

## select Objekt des Internet Explorer

select Objekt des Internet Explorer .....1  
 select.option Objekt des Internet Explorer .....7

Drop-Down-Liste oder Liste in einer Box (List-Box) als Objekt:

Die Zeiger aller Optionen werden in der Collection document.select.options gesammelt.

Ab IE 5.5 werden option Objekte zusätzlich auch über die Collection **document.all** referenzierbar.

Achtung: .z-index Attribut wird **nicht** unterstützt.

Beispiel 1:

```
<SELECT NAME="Katzen" SIZE="1">
  <OPTION VALUE="1">Mischling
  <OPTION VALUE="2">Siamese
  <OPTION VALUE="3" SELECTED>Kratzkel
</SELECT>
```

Beispiel 2:

```
<SELECT NAME="Autos" SIZE="3" MULTIPLE>
  <OPTION VALUE="1" SELECTED>BMW
  <OPTION VALUE="2">Porsche
  <OPTION VALUE="3" SELECTED>Mercedes
</SELECT>
```

Beispiel 3:

```
<SCRIPT LANGUAGE="JScript">
  var OptionObjekt = document.createElement("OPTION");
  OptionObjekt.text="Ferrari";
  OptionObjekt.value="4";

  // Annahme: selection Objekt sei vorhanden
  SelektionObjekt.add(OptionObjekt);
</SCRIPT>
```

Beispiel 4: Leere Textarea per Textwert einer selektierten Option aus Drop-Down-Liste füllen:

```
<SCRIPT LANGUAGE="JScript">
  function TextareaErweitern()
  {
    // Index der selektierte Option holen
    var Index = ID_Select.selectedIndex;

    // Text-Wert der selektieren Option holen
    var TextWert = ID_Select.options[Index].text;

    // Textwert an Textarea anhängen
    ID_Textarea.value+= TextWert + "\n";
  }
</SCRIPT>
<SELECT ID="ID_Select" SIZE="1" onchange="TextareaErweitern()">
  <OPTION VALUE="1">BMW
  <OPTION VALUE="2">PORSCHE
  <OPTION VALUE="3" SELECTED>MERCEDES
</SELECT>
<TEXTAREA ID="ID_Textarea"></TEXTAREA>
```

Beispiel 5:

```
<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>

```

Beispiel 6:

```

<SELECT DATASRC="#Anker" DATAFLD="KartenTyp">
  <OPTION>Visa
  <OPTION>Mastercard
  <OPTION>American Express
  <OPTION>Diner's Club
  <OPTION>Sparkasse
</SELECT>

```

Beispiel 7:

```

<SELECT ID="ID_select" MULTIPLE>
  <OPTION>Auswahl 1</OPTION>
  <OPTION> Auswahl 2</OPTION>
  <OPTION> Auswahl 3</OPTION>
</SELECT>
<BUTTON onclick="ID_select.multiple=false">keine multiple Auswahl</BUTTON>
<BUTTON onclick="ID_select.multiple=true">multiple Auswahl</BUTTON>

```

Beispiel 8:

```

<SCRIPT>
  var ZeigerAufNeueOption = document.createElement("OPTION");
  ID_Select.options.add(ZeigerAufNeueOption);
  ZeigerAufNeueOption.innerText = "Option 2";
  ZeigerAufNeueOption.Value = "2";
</SCRIPT>
<SELECT ID="ID_Select" >
  <OPTION VALUE="1">Option 1</OPTION>
</SELECT>

```

Beispiel 9:

```

<SCRIPT>
function Erzeugen()
{
  ID_Span.innerHTML="";
  var FeldDerZeigerAufOption = ID_Select.options[ID_Select.selectedIndex];

  if(FeldDerZeigerAufOption.text.length>0)
  {
    var ZeigerAufElement= document.createElement(FeldDerZeigerAufOption.text);
    eval(
      " ZeigerAufElement."
      + FeldDerZeigerAufOption.value
      + "="
      + ID_Input.value
      + ""
    );

    if(FeldDerZeigerAufOption.text=="A")
    { ZeigerAufElement.href="javascript:alert('A link.');" }

    ID_Span.appendChild(ZeigerAufElement);
  }
}
</SCRIPT>
<SELECT ID="ID_Select" onchange="Erzeugen()">
  <OPTION VALUE="innerText">A
  <OPTION VALUE="value">&lt;INPUT TYPE="button"&gt;
</SELECT>
<INPUT TYPE="text" ID="ID_Input" VALUE="Beispiel Text">
<SPAN ID="ID_Span" ></SPAN>

