
        <IMG SRC="test.jpg">
      </TD>
    </TR>
  </TABLE>

```

Beispiel 2:

```

<TD onmouseover="this.style.borderWidth='0.5cm'; this.style.borderStyle='groove'">

```

.style.borderTop Rahmen oben: Art und Dicke
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-top:0.5cm solid yellow }
  .change { border-top:0.5cm groove pink }
</STYLE>
</HEAD>
<BODY>
  <TABLE>
    <TR>
      <TD
        onmouseover="this.className='change'"

```



```

onmouseout="this.className=""
>
<IMG SRC="test.jpg">
</TD>
</TR>
</TABLE>
</BODY>

```

Beispiel 2:

```
<TD onmouseover="this.style.borderTop=0.3cm groove yellow">
```

.style.borderTopColor Rahmen oben: Farbe
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-top-color: red; border-width: 0.5cm; border-style: groove; }
  .blue { border-top-color: blue; }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD
      onmouseover="this.className='blue'"
      onmouseout="this.className=""
    >
      <IMG SRC="test.jpg">
    </TD>
  </TR>
</TABLE>
</BODY>

```

Beispiel 2:

```
<TD onmouseover="this.style.borderWidth='0.5cm'; this.style.borderTopColor='blue'">
```

.style.borderTopStyle Rahmen oben: Art
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-top-style:solid; border-width=0.3cm }
  .change { border-top-style:groove; }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD
      onmouseover="this.className='change'"
      onmouseout="this.className=""
    >
      <IMG SRC="test.jpg">
    </TD>
  </TR>
</TABLE>

```

Beispiel 2:

```
<TD onmouseover="this.style.borderWidth='0.5cm'; this.style.borderTopStyle='groove'">
```

.style.borderTopWidth Rahmen oben: Dicke
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-top-width:3mm }
  .changeborder1 { border-top-width:1cm }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD
      onclick="this.className='changeborder1'"
      ondblclick="this.className=""
    >

```



```

    >
    </TD> <IMG SRC="test.jpg">
  </TR>
</TABLE>
</BODY>

```

Beispiel 2:

```
<TD onclick="this.style.borderTopWidth='1cm'">
```

.style.borderWidth Rahmen alle 4 Seiten: Dicke
 vor IE 5.5: Objekt muss absolut positionierte sein, oder Angabe zu Höhe bzw. Breite des
 Objektes besitzen

Beispiel 1:

```

<HEAD>
<STYLE>
  TD { border-width:3mm }
  .changeborder1 { border-width:1cm }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
  <TR>
    <TD      onclick="this.className='changeborder1'"
    <TD      ondblclick="this.className=''"
    >
    <IMG SRC="test.jpg">
  </TD>
  </TR>
</TABLE>
</BODY>

```

Beispiel 2:

```
<TD onclick="this.style.borderWidth='1cm'">
```

.style.bottom Abstand unterer Objektrand zur Umgebung Oberkante
 Objekt muss position-Eigenschaft kodiert haben
 identisch mit Attribut **.style.pixelBottom** und **.style.posBottom**
 aber **.style.bottom:** String
 .style.pixelBottom: Integer, nur Pixel
 .style.posBottom: Float pointing und Integer

Beispiel:

```

<DIV STYLE = "position:absolute; bottom:50px">
  ...
</DIV>

```

.style.clear Textumfluss links, rechts ("left", "right", "both", "none")
 siehe auch Eigenschaften **.style.whiteSpace**
 .style.float
 .style.styleFloat

Beispiel 1:

```

<HEAD>
<STYLE>
  I { clear:left }
</STYLE>
</HEAD>

```

Beispiel 2:

```

<HEAD>
<SCRIPT>
  function Wechsel1()
  { ID_Span.style.clear="left";}

  function Wechsel2()
  { ID_Span.style.clear="none";}
</SCRIPT>
</HEAD>
<BODY>
  <IMG SRC ="test.jpg" STYLE="float:left">
  <SPAN ID="ID_Span">
    TestText
  </SPAN>
  <INPUT TYPE=button VALUE="clear = left" onclick="Wechsel1()">

```



```
<INPUT TYPE=button VALUE="clear = none" onclick="Wechsel2()">
</BODY>
```

.style.clip

Position sichtbarer Bereich über dem Objekt
 Position immer bezüglich Umgebung
 Objekt muss absolut positioniert sein !
 ausserhalb des Clipfensters wird alles transparent
 siehe auch Eigenschaften

- .style.overflow
- .style.overflowX
- .style.overflowY
- .style.display
- .style.visibility

Beispiel 1:

```
<DIV STYLE="position:absolute;top:0;left:200; clip:rect(15px auto auto auto)">
  <IMG SRC="test.jpg">
</DIV>
```

Beispiel 2:

```
<IMG ID="ID_IMG" SRC="test.jpg" STYLE="position:absolute;top:0cm;left:0cm;">
<BUTTON onclick="ID_IMG.style.clip=rect(0.2cm 0.6cm 1cm 0.1cm)">
  Sichtbarkeitsbereich über Image
</BUTTON>
```

.style.color

Vordergrundfarbe (Textfarbe)

Beispiel 1:

```
<HEAD>
<STYLE>
  .color1 { color:red }
  .color2 { color:gray }
</STYLE>
</HEAD>
<BODY>
  <SPAN STYLE="font-size:14"
    onmouseover="this.className='color1'"
    onmouseout="this.className='color2'"
  >
    Testtext
  </SPAN>
</BODY>
```

Beispiel 2:

```
<SPAN STYLE="font-size:14" onmouseover="this.style.color='red'">
  Testtext
</SPAN>
```

Beispiel 3 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 Schritte 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
                    / f arbwechsel_anzahl
                )
        }

        if(farbwechsel_zahler >= farbwechsel_anzahl)
        {farbwert_wechselrichtung = -1}
        else
        {
            if(farbwechsel_zahler <= 1)
            {farbwert_wechselrichtung = 1}
        }

        farbwechsel_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 + "')", farbwechsel_tempo);
    }

    function start(
        ueberschrift_id,
        startfarbe_rot, startfarbe_gruen, startfarbe_blaue,
        endfarbe_rot, endfarbe_gruen, endfarbe_blaue
    )
    {
        farbwert_start[0] = startfarbe_rot;
        farbwert_start[1] = startfarbe_gruen;
        farbwert_start[2] = startfarbe_blaue;
        farbwert_ende[0] = endfarbe_rot;
        farbwert_ende[1] = endfarbe_gruen;
        farbwert_ende[2] = endfarbe_blaue;
        farbwechsel_zahler = 0;
        farbewechseln(ueberschrift_id);
    }
}
//-->
</SCRIPT>
</HEAD>

<BODY onLoad="start('farbwechsel', 0, 255, 0, 255, 0, 255)">
    <H1 ID="farbwechsel">&Uuml;berschrift</H1>
</BODY>
</HTML>

```

.style.cssText Wert des STYLE-Attributes im Tag des HTML-Elementes

Beispiel:

```
<P ID="ID_P" STYLE="color:'green'; font-weight:bold">
    Testtext
```

```
</P>
```

```
<BUTTON onclick="alert(ID_P.style.cssText)">
    Anzeigen
```



```
</BUTTON>
```

`.style.cursor` Mauscursor Arten
z.T. erst ab IE 6.x

Beispiel 1:

```
<HEAD>
<STYLE>
P { cursor : hand; }
</STYLE>
</HEAD>
```

Beispiel 2:

```
<P onmouseover="this.style.cursor='hand'">
```

Beispiel 3:

```
zeiger_auf_objekt.style.cursor = url('mycursor.cur');
```

Hinweis: `style.cursor` ist standardgemäß mit einer Leerkette belegt, solange **kein** Cursor gesetzt wird kann nicht mit `url(cursor_form)` belegt werden, wenn es sich um eine **standardgemäß im Browser implementierte** Cursorform handelt wie z.B. 'hand' oder 'normal'. Grund: Diese Cursorformen sind keine Dateien, obwohl alert den String 'url(cursor_form)' anzeigt.

