

## currentStyle Objekt des Internet Explorer

referenziert **alle** zur Laufzeit **real im Element** vorhandenen **und zum Zeitpunkt aktuellen** Styles (CSS)

CSS kann für ein Element im Dokument definiert worden sein als  
 im Dokument global z.B. per userdefinierte Style-Regel im HEAD  
 im Element als Inline-Styles per STYLE-Attribut oder style Objekt  
 Pseudoklassen

Script ab IE 5.x

Style-Arten und deren Verwendung bei Objekten: siehe Objekt style

Style-Eigenschaften sind **nur lesbar**

Hinweis: Objekt style repräsentiert **nur diejenigen** zur Laufzeit **real im Element** vorhandenen **und zum Zeitpunkt aktuellen** Styles (CSS), die als Inline-Style per STYLE-Attribut oder style Objekt erzeugt wurden, aber **keinen globalen** CSS und keine Pseudoklassen.

Styles können gelesen **und** geschrieben werden

Werte von CSS-Angaben lesen: es kann nur der Wert geliefert werden, der zugewiesen wurde bzw. Standard ist

Bsp: Zuweisung von "green"  
 lesen von "green" und **nicht** #00FF00  
 erst nach dem kompletten Laden des Dokumentes  
**unmittelbar nach** dem Setzen einer CSS-Angabe nicht möglich  
 (Browser benötigt Zeit zur Verabreichung)

Empfehlung: Funktionsaufruf je für Setzen und Lesen verschafft dem Browser Zeit

### Syntax:

[ var Wert = ] object.currentStyle..style\_eigenschaft

object Zeiger auf das Objekt mit CSS

nur lesen

Beispiel 1:

```
<HEAD>
<SCRIPT>
function BraunZuWeiss(ZeigerAufObjekt)
{
    if (ZeigerAufObjekt.currentStyle.backgroundColor == 'brown')
    { ZeigerAufObjekt.style.backgroundColor = 'white'; }
    else
    { ..... }
}
</SCRIPT>
</HEAD>
<BODY>
    <P STYLE="background-color: 'brown'" onclick=" BraunZuWeiss(this)">
</BODY>
```

Beispiel 2

```
<HEAD>
<STYLE>
    P { globaler_style: myvalue }
</STYLE>
<HEAD>
<BODY>
    <P ID="ID_P">
    <SCRIPT>
        alert(ID_P.currentStyle.globaler_style);
    </SCRIPT>
</BODY>
```

Beispiel 3

```
<BODY ID="ID_Body">
<TABLE BORDER=
    <TR>
        <TD WIDTH=1100 ID="ID_TD">
            text
        </TD>
    </TR>
</TABLE>
<SCRIPT>
    alert(
        "TD hat in currentStyle.width den Wert " + ID_TD.currentStyle.width
        + ".\nDie Breites des Fensters ist " + ID_Body.clientWidth + "px."
        + "\nDie Breite der Bildschirmauflösung ist " + screen.width + "px."
    );
```



```

</SCRIPT>
</BODY>

```

Beispiel 4:

```

<SCRIPT>
function AusschnittSetzen()
{
    ID_Image.style.clip= "rect(0,100, "
                        + ID_select.options(ID_select.selectedIndex).value
                        + ",100)";

    if (ID_Image.currentStyle.clipBottom == "60px")
    { alert("60 Pixel"); }
}
</SCRIPT>
<IMG ID="ID_Image" SRC="test.jpg">
<SELECT ID="ID_Select" onchange="AusschnittSetzen()">
  <OPTION VALUE=100>100 Pixel </OPTION>
  <OPTION VALUE=40>40 Pixel </OPTION>
  <OPTION VALUE=50>50 Pixel </OPTION>
  <OPTION VALUE=60>60 Pixel </OPTION>
</SELECT>

```

**Eigenschaften:**

alle Style-Eigenschaften des style Objektes (auch die Pseudoklassen) , die aber **nur** lesbar sind

.blockDirection	Umfluss um ein Objekt "ltr" Umfluss von links nach rechts "rtl" Umfluss von rechts nach links														
.clipBottom	untere Koordinate des Ausschnittes (Clipping Region) per currentStyle Objekt prüfen ob ein Ausschnitt vorliegt														
.clipLeft	linke Koordinate des Ausschnittes (Clipping Region) per currentStyle Objekt prüfen ob ein Ausschnitt vorliegt														
.clipRight	rechte Koordinate des Ausschnittes (Clipping Region) per currentStyle Objekt prüfen ob ein Ausschnitt vorliegt														
.clipTop	obere Koordinate des Ausschnittes (Clipping Region) per currentStyle Objekt prüfen ob ein Ausschnitt vorliegt														
.hasLayout	Objekt mit Style-Layout ab IE 5.5 Style-Layout wird erzeugt durch Kodierung einer Style-Eigenschaft (siehe style Objekt) oder durch Setzen der Eigenschaft .contentEditable auf true Folgende Objekte haben immer ein Style-Layout: BODY, IMG, INPUT, TABLE, TD Objekte mit <table border="0" style="margin-left: 20px;"> <tr><td>.style.display</td><td>auf inline-block</td></tr> <tr><td>.style.height</td><td></td></tr> <tr><td>float</td><td>auf left oder right</td></tr> <tr><td>.style.position</td><td>auf absolute</td></tr> <tr><td>.style.width</td><td></td></tr> <tr><td>.style.writingMode</td><td>auf tb-rl</td></tr> <tr><td>.style.zoom</td><td></td></tr> </table>	.style.display	auf inline-block	.style.height		float	auf left oder right	.style.position	auf absolute	.style.width		.style.writingMode	auf tb-rl	.style.zoom	
.style.display	auf inline-block														
.style.height															
float	auf left oder right														
.style.position	auf absolute														
.style.width															
.style.writingMode	auf tb-rl														
.style.zoom															
	false Default. Objekt hat kein Layout														
	true Objekt hat Layout														
.onOffBehavior	deprecated ab IE 5.x Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound														

**Methoden:**

.getAttribute()	Wert eines per HTML erzeugten Attributes liefern DOM nicht geändert		
.getExpression()	Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern Style-Eigenschaft ist per Methoden <table border="0" style="margin-left: 20px;"> <tr><td>expression()</td><td>oder setExpression()</td></tr> </table> zu definieren DOM wird nicht verändert (nur Werteveränderung), aber das Dokument-Layout (nach dem eventuellen expliziten Dokument-Refresh)	expression()	oder setExpression()
expression()	oder setExpression()		
.setAttribute()	Wert von vorhandenem Attribut setzen wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt DOM wird nur bei Erzeugung geändert		
.setExpression()	Wert definieren, der als Ausdruck für die Methode .getExpression() zur Berechnung einer Style-Eigenschaft als Objektreferenz der Form <table border="0" style="margin-left: 20px;"> <tr><td>objekt.style.eigenschaft.</td></tr> </table> dient Ausdruck nur als Script kodierbar DOM wird nicht geändert	objekt.style.eigenschaft.	
objekt.style.eigenschaft.			