```

**Zugriff:**

```

document.all.zeiger_auf_select_objekt.eigenschaft
document.all.zeiger_auf_select_objekt.methode()

```

zeiger\_auf\_select\_objekt      z.B. laut ID-Attribut

**Eigenschaften:**

.accessKey

Tastaturzugriff auf ein Objekt per Alt + Taste  
bei Ausführung der Tastenkombination wird  
das Sprungziel wird focussiert  
die Sprungquelle defocussiert



|                    |   |
|--------------------|---|
|                    | das Focus-Ereignis ausgelöst  |
|                    | vor IE 5.x funktioniert Tastenkombination nicht mit jedem Objekt  |
| .align             | Ausrichtung   |
| ATOMICSELECTION    | Selektierbarkeit einstellen   |
| .canHaveChildren   | prüfen ob Kind möglich ist, also ob Objekt Parent sein kann   |
| .canHaveHTML       | prüfen ob Objekt HTML-Tags enthalten darf   |
| .className         | Klassenreferenz   |
| .clientHeight      | Objekthöhe in Pixel ohne Abstand zum Umgebungsobjekt<br>ohne Rahmen<br>ohne Scrollbalken  |
| .clientLeft        | Abstand in Pixel zum linken Rand des Fensters   |
| .clientTop         | Abstand in Pixel zum oberen Rand des Fensters   |
| .clientWidth       | Objekt-Breite in Pixel ohne Abstand zum Umgebungsobjekt<br>ohne Rahmen<br>ohne Scrollbalken   |
| .dataFld           | Datenquelle-Name vergeben (ID)  |
| .dataSrc           | Datenquelle als Anker festlegen   |
| .dir               | Umflussrichtung   |
| .disabled          | Interaktionsfähigkeit<br>nur wenn sichtbar so User-Interaktion möglich  |
| .firstChild        | Zeiger auf das ERSTE Kind laut childNodes-Collection eines Objektes   |
| .form              | Zeiger auf das Formular (Formular als Container)<br>ab IE 6.x für Elemente fieldSet, label, legend  |
| .hideFocus         | Focussierbarkeit  |
| .id                | Bezeichner des Objektes für Referenzierung des Objektes (Zeiger, ID)  |
| .innerHTML         | Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag<br>ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B. <B><br>dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes,<br>also nach dem Laden des Objektes<br>aber wirksam erst mit parsen des HTML-Endetag |
| .innerText         | Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag<br>ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B. <B><br>dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes,<br>also nach dem Laden des Objektes<br>aber wirksam erst mit parsen des HTML-Endetag |
| .isContentEditable | Editierbarkeit des Objekt-Content (auch wenn kein Layout hat)<br>Content = Beziehung des Objektes zum Umfeld z.B. bezüglich Layout etc.   |
| .isDisabled        | Interaktionsfähigkeit<br>nur wenn sichtbar so User-Interaktion möglich  |
| .isMultiLine       | Mehrzeiligkeit des Objektinhaltes   |
| .isTextEdit        | Erzeugbarkeit eines Textbereiches   |
| .lang              | Sprache für Anzeige von Sonderzeichen etc.  |
| .language          | Sprache für Script festlegen  |
| .lastChild         | Zeiger auf das LETZTE Kind laut childNodes collection eines Objektes  |
| .length            | Anzahl der Feldelemente also Feldlänge der Collection document.select.options<br>(siehe Eigenschaft .options)   |
| .multiple          | Multiple Auswahl bei document.select  |
| .name              | Name des Objektes (nicht ID !!!)<br>muss beim Formular für alle zu sendenden Felder kodiert sein !!<br>Element darf nicht per Methode .createElement() erzeugt worden sein  |
| .nextSibling       | Zeiger auf das NACHFOLGENDE Kind laut childNodes collection eines Objektes  |
| .nodeName          | String als Name des Kindes (Knoten, Node, Element)<br>also TAG-Bezeichner, Attribut-Name; #text für Anker   |
| .nodeType          | Knotentyp laut attributes Collection  |
| .nodeValue         | Knotenwert (Wert des Kindes, Node, Elementes)<br>nur für Text- und Attribut-Elemente<br>nicht für Element-Knoten (Knotentyp 1)  |
| .offsetHeight      | Y-Koordinate der rechten unteren Ecke des Objektes bezüglich Koordinatensystem<br>des Elternobjektes (offsetParent)   |
| .offsetLeft        | X-Koordinate der linken oberen Ecke Objektes bezüglich Koordinatensystem<br>des Elternobjektes (offsetParent)   |
| .offsetParent      | Referenz der Eltern<br>für Nutzung von offsetHeight, offsetLeft, offsetTop und offsetWidth  |
| .offsetTop         | Y-Koordinate der linken oberen Ecke des Objektes bezüglich Koordinatensystem<br>des Elternobjektes (offsetParent)   |
| .offsetWidth       | X-Koordinate der rechten unteren Ecke des Objektes bezüglich Koordinatensystem<br>des Elternobjektes (offsetParent)   |
| .onOffBehavior     | deprecated ab IE 5.x  |
| .options           | Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound  |
| .outerHTML         | Zeiger auf die Collection document.select.options des document.select Objektes<br>Referenz auf Bereich ab inklusive HTML-Start bis hinter -Ende-Tag<br>wirksam mit parsen des Ende-Tag<br>nur nach kompletten Einlesen des Dokumentes nutzbar                                       |