`.style.direction` Horizontale Richtung des Renderns z.B. vom Plain-Textteil eines Elementes, also Leserichtung für den User einstellen (z.B. von rechts nach links)
siehe auch Eigenschaft `.style.unicodeBidi`

Beispiel:

```
<SCRIPT>
function Aendern()
{
// den Text des geklickten Ankers als Direction
// dem Div zuweisen
ID_Div.style.direction = event.srcElement.innerText;

// und den Text des Ankers dem Span als Text zuweisen
ID_Span.innerText = ID_Div.style.direction;
}
</SCRIPT>
<SPAN ID="ID_Span" style="color:red">
Testtext
</SPAN>
```

Bitte einen der Anker waehlen !

```
<A href="#" onclick=Aendern()>ltr</A>
<A href="#" onclick=Aendern()>rtl</A>
<A href="#" onclick=Aendern()>inherit</A>
```

```
<DIV ID="ID_Div">
0123456789
</DIV>
```

`.style.display` Art der Anzeige (des Renderns) eines Objektes (auch mit Ausrichtung)
siehe auch Eigenschaften `.style.overflow`
`.style.overflowX`
`.style.overflowY`
`.style.clip`
`.style.visibility`
z.T. ab IE 6.x
Hinweis: Wenn !DOCTYPE am Dokumentanfang kodiert wurde, so darf `.style.display` **nicht** auf "list-item" gesetzt sein.
Wenn **nicht** !DOCTYPE am Dokumentanfang kodiert wurde, so **kann** `.style.display` auf "list-item" gesetzt sein
!DOCTYPE am Dokumenten-Anfang ist erst ab IE 6.x kodierbar !

Beispiel 1:

```
<SPAN ID="ID_Span" >
Testtext
</SPAN>
<INPUT TYPE=button VALUE="Block" onclick=" ID_Span.style.display='block'">
```

Beispiel 2:

```
<SCRIPT>
```



```

function Aendern()
{
    ID_Zelle1.style.display="none";
    ID_Zelle2.style.display="block";
    ID_Zelle3.style.display="none";
}
</SCRIPT>
<TABLE>
<TR>
<TD ID="ID_Zelle1">
    Zelle1
</TD>
</TR>
<TR>
<TD ID="ID_Zelle2">
    Zelle2
</TD>
</TR>
<TR>
<TD ID="ID_Zelle3">
    Zelle3
</TD>
</TR>
</TABLE>
<INPUT TYPE=button VALUE="Aendern" onclick="Aendern()">

```

.style.filter Filter des Internet Explorer implementieren (siehe Objekt filter)

Beispiel Filter lesen

```

<SCRIPT>
alert(ID_Div.style.filter);
// zeigt den String
// "filter:progid:DXImageTransform.Microsoft.Wave(strength=100)
// progid:DXImageTransform.Microsoft.CheckerBoard(duration=4)"
// an
</SCRIPT>
<DIV ID="ID_Div" STYLE="height:50;width:50;
    filter:progid:DXImageTransform.Microsoft.Wave(strength=100)
    progid:DXImageTransform.Microsoft.CheckerBoard(duration=4)"
>
    Test-Text
</DIV>

```

Beispiel Filter schreiben

```

<DIV ID="ID_Div" STYLE="height:50;width:50;filter:
    progid:DXImageTransform.Microsoft.Wave(strength=100)
    progid:DXImageTransform.Microsoft.BasicImage(rotation=2, mirror=1)"
>
    Test-Text
</DIV>
<SCRIPT>
// hinzufügen
ID_Div.style.filter += "progid:DXImageTransform.Microsoft.Iris(irisstyle='STAR',duration=4)"
</SCRIPT>

```

float Textumfluss links, rechts
 nur im STYLE-Attribut des HTML-Elementes kodierbar
 per Script nicht ansprechbar
 unter IE 5.x : DIV und SPAN müssen Eigenschaft .style.width besitzen
 siehe auch Eigenschaften .style.whiteSpace
 .style.clear
 .style.styleFloat

Beispiel:

```

<IMG SRC="test.jpg" STYLE="float:left">

```

.style.font Textfont komplett mit allen Teileigenschaften (ebenfalls Style-Eigenschaften) oder Standard-Textfont

Beispiel 1:

```

<SPAN STYLE="font:italic normal bold 12pt Arial">
    Testtext
</SPAN>

```



Beispiel 2

```
<DIV onmouseover="this.style.font = 'italic small-caps bold 12pt serif'">
  Testtext
</DIV>
```

.style.fontFamily Textfont Familie

Beispiel 1:

```
<HEAD>
<STYLE>
  P { font-family:"Arial"}
  .other { font-family:"Courier"}
</STYLE>
</HEAD>
```

Beispiel 2:

```
<DIV onmousedown="this.style.fontFamily='Courier'">
```

.style.fontSize Textfont Höhe

Beispiel 1:

```
<HEAD>
<STYLE>
  BODY {font-size: 10pt }
  .P1 {font-size: 14pt }
  .P2 {font-size: 75% }
  .P3 {font-size: xx-large }
  .P4 {font-size: larger }
</STYLE>
</HEAD>
```

Beispiel 2:

```
<DIV STYLE="font-size:12pt" onmouseover="this.style.fontSize='14pt'">
  Testtext
</DIV>
```

.style.fontStyle Textfont Stil

Beispiel 1:

```
<HEAD>
<STYLE>
  H3 { font-style:italic }
</STYLE>
</HEAD>
```

Beispiel 2:

```
<DIV onmousedown="this.style.fontStyle='italic'">
```

.style.fontVariant Textfont Stil-Variante

Beispiel 1:

```
<P STYLE="font-variant:small-caps">
```

Beispiel 2:

```
<DIV onmousedown="this.style.fontVariant='small-caps'">
```

.style.fontWeight Textfont Fettheit

Beispiel 1:

```
<HEAD>
<STYLE>
  LI { font-weight:bolder }
</STYLE>
</HEAD>
```

Beispiel 2:

```
<P STYLE="font-size:14" onmouseover="this.style.fontWeight='bolder'">
```

.style.height Höhe Objekt
identisch mit .style.pixelHeigth und .style.posHeight, aber
.style.heigt: String
.style.pixelHeigt: Integer, nur Pixel
.style.posHeigth: Float pointing und Integer
Hinweis: Wenn !DOCTYPE am Dokumentanfang kodiert wurde, so kann
.style.heigt auf "auto "



gesetzt sein
Wenn **nicht** !DOCTYPE am Dokumentanfang kodiert wurde, so **darf**
.style.height **nicht** auf "auto "
gesetzt sein
!DOCTYPE am Dokumenten-Anfang ist erst ab IE 6.x kodierbar !

Beispiel 1:

Beispiel 2:
<BUTTON onclick="height1.style.height='1cm'">Test</BUTTON>

.style.imeMode Status des Input Method Editor (IME) für Eingabe von Chinese, Japanese oder Korean Zeichen

Beispiel:
<INPUT TYPE = text STYLE = "ime-mode:active" >

.style.layoutFlow deprecated
zu ersetzen durch Eigenschaften .style.writingMode
Ost-Asiatische Darstellung (Flussrichtung)
siehe auch Eigenschaft .style.verticalAlign für Ausrichtung vertikal für Objekte mit VALIGN-Attribut

.style.layoutGrid Textzeichen-Layout-Gitter z.B. für asiatische Sprachen

Beispiel:
<HEAD>
<STYLE>
DIV.layout { layout-grid: both fixed 12px 12px }
</STYLE>
</HEAD>
<BODY>
<DIV CLASS = "layout">
 Testtext
</DIV>
</BODY>

.style.layoutGridChar Textzeichen-Layout-Gitter Zeichengröße
nur für Blockelemente wie SPAN und DIV etc.
verlangt .style.layoutGridMode auf Wert "line" oder "both"

Beispiel:
<HEAD>
<STYLE>
DIV.layout { layout-grid-char: auto }
</STYLE>
</HEAD>
<BODY>
<DIV CLASS = "layout">
 Testtext
</DIV>
</BODY>

.style.layoutGridLine Textzeichen-Layout-Gitter Linie
nur für Blockelemente wie SPAN und DIV etc.
verlangt .style.layoutGridMode auf Wert "line" oder "both"

Beispiel:
<HEAD>
<STYLE>
DIV.layout { layout-grid-line: auto }
</STYLE>
</HEAD>
<BODY>
<DIV CLASS = "layout">
 Testtext
</DIV>
</BODY>

.style.layoutGridMode Textzeichen-Layout-Gitter 2D

Beispiel:
<HEAD>
<STYLE>
DIV.layout { layout-grid-mode: line }
</STYLE>
</HEAD>
<BODY>



```

<DIV CLASS = "layout">
  Testtext
</DIV>
<BODY>

```

.style.layoutGridType Textzeichen-Layout-Gitter Typ
 nur für Block-Elemente wie SPAN, DIV

Beispiel:

```

<HEAD>
<STYLE>
  DIV.layout { layout-grid-type: strict }
</STYLE>
</HEAD>
<BODY>
  <DIV CLASS = "layout">
    Testtext
  </DIV>
</BODY>

```

.style.left Abstand linker Objektrand zur Umgebung rechte Kante
 Abstand des linken Objektrandes des Kindes zum linken Objektrand des Elternelementes,
wenn im Elternelement STYLE="position:....." **kodiert wurde** (Wert von position ist egal)
 Bsp.: IMG als Kind eines DIV (IMG liegt innerhalb eines DIV):

Abstand der linken Kante des IMG zur linken Kante des DIV
 Objekt muss Eigenschaft .style.position kodiert haben
 identisch zu .style.pixelLeft und .style.posLeft
 aber .style.left: String
 .style.pixelLeft: Integer, nur Pixel
 .style.posLeft: Float pointing und Integer

Beispiel 1:

```

<DIV STYLE="position:absolute;left:100px">
  <IMG SRC="test.jpg">
</DIV>

```

Beispiel 2:

```

<BUTTON onclick="cone.style.left='100px'; sphere.style.left='200px'">
  ...
</BUTTON>

```

.style.letterSpacing Textzeichen Abstand

Beispiel 1:

```

<HEAD>
<STYLE>
  BLOCKQUOTE { letter-spacing:-0.2mm }
</STYLE>
</HEAD>

```

Beispiel 2:

```

<DIV STYLE="font-size:14" onmouseover="this.style.letterSpacing='1mm'">
  ....
</DIV>

```

.style.lineBreak Zeilenumbruch normal

.style.lineHeight Abstand zweier Objekte oder zweier Textzeilen
 ab IE 4.x

Beispiel 1:

```

<HEAD>
<STYLE>
  P { line-height:8mm}
  BLOCKQUOTE { line-height:4mm }
</STYLE>
</HEAD>

```

Beispiel 2:

```

<DIV STYLE="font-size:14" onmouseover="this.style.lineHeight='6mm'">
  .....
</DIV>

```

.style.listStyle Listendarstellung Aufzählungsliste-Elemente
 benötigt linken Abstand zum linken Objekt von mindestens 30 Bildpunkten

Beispiel 1:

```

<HEAD>

```




```

<BODY>
  <HR onclick="this.className='margin1'" ondblclick="this.className=''>
</STYLE>

```

Beispiel 2:

```

<HR style="margin-top: 2cm;"
ondblclick="this.style.marginTop=''"
>

```

.style.minHeight minimale Höhe des Objektes
 ab IE 6.x **und** nur wenn **nicht** !DOCTYPE am Dokumentanfang kodiert wurde
 und dann auch nur für TD, TH und TR innerhalb eines fixierten Tabellenlayouts
 Tabelle mit fixiertem Layout erzeugen per Eigenschaft .style.tableLayout mit Wert "fixed"
 wird schneller dargestellt als bei Auto-Layout
 Hinweis: Standard bei Tabellen Eigenschaft .style.tableLayout mit Wert "auto"
 also Auto-Layout, also kein fixiertes Tabellenlayout

.style.onOffBehavior deprecated ab IE 5.x
 Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound

.style.overflow Anzeige des Objektes, wenn Objektdimension in Höhe und/ oder Breite größer als das Elternfenster,
 also Überlauf der Anzeige gegenüber Elternfenster.
 Anzeige des Objektes mit/ohne Scrollelemente vertikal und horizontal
 Anzeige des Objektes ein/aus
 siehe auch Eigenschaften .style.overflowX
 .style.overflowY
 .style.clip
 .style.display
 .style.visibility
 .style.textOverflow
 ab IE 5.x

Beispiel 1

```

<DIV STYLE="width: 200px; height: 200px; overflow: auto;">
  Testtext
</DIV>

```

Beispiel 2

```

<SCRIPT>
  function AnzeigeArtSetzen(ObjectZeiger, AnzeigeArtAlsKette)
  {
    ObjectZeiger.style.overflow = AnzeigeArtAlsKette;
  }
</SCRIPT>
<DIV ID="ID_Div" STYLE="font-size: 18pt; background-color: yellow; height: 50px; width: 75px">
  Testtext
</DIV>
<SELECT onchange="AnzeigeArtSetzen(ID_Div, this.options[this.selectedIndex].text)">
  <OPTION SELECTED>visible
  <OPTION>scroll
  <OPTION>hidden
  <OPTION>auto
</SELECT>

```

.style.overflowX Anzeige des Objektes, wenn Objektdimension in Höhe und/ oder Breite größer als das Elternfenster,
 also Überlauf der Anzeige gegenüber Elternfenster.
 Anzeige des Objektes mit/ohne Scrollelemente nur horizontal
 Anzeige des Objektes ein/aus
 siehe auch Eigenschaften .style.overflow
 .style.overflowY
 .style.clip
 .style.display
 .style.visibility
 .style.textOverflow
 ab IE 5.x

.style.overflowY Anzeige des Objektes, wenn Objektdimension in Höhe und/ oder Breite größer als das Elternfenster,
 also Überlauf der Anzeige gegenüber Elternfenster.
 Anzeige des Objektes mit/ohne Scrollelemente nur vertikal
 Anzeige des Objektes ein/aus
 siehe auch Eigenschaften .style.overflow
 .style.overflowX
 .style.clip
 .style.display
 .style.visibility
 .style.textOverflow



ab IE 5.x

`.style.padding` Abstand links, rechts, oben, unten zwischen Objekt und Margin bzw. Rahmen
 ab IE 6.x auch für `img` Objekt
 unter IE 5.5 nur für Elemente die absolut positioniert sind oder Angaben zu Höhe und Breite haben
 Margin: Abstand des Aussenrandes eines Objektes zur Umgebung
 Padding: Abstand des Objektinhaltes zum Aussenrand

Beispiel 1:

```
<HEAD>
<STYLE>
  TD { padding:3mm 8mm }
  .padding1 { padding:1cm }
</STYLE>
</HEAD>
<BODY>
  <TABLE BORDER>
    <TD ALIGN=middle
      onmouseover="this.className='padding1'"
      onmouseout="this.className=''"
    >
      <IMG SRC="test.jpg">
    </TD>
  </TABLE>
</BODY>
```

Beispiel 2:

```
<TABLE BORDER>
  <TD ALIGN=middle
    onmouseover="this.style.padding='0.5cm 0.2cm'"
    onmouseout="this.style.padding=''"
  >
    <IMG SRC="test.jpeg">
  </TD>
</TABLE>
```

`.style.paddingBottom` Abstand unten zwischen Objekt und Margin bzw. Rahmen
 ab IE 6.x auch für `img` Objekt
 unter IE 5.5 nur für Elemente die absolut positioniert sind oder Angaben zu Höhe und Breite haben
 Margin: Abstand des Aussenrandes eines Objektes zur Umgebung
 Padding: Abstand des Objektinhaltes zum Aussenrand

Beispiel 1:

```
<HEAD>
<STYLE>
  TD { padding-bottom:1cm }
</STYLE>
</HEAD>
```

Beispiel 2:

```
<TABLE BORDER>
  <TD onmouseover="this.style.paddingBottom='1cm'"
    onmouseout="this.style.paddingBottom=''"
  >
    <IMG SRC="test.jpeg">
  </TD>
</TABLE>
```

`.style.paddingLeft` Abstand links zwischen Objekt und Margin bzw. Rahmen
 ab IE 6.x auch für `img` Objekt
 unter IE 5.5 nur für Elemente die absolut positioniert sind oder Angaben zu Höhe und Breite haben
 Margin: Abstand des Aussenrandes eines Objektes zur Umgebung
 Padding: Abstand des Objektinhaltes zum Aussenrand

Beispiel 1:

```
<HEAD>
<STYLE>
  TD { padding-left:1cm }
</STYLE>
</HEAD>
```

Beispiel 2:

```
<TABLE BORDER>
  <TD onmouseover="this.style.paddingLeft='1cm'"
    onmouseout="this.style.paddingLeft=''"
  >
```



```

        <IMG SRC="test.jpeg">
    </TD>
</TABLE>

```

.style.paddingRight Abstand rechts zwischen Objekt und Margin bzw. Rahmen
 ab IE 6.x auch für img Objekt
 unter IE 5.5 nur für Elemente die absolut positioniert sind oder Angaben zu Höhe und Breite haben
 Margin: Abstand des Aussenrandes eines Objektes zur Umgebung
 Padding: Abstand des Objektinhaltes zum Aussenrand

Beispiel 1:

```

<HEAD>
<STYLE>
    TD { padding-right: 1cm }
</STYLE>
</HEAD>

```

Beispiel 2:

```

<TABLE BORDER>
  <TD onmouseover="this.style.paddingRight='1cm'"
    onmouseout="this.style.paddingRight=''"
    >
    <IMG SRC="test.jpeg">
  </TD>
</TABLE>

```

.style.paddingTop Abstand oben zwischen Objekt und Margin bzw. Rahmen
 ab IE 6.x auch für img Objekt
 unter IE 5.5 nur für Elemente die absolut positioniert sind oder Angaben zu Höhe und Breite haben
 Margin: Abstand des Aussenrandes eines Objektes zur Umgebung
 Padding: Abstand des Objektinhaltes zum Aussenrand

Beispiel 1:

```

<HEAD>
<STYLE>
    TD { padding-top: 1cm }
</STYLE>
</HEAD>

```

Beispiel 2:

```

<TABLE BORDER>
  <TD onmouseover="this.style.paddingTop='1cm'"
    onmouseout="this.style.paddingTop=''"
    >
    <IMG SRC="test.jpeg">
  </TD>
</TABLE>

```

.style.pageBreakAfter Seitenumbruch nach dem Druck des Objektes beim Druck des Dokumentes
 nicht möglich für BR und HR-Objekt
 siehe auch Eigenschaften `.style.whiteSpace`
`.style.wordBreak`
`.style.wordWrap`
`.style.pageBreakBefore`

Beispiel 1:

```

<HEAD>
<STYLE>
    P { page-break-after: always }
</STYLE>
</HEAD>
<BODY>
<P ID="ID_P">
    .....
</P>

```

Beispiel 2:

```

<SCRIPT>
    function KeinSeitenumbruch()
    {ID_P.style.pageBreakAfter='';}
</SCRIPT>
<BUTTON onClick="KeinSeitenumbruch()">Kein Seitenumbruch </BUTTON>

```

.style.pageBreakBefore Seitenumbruch vor dem Druck des Objektes beim Druck des Dokumentes
 nicht möglich für BR und HR-Objekt
 siehe auch Eigenschaften `.style.whiteSpace`
`.style.wordBreak`



```
.style.wordWrap
.style.pageBreakAfter
```

Beispiel 1:

```
<HEAD>
<STYLE>
  P { page-break-before: always }
</STYLE>
</HEAD>
<BODY>
<P ID="ID_P">
  .....
</P>
```

Beispiel 2:

```
<SCRIPT>
  function KeinSeitenbruch()
  {ID_P.style.pageBreakBefore="";}
</SCRIPT>
<BUTTON onClick="KeinSeitenbruch()">Kein Seitenbruch </BUTTON>
```

.style.pixelBottom	Abstand unterer Objektrand zur Umgebung Oberkante identisch mit Attribut .style.bottom und style.posBottom aber .style.bottom: String .style.pixelBottom: Integer, nur Pixel .style.posBottom: Float pointing und Integer
.style.pixelHeight	Objekthöhe identisch mit .style.heigth und .style.posHeigt aber .style.heigt: String .style.pixelHeigt: Integer, nur Pixel .style.posHeigth: Float pointing und Integer
.style.pixelLeft	Abstand linker Objektrand zur Umgebung rechte Kante Abstand des linken Objektrandes des Kindes zum linken Objektrand des Elternelementes, wenn im Elternelement STYLE="position:....." kodiert wurde (Wert von position ist egal) Bsp.: IMG als Kind eines DIV (IMG liegt innerhalb eines DIV): Abstand der linken Kante des IMG zur linken Kante des DIV identisch mit Attribut .style.left und .style.posLeft aber .style.left: String .style.pixelLeft: Integer, nur Pixel .style.posLeft: Float pointing und Integer
.style.pixelRight	Abstand rechter Objektrand zur Umgebung linke Kante identisch mit Attribut .style.rigth und .style.posRight aber .style.right: String .style.pixelRight: Integer, nur Pixel .style.posRight: Float pointing und Integer
.style.pixelTop	Abstand oberer Objektrand zur Umgebung Unterkante Abstand des oberen Objektrandes des Kindes zum oberen Objektrand des Elternelementes, wenn im Elternelement STYLE="position:....." kodiert wurde (Wert von position ist egal) Bsp.: IMG als Kind eines DIV (IMG liegt innerhalb eines DIV): Abstand der oberen Kante des IMG zur oberen Kante des DIV identisch mit Attribut .style.top und .styleposTop aber .style.top: String .style.pixelTop: Integer, nur Pixel .style.posTop: Float pointing und Integer
.style.pixelWidth	Objektbreite identisch mit .style.width und .style.posWidth aber .style.width: String .style.pixelWidth: Integer, nur Pixel .style.posWidth: Float pointing und Integer
.style.posBottom	Abstand unterer Objektrand zur Umgebung Oberkante identisch mit Attribut .style.bottom und .style.pixelBottom aber .style.bottom: String .stylepixelBottom: Integer, nur Pixel .style.posBottom: Float pointing und Integer
.style.posHeight	Objekthöhe identisch mit .style.heigth und .style.pixelHeigt aber .style.heigt: String .style.pixelHeigt: Integer, nur Pixel



.style.posHeight: Float pointing und Integer

Beispiel:

```
<SCRIPT LANGUAGE="JScript">
  document.all.tags("IMG").item(0).style.posHeight += 10;
</SCRIPT>
```

.style.position Art der Objektpositionierung innerhalb Eltern z.B. BODY

Beispiel 1:

```
<SPAN STYLE="position:relative; top:-3px">
  Testtext
</SPAN>
```

Beispiel 2:

```
<HEAD>
<STYLE>
  .StandardKlasse { position: static }
</STYLE>
<SCRIPT>
  function AbsolutPositionieren()
  {ID_Span.style.position="absolute";}

  function RelativPositionieren()
  {ID_Span.style.position="relative";}

  function HTMLPositionieren()
  {ID_Span.style.position="static";}
</SCRIPT>
</HEAD>
<BODY>
  <SPAN ID="ID_Span" CLASS="StandardKlasse">
    Testtext
  </SPAN>
  <INPUT onclick="RelativPositionieren()" TYPE=button VALUE="relativ">
  <INPUT onclick="AbsolutPositionieren()" TYPE=button VALUE="absolut">
  <INPUT onclick="HTMLPositionieren()" TYPE=button VALUE="HTML-statisch">
</BODY>
```

.style.posLeft Abstand linker Objektrand zur Umgebung rechte Kante
Abstand des linken Objektrandes des Kindes zum linken Objektrand des Elternelementes,
wenn im Elternelement STYLE="position:....." **kodiert wurde** (Wert von position ist egal)
Bsp.: IMG als Kind eines DIV (IMG liegt innerhalb eines DIV):
Abstand der linken Kante des IMG zur linken Kante des DIV
identisch mit Attribut .style.left und .style.pixelLeft
aber .style.left: String
.style.pixelLeft: Integer, nur Pixel
.style.posLeft: Float pointing und Integer

Beispiel:

```
<SCRIPT LANGUAGE="JScript">
  document.all.tags("IMG").item(0).style.posLeft -= 10;
</SCRIPT>
```

.style.posRight Abstand rechter Objektrand zur Umgebung linke Kante
identisch mit Attribut .style.rigth und .style.pixelRight
aber .style.right: String
.style.pixelRight: Integer, nur Pixel
.style.posRight: Float pointing und Integer

.style.posTop Abstand oberer Objektrand zur Umgebung Unterkante
Abstand des oberen Objektrandes des Kindes zum oberen Objektrand des Elternelementes,
wenn im Elternelement STYLE="position:....." **kodiert wurde** (Wert von position ist egal)
Bsp.: IMG als Kind eines DIV (IMG liegt innerhalb eines DIV):
Abstand der oberen Kante des IMG zur oberen Kante des DIV
identisch mit Attribut .style.top und .style.pixelTop
aber .style.top: String
.style.pixelTop: Integer, nur Pixel
.style.posTop: Float pointing und Integer

Beispiel:

```
<SCRIPT LANGUAGE="JScript">
  document.all.tags("IMG").item(0).style.posTop -= 10;
</SCRIPT>
```

.style.posWidth Objektbreite
identisch mit .style.width und .style.pixelWidth



aber	.style.width:	String
	.style.pixelWidth:	Integer, nur Pixel
	.style.posWidth:	Float pointing und Integer

Beispiel:

```
<SCRIPT LANGUAGE="JScript">
  document.all.tags("IMG").item(0).style.posWidth += 10;
</SCRIPT>
```

.style.right	Abstand rechter Objektrand zur Umgebung linke Kante benötigt kodiertes .style.position Attribut identisch mit Attribut .style.posRigth und .style.pixelRight	
aber	.style.right:	String
	.style.pixelRight:	Integer, nur Pixel
	.style.posRight:	Float pointing und Integer

Beispiel:

```
<DIV STYLE = "position:absolute; right:50px">
  ...
</DIV>
```

.style.rubyAlign	Ausrichtung des ruby Objektes hier nicht weiter erklärt im STYLE-Attribut ruby-align kodieren
------------------	---

.style.rubyOverhang	Überhang des ruby Objektes hier nicht weiter erklärt im STYLE-Attribut ruby-overhang kodieren
---------------------	---

.style.rubyPosition	Position des ruby Objektes hier nicht weiter erklärt im STYLE-Attribut ruby-position kodieren
---------------------	---

.style.scrollbar3dLightColor	xxxxxxx	x	scrollbar3dLightColor
	yyyyyyyyy	y	scrollbarHightlightColor
	xy.....z	scrollbarFaceColor
	xy.../\...z	/\	scrollbarArrowColor (nur falls Pfeil vorhanden ist)
	xy.....z		
	xzzzzzzz	z	scrollbarShadowColor
	wwwwwww	w	scrollbarDarkShadowColor

Beispiel:

```
<HEAD>
<STYLE>
  TEXTAREA.Blue3dLight { scrollbar-3dlight-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="Blue3dLight">
    Testtext
  </TEXTAREA>
</BODY>
```

.style.scrollbarArrowColor	xxxxxxx	x	scrollbar3dLightColor
	yyyyyyyyy	y	scrollbarHightlightColor
	xy.....z	scrollbarFaceColor
	xy.../\...z	/\	scrollbarArrowColor (nur falls Pfeil vorhanden ist)
	xy.....z		
	xzzzzzzz	z	scrollbarShadowColor
	wwwwwww	w	scrollbarDarkShadowColor

Beispiel:

```
<HEAD>
<STYLE>
  TEXTAREA.BlueArrow { scrollbar-arrow-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="BlueArrow">
    Testtext
  </TEXTAREA>
</BODY>
```

.style.scrollbarBaseColor	Basisfarbe aller Scrollbalken-Elemente automatische Farbstafflung für sämtliche Teilkomponentender Scrollbalken-Elemente
---------------------------	--

Beispiel 1:

```
<HEAD>
<STYLE>
```



```

TEXTAREA.BlueScrollbar { scrollbar-base-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="BlueScrollbar">
    Testtext
  </TEXTAREA>
</BODY>

```

Beispiel 2:

```

<HEAD>
<STYLE>
  BODY { scrollbar-base-color:darkolivegreen; background-color:tan }
  H1 { color:bisque }
  P { color:darkslategray }
</STYLE>
</HEAD>

```

.style.scrollbarDarkShadowColor

```

xxxxxxxxxx x scrollbar3dLightColor
yyyyyyyyyyw y scrollbarHightlightColor
xy.....zw ..... scrollbarFaceColor
xy...../.....zw /\ scrollbarArrowColor (nur falls Pfeil vorhanden ist)
xy.....zw
xzzzzzzzzw z scrollbarShadowColor
wwwwwwwww w scrollbarDarkShadowColor

```

Beispiel:

```

<HEAD>
<STYLE>
TEXTAREA.BlueShadow { scrollbar-darkshadow-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="BlueShadow">
    Testtext
  </TEXTAREA>
</BODY>

```

.style.scrollbarFaceColor

```

xxxxxxxxxx x scrollbar3dLightColor
yyyyyyyyyyw y scrollbarHightlightColor
xy.....zw ..... scrollbarFaceColor
xy...../.....zw /\ scrollbarArrowColor (nur falls Pfeil vorhanden ist)
xy.....zw
xzzzzzzzzw z scrollbarShadowColor
wwwwwwwww w scrollbarDarkShadowColor

```

Beispiel:

```

<HEAD>
<STYLE>
TEXTAREA.BlueFace { scrollbar-face-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="BlueFace">
    Testtext
  </TEXTAREA>
</BODY>

```

.style.scrollbarHighlightColor

```

xxxxxxxxxx x scrollbar3dLightColor
yyyyyyyyyyw y scrollbarHightlightColor
xy.....zw ..... scrollbarFaceColor
xy...../.....zw /\ scrollbarArrowColor (nur falls Pfeil vorhanden ist)
xy.....zw
xzzzzzzzzw z scrollbarShadowColor
wwwwwwwww w scrollbarDarkShadowColor

```

Beispiel:

```

<HEAD>
<STYLE>
TEXTAREA.BlueHighlight { scrollbar-highlight-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="BlueHighlight">
    Testtext
  </TEXTAREA>

```



</BODY>

.style.scrollbarShadowColor xxxxxxxxxxxx x scrollbar3dLightColor
 .style.scrollbarHighlightColor yyyyyyyyyyw y scrollbarHightlightColor
 .style.scrollbarFaceColor xy.....zw scrollbarFaceColor
 .style.scrollbarArrowColor xy.....\.....zw /\ scrollbarArrowColor (nur falls Pfeil vorhanden ist)
 .style.scrollbarShadowColor xy.....zw scrollbarShadowColor
 .style.scrollbarDarkShadowColor xxxxxxxxxxxzw z scrollbarShadowColor
 .style.scrollbarDarkShadowColor wwwwwwwww w scrollbarDarkShadowColor

Beispiel:

```
<HEAD>
<STYLE>
  TEXTAREA.BlueShadow { scrollbar-shadow-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="BlueShadow">
    Testtext
  </TEXTAREA>
</BODY>
```

.style.scrollbarTrackColor Farbe des Trackelementes von Scrollbalken (Track = ziehen)

Beispiel:

```
<HEAD>
<STYLE>
  TEXTAREA.BlueTrack { scrollbar-track-color:blue }
</STYLE>
</HEAD>
<BODY>
  <TEXTAREA CLASS="BlueTrack">
    Testtext
  </TEXTAREA>
</BODY>
```

.style.styleFloat Textumfluss links, rechts
 unter IE 5.x: DIV und SPAN müssen .style.width kodiert haben
 siehe auch Eigenschaften .style.whiteSpace
 .style.clear
 .style.float

Beispiel:

```
<IMG ID="ID_IMG" SRC="test.jpg">
<BUTTON        onmouseover="ID_IMG.style.styleFloat='right'"
                 onmouseout="ID_IMG.style.styleFloat='left'"
>
  Flip-Flop-Bild
</BUTTON>
```

.style.tableLayout Tabellen-Layout (auto oder fixed)
 siehe auch Eigenschaft .style.minHeight
 Tabelle mit fixiertem Layout erzeugen per Eigenschaft .style.tableLayout mit Wert "fixed"
 wird schneller dargestellt als bei Auto-Layout
 Hinweis: Standard bei Tabellen Eigenschaft .style.tableLayout mit Wert "auto"
 also Auto-Layout, also kein fixiertes Tabellenlayout

Beispiel:

```
<TABLE STYLE="table-layout:fixed" WIDTH=600>
  <COL WIDTH=100>
    <COL WIDTH=300>
      <COL WIDTH=200>
        <TR HEIGHT=20>
          <TD>...</TD>
          <TD>...</TD>
          <TD>...</TD>
        </TR>
      ....
    </TABLE>
```

.style.textAlign Textausrichtung nur für Block-Elemente wie SPAN oder DIV
 siehe auch .style.textAlignLast und .style.textJustify

Beispiel 1:

```
<HEAD>
<STYLE>
  P { text-align:center }
```



```

        .align1 { text-align:right }
        .align2 { text-align:justify }
</STYLE>
</HEAD>
<BODY>
    <P      onclick="this.className='align1'"
           ondblclick="this.className='align2'"
    >
        ...
    </P>
</BODY>

```

Beispiel 2:

```

<P      STYLE="font-size:14"
        onmouseover="this.style.textAlign='center'"
>
    ...
</P>

```

.style.textAlignLast Textausrichtung der letzten Zeile
 nur für Block-Elemente wie SPAN oder DIV
 siehe auch **.style.textAlign** und **.style.textJustify**

Beispiel:

```

<P      STYLE="font-size:14"
        onmouseover="this.style.textAlignLast='center'"
>
    ...
</P>

```

.style.textAutospace Zusatz-Textabstand bei asiatischen Zeichen

.style.textDecoration dekoratives Layout eines Textes für nichtleere Textobjekte (unterstreichen, überstreichen, durchstreichen)
 z.B. `` ist ein leeres
 bei Block-Element: Layout wird an Kinder vererbt
 siehe auch Eigenschaft **.style.textDecorationBlink**

Beispiel 1:

```

<DIV STYLE="text-decoration:line-through">
    .....
</DIV>

```

Beispiel 2:

```

<SPAN  STYLE="font-size:14"
        onmouseover="this.style.textDecoration='underline'"
>
    .....
</SPAN>

```

.style.textDecorationBlink Wert der Eigenschaft **.style.textDecoration** auf "blink" prüfen
 Text auf blinkend prüfen
 true **.textDecoration** steht auf "blink"
 false **.textDecoration** steht auf "blink"

.style.textDecorationLineThrough Textdekoration "line-through" (durchstreichen)

Beispiel:

```

<P onclick="this.style.textDecorationLineThrough=true;">
    Testtext
</P>

```

.style.textDecorationNone Textdekoration "none" (deaktivieren aller aktiven Textdekorationen)
 true none aktiv, es wird deaktiviert
 false none nicht aktiv, kein Deaktivieren

.style.textDecorationOverline Textdekoration "overline" (überstreichen)

Beispiel:

```

<P      onmouseover="this.style.textDecorationOverline=true;">
    Testtext
</P>

```

.style.textDecorationUnderline Textdekoration "underline" (unterstreichen)
 siehe auch Eigenschaft **.style.textUnderlinePosition**

Beispiel:

```

<P onclick="this.style.textDecorationUnderline=true;">

```



Testtext
</P>

.style.textIndent Position-Ident in der der ERSTEN Zeile eines Textes
Ident nicht möglich in der Mitte eines umgebrochenen Objekts
(Umbrechung z.B. per BR)

Beispiel 1:

```
<HEAD>
<STYLE>
    DIV { text-indent:2cm }
    .click1 { text-indent:50% }
    .click2 { text-indent: }
</STYLE>
</HEAD>
<BODY>
<DIV onclick="this.className='click1'"
ondblclick="this.className='click2'"
>
    ...
</DIV>
```

Beispiel 2:

```
<DIV onmouseover="this.style.textIndent=2cm">
    .....
</DIV>
```

.style.textJustify Textausrichtung Blocksatz nur für Blockelemente wie SPAN oder DIV
Attribut .style.textAlign muss Wert "justify" haben

Beispiel:

```
<DIV STYLE="text-align='justify'; text-justify='auto'">
    .....
</DIV>
```

.textKashidaSpace Typographischer Effekt "Kashida" für Arabisch
siehe auch Eigenschaft .style.textJustify
nur für Blockelemente wie DIV und SPAN

.style.textOverflow Rahmen um Text
ab IE 6.x
Eigenschaft ermöglicht einen Rahmen um einen Text, wobei der Text im Rahmen
überlaufen kann
Überlauf und Rahmen können folgende Formen haben:
Abschneiden an Rahmengrenze
Abschneiden an Rahmengrenze mit automatischem Anfügen von ".."
als Andeutung einer Fortsetzung
kein Abschneiden, also Rahmen in voller Textbreite
Achtung: Damit der Text nicht umgebrochen wird im Rahmen, muss die Eigenschaft
.style.whiteSpace auf "nowrap" gesetzt sein
bzw. der Text in das NOBR-Tag eingeschlossen sein.
Desweiteren sollte die Eigenschaft .style.overflow auf "hidden" gesetzt sein,
damit keine Scrollbalken erscheinen (im Gegensatz zu "auto" oder
"scroll")

Beispiel 1 Rahmen mit Abschneiden an Rahmengrenze

```
STYLE ="text-overflow : clip; overflow : hidden"
```

Beispiel 2 Rahmen mit Abschneiden aber automatisch ".." als Andeutung der Fortsetzung

```
STYLE ="text-overflow : ellipsis; overflow : hidden"
```

Beispiel 3 Rahmen in kompletter Breite des Textes

```
STYLE ="text-overflow : ellipsis; overflow : visible"
```

.style.textTransform Textkonvertierung nach Gross oder Klein oder erstes Zeichen im Wort stets groß
siehe auch Objekt String

Beispiel 1:

```
<HEAD>
<STYLE>
    .transform1 { text-transform:uppercase }
    .transform2 { text-transform:lowercase }
    .transform3 { text-transform:none }
</STYLE>
```



```

</HEAD>
<BODY>
  <DIV STYLE="font-size:14"
    onmouseover="this.className='transform1'"
    onclick="this.className='transform2'"
    ondblclick="this.className='transform3'"
  >
    .....
  </DIV>
</BODY>

```

Beispiel 2:

```

<DIV STYLE="font-size:14"
  onmouseover="this.style.textTransform='uppercase'"
  onmouseout="this.style.textTransform='lowercase'"
  onclick="this.style.textTransform='none'"
>
  .....
</DIV>

```

.style.textUnderlinePosition Position der Textdekoration "underline" (unterstreichen)
siehe auch Eigenschaft .style.textDecorationUnderline

Beispiel 1:

```

<HTML STYLE="text-underline-position:above">
<BODY>
<SPAN STYLE="text-decoration:underline">
  Testtext
</SPAN>
</BODY>
</HTML>

```

.style.top Abstand oberer Objektrand zur Umgebung Unterkante
Abstand des oberen Objektrandes des Kindes zum oberen Objektrand des Elternelementes,
wenn im Elternelement STYLE="position:....." **kodiert wurde** (Wert von position ist egal)
Bsp.: IMG als Kind eines DIV (IMG liegt innerhalb eines DIV):
Abstand der oberen Kante des IMG zur oberen Kante des DIV
benötigt kodierte Eigenschaft .style.position
identisch mit Attribut .style.posTop und style.pixelTop
aber .style.top: String
.style.pixelTop: Integer, nur Pixel
.style.posTop: Float pointing und Integer

Beispiel 1:

```

<DIV STYLE="position:absolute;top:100px">
  ...
</DIV>

```

Beispiel 2:

```

<IMG SRC="test.jpg"
  STYLE="position:absolute; top:80px;" onmouseover="this.style.top='100px'"
  onmouseout="this.style.top='80px'"
>

```

.style.unicodeBidi Bidirektionale Dateneingabe von Unicode in das Element (Zeichenstrom, der vom Elternobjekt
in das Element fließt, kann seine eigene Flussrichtung besitzen).

Typische Anwendung ist z.B. der Mix aus Zeichen, die mal von links nach rechts und mal von
rechts nach links gerendert werden sollen. Damit ist für den User eine variable Leserichtung
möglich, falls der User beide Richtungen benutzen will.

Es ist möglich, den Datenstrom von Zeichen trotz aktiver bidirektionaler Dateneingabe
genau nach Vorgabe laut Eigenschaft .style.direction fließen zu lassen.
Bsp.: Wenn .style.direction auf "ltr" gesetzt ist, so wird diese Richtung konsequent eingehalten,
auch wenn der Datenstrom von rechts nach links fließen will.

Die Eigenschaft .style.direction sollte auf "inherit" gesetzt sein, wenn bidirektionaler Datenfluss
immer erfolgreich sein soll und die Flussrichtung des Datenstromes unbekannt ist.
Damit wird die Richtung des Elternobjektes übernommen.

.style.verticalAlign Ausrichtung vertikal für Objekte
mit VALIGN-Attribut
und bei Textobjekten ohne VALIGN-Attribut

Beispiel:

```

<TABLE BORDER width=100>
<TR>
  <TD onmouseover="this.style.verticalAlign='bottom'"

```



```

        onmouseout="this.style.verticalAlign=""
    >
        Testtext
    </TD>
</TR>
</TABLE>

```

.style.visibility Objekt-Sichtbarkeit ohne Reservierung des Platzs im Layout der Umgebung
 ab IE 5.x Sichtbarkeit eines Kindobjektes, auch wenn Eltern unsichtbar sind
 siehe auch Eigenschaften `.style.overflow`
 `.style.overflowX`
 `.style.overflowY`
 `.style.clip`
 `.style.display`

Hinweis: `display:none`
 Objekt behält den Platz im Layout der Umgebung,
 obwohl Objekt unsichtbar ist

Beispiel 1:

```

<HEAD>
<STYLE>
    .vis1 { visibility:visible }
    .vis2 { visibility:hidden }
</STYLE>
</HEAD>
<BODY>
    <IMG ID="ID_IMG" SRC="test.jpg">
    <P   onmouseover="ID_IMG.className='vis2'"
        onmouseout="ID_IMG.className='vis1'"
    >
        Testtext
    </P>
</BODY>

```

Beispiel 2:

```

<SCRIPT>
    function Verstecken()
    {ID_IMG.style.visibility="hidden"; }

    function Anzeigen()
    {ID_IMG.style.visibility="visible"; }
</SCRIPT>
<IMG   ID="ID_IMG"
    SRC="test.jpeg">
<SPAN  onmouseover="Verstecken()"
    onmouseout="Anzeigen()"
>
    Testtext
</SPAN>

```

.style.whiteSpace automatischer Zeilenumbruch bei Block-Elementen wie DIV und SPAN
 Umbrechen erfolgt an den Stellen vo Blanks, Tabs stehen, aber nicht an der Stelle
 eines geschützten Blanks (` `)
 siehe auch Eigenschaften `.style.pageBreakAfter`
 `.style.pageBreakBefore`
 `.style.wordBreak`
 `.style.wordWrap`

Beispiel:

```

<HEAD>
<STYLE>
    .clsOneLiner {white-space: nowrap}
    .clsAutoBreak {white-space: normal}
</STYLE>
</HEAD>
<BODY>
<P   onmouseover="this.className='clsOneLiner';"
    onmouseout="this.className='clsAutoBreak';"
>
    Testtext
</P>
</BODY>

```

.style.width Objektbreite
 identisch mit Attribut `.style.posWidth` und `.style.pixelWidth`
 aber `.style.width:` String
 `.style.pixelWidth:` Integer, nur Pixel



.style.posWidth: Float pointing und Integer

Beispiel 1:

```
<DIV STYLE="position:absolute;top:10px;left:10px;width=1in">
.....
</DIV>
```

Beispiel 2:

```
<IMG SRC="test.jpg"
onclick="this.style.width='1cm'"
ondblclick="this.style.width=""
>
```

.style.wordBreak

Zeilenumbruch in Worten
siehe auch Eigenschaften

.style.pageBreakAfter
.style.pageBreakBefore
.style.wordWrap
.style.whiteSpace

Beispiel:

```
<DIV STYLE="word-break=normal">
.....
</DIV>
```

.style.wordSpacing

Zusätzlicher Platz zwischen Worten
ab IE 6.x

Beispiel:

```
<HEAD>
<STYLE>
SPAN.spacing {word-spacing: 10;}
</STYLE>
<SCRIPT>
function PlatzAendern()
{
ID_Span.style.wordSpacing =
ID_Select.options[ID_Select.selectedIndex].text;
}
</SCRIPT>
<SELECT ID = "ID_Select" onchange = "PlatzAendern()">
<OPTION>10
<OPTION>15
<OPTION>20
</SELECT>
<SPAN ID = "ID_Span" CLASS = "spacing">
Testtext
</SPAN>
```

.style.wordWrap

Wortumbruch bei Überschreitung der Objektgrenzen
für Block-Elemente wie DIV oder SPAN
Elemente müssen kodiert haben

.style.height und/oder .style.width
.style.position mit "absolute"

siehe auch Eigenschaften

.style.pageBreakAfter
.style.pageBreakBefore
.style.wordBreak
.style.whiteSpace

Beispiel:

```
<P STYLE="word-wrap:break-word;width:100%;left:0">
Testtext
</P>
```

.style.writingMode

Flussrichtung bei Objektanzeige
wird nicht vererbt
ersetzt Eigenschaft .style.layoutFlow, da diese deprecated ist
siehe auch Eigenschaft .style.verticalAlign

Beispiel:

```
<HEAD>
<STYLE>
.clsHoriz { writing-mode:lr-tb }
</STYLE>
</HEAD>
<BODY>
<DIV STYLE="writing-mode:tb-rl">
vertikal
<SPAN CLASS="clsHoriz">
horizontal
```



```

        </SPAN>
        vertikal
        <SPAN STYLE="writing-mode:lr-tb">
            horizontal
        </SPAN>
    </DIV>
</BODY>

```

.style.zIndex Reihenfolge von überlappten Objekten
 nur für Elemente mit kodierter Eigenschaft .style.position mit dem Wert "relative" oder "absolute"
 nicht für Elemente in oder mit Fenster
 z.B. Control-Objekte mit eigenem Fenster wie Dialogbox
 im Fenster selektieres Objekt
 ab IE 5.5: iframe Objekt unterstützt z-index, da fensterlos
 siehe auch layer Objekt des Netscape unter 6.x

Beispiel 1:

```

<IMG SRC="test.jpg" STYLE="position:absolute; top:100; left:100; z-index:4">
<DIV STYLE="position:absolute; top:100; left:100; color:red; background-color:beige; font-weight:bold; z-index:1">
    .....
</DIV>

```

Beispiel 2:

```

<IMG ID="ID_Img1" SRC="test1.jpeg"
STYLE="position:absolute;top:10px;left:10px;"
onclick="ID_Img1.style.zIndex=1; ID_Img2.style.zIndex=2"
>
<IMG ID="ID_Img2" SRC="test2.jpg"
STYLE="position:absolute;top:1px;left:1px;"
onclick="ID_Img1.style.zIndex=2; ID_Img2.style.zIndex=1"
>

```

.style.zoom Objekt zoomen mit Layoutkorrektur der Objektumgebung
 Hinweis: Ein Fenster kann nur nach rechts und unten erweitert werden, wenn das gezoomte Objekt die Fenstergrenze überschreitet (falls das Fenster überhaupt erweiterbar ist). Das Zoomen wird also als Erweiterung gegenüber der linke oberen Ecke des Objektes stattfinden.
 Als Zoombasis dient immer die Objektgröße vor dem ersten Zoomen und nicht die aus dem Zoomvorgang resultierende (kumulative) Objektgröße.

Beispiel 1:

```

<HEAD>
<STYLE>
.clsTeenyWeeny { zoom: 0.10 }
</STYLE>
</HEAD>

```

Beispiel 2:

```

<P onmouseover="this.style.zoom='200%'"
onmouseout="this.style.zoom='normal'"
>
    Testtext
</P>

```

Beispiel 3:

```

<HTML>
<HEAD>
<SCRIPT>
var ZoomFaktorStartWert = 10; // > 0, ganzzahlig, in Prozent
var ZoomSchrittWeite = 1; // > 0, ganzzahlig
var DrehSchrittWeite = 1; // 1 oder 2 oder 3 da Faktor von 120 Grad
var ZoomFaktor = ZoomFaktorStartWert; // > 0, ganzzahlig, in Prozent

function VergroessernVerkleinern()
{
    if (event.wheelDelta >= (DrehSchrittWeite * 120))
    { ZoomFaktor += ZoomSchrittWeite; }
    else
    {
        if (event.wheelDelta <= (-1 * (DrehSchrittWeite * 120)))
        { ZoomFaktor -= ZoomSchrittWeite; }
    }

    if (ZoomFaktor <= 0)
    { ZooFaktor = ZoomFaktorStartWert; }

    ID_Image.style.zoom = ZoomFaktor + '%';

```



```

    }
</SCRIPT>
</HEAD>
<BODY>
    <IMG ID="ID_Image" SRC="test.jpg" onmousewheel="VergroessernVerkleinern()">
</BODY>
</HTML>

```

Beispiel 4:

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JScript">
<!--
    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'
        );

    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>

```

7.3. Style-Methoden

url() Diese Methode ist nicht direkt als Methode des Style-Objektes implementiert und darf daher **nicht** mit Punktnotation kodiert werden (vermutlich parserintern)

Beispiele für Bild:

```

<STYLE>
    .style1 {background:beige url(sphere.jpg) no-repeat top center}

```



```

        .style2{background:ivory url(sphere.jpeg) no-repeat bottom right}
</STYLE>

<SPAN onclick="this.style.background='beige url(sphere.jpeg) no-repeat top center'"></SPAN>

<STYLE>
        .setUrl { background-image: url(sphere.jpg) }
        .loseUrl { background-image: url(none) }
</STYLE>
<SPAN STYLE="font-size:14"
        onmouseover="this.className='setUrl'"
        onmouseout="this.className='loseUrl'"
>
</SPAN>

```

Beispiele für Behavior:

```

url(#objID)           mit objID als ID des OBJECT-Tags
url(#default#behaviorName)  eines Standard-Behaviors des IE

STYLE="behavior:url(a1.htc) url(a2.htc)"

<STYLE>
        .Klasse{ behavior:url(#myObject) }
</STYLE>

<STYLE>
        DIV { behavior:url(fly.htc) url (zoom.htc) url (fade.htc)}
</STYLE>

```

Beispiel für Cursorformen aus **Datei** und **nicht** für im Browser implementierte Standard-Cursorformen:
url('mycursor.cur')

Hinweise: alert (zeiger_auf_objekt.style.cursor); liefert beim IE den String 'url(cursor_form)'
mit cursor_form für den aktuellen **und** gesetzten Cursor.

style.cursor ist standardgemäß mit einer Leerkette belegt, solange **kein** Cursor gesetzt wird
kann nicht mit url(cursor_form) belegt werden, wenn es sich um eine **standardgemäß im
Browser implementierte** Cursorform handelt wie z.B. 'hand' oder 'normal'.
Grund: Diese Cursorformen sind keine Dateien, obwohl alert den String
'url(cursor_form)' anzeigt.