|                  |   |
|------------------|---|
| .outerText       | Referenz auf den gesamten Plain-Text im Objekt<br>nur nach kompletten einlesen des Dokumentes nutzbar   |
| .ownerDocument   | Referenz auf das document Objekt zu dem der Knoten gehört, also in dem der Knoten erzeugt wurde   |
| .parentElement   | Referenz auf das Elternobjekt, also nicht Elternknoten innerhalb DOM  |
| .parentNode      | Referenz auf Elternknoten innerhalb der DOM-Hierarchie  |
| .parentTextEdit  | Textbereich des Elternobjektes referenzieren  |
| .previousSibling | Referenz auf das Vorgängerkind  |
| .readyState      | aktueller Status des Objektes beim Füllen des Objektes mit Daten  |
| .recordNumber    | Datenquelle-Satznummer eines Datenfeldes  |
| .scopeName       | Namensraum laut XMLNS-Attribut  |
| .scrollHeight    | Höhe des vertikalen Scrollbereiches, also Abstand von oberen und unteren sichtbaren Rand des Umgebungsobjektes  |
| .scrollLeft      | Abstand von linken sichtbaren Randes des Umgebungsobjektes zum linken Rand des Objektes<br>nur nutzbar nach dem kompletten Laden des Dokumentes   |
| .scrollTop       | Abstand von oberen sichtbaren Randes des Umgebungsobjektes zum oberen Rand des Objektes<br>nur nutzbar nach dem kompletten Laden des Dokumentes   |
| .scrollWidth     | Breite des horizontalen Scrollbereiches, also Abstand von linken und rechten sichtbaren Rand des Umgebungsobjektes  |
| .selectedIndex   | Index der Option in der Collection document.select.options des Objektes select<br>jedes option Objekt eines select Objektes wird im Feld, das Teil des select Objektes ist, referenziert<br>es kann immer nur eine Option innerhalb einer Optionsgruppe selektiert sein   |
| .size            | Anzahl der Zeilen in einer List-Box als select Objekt   |
| .sourceIndex     | Index des Objektes in der Collection document.all   |
| STYLE            | direkt im HTML-Element kodierter Style (Inline-Style)<br>Hinweis: für Scripting ist das Style-Objekt zu nutzen<br>Achtung: Bezüglich der Style-Eigenschaften .background-color und .color kann nur genau 1 Style deklariert werden für<br><b>alle</b> Objekte document.select.option<br><b>und</b> das document.select Objektes<br>insgesamt.<br>Es wird der Style des ersten in der Kodierungsfolge gefundenen Objektes document.select.option für alle anderen <b>und</b> das Objekt document.select selbst verwendet.<br>Es wird angeraten, <b>nur</b> im document.selelet das STYLE-Attribut zu verwenden.<br>Das Objekt document.select.option kann <b>nur</b> genau folgende Style-Eigenschaften erhalten: .background-color und .color |
| .tabIndex        | Index des Elementes in der Tab-Tasten-Folge für Anspringen des Dokumentes<br>Anspringen verbunden mit Focus erhalten<br>--> Ereignisse werden ausgelöst !!<br>unter IE 5.x<br>onblur, onfocus<br>ab IE 5.x<br>onblur, onfocus, onkeydown, onkeypress, onkeyup<br><br>Anspringen default per TAB-Taste<br>für A, BODY, BUTTON, FRAME, IFRAME, IMG, INPUT, SELECT, TEXTAREA<br><br>Anspringen default nicht per TAB-Taste<br>für APPLET, DIV, FRAMESET, SPAN, TABLE, TD   |
| .tagName         | Tag-Bezeichner des Objektes   |
| .tagUrn          | Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut   |
| .type            | prüfen auf multiples select Objekt  |
| .uniqueID        | durch den Browser automatisch-generiertes ID des Objektes<br>Browser generiert zu verschiedenen Zeitpunkten auch verschiedene ID, wenn Objekt mehrmals geladen wurde<br>kann anstelle eines privat vergebenen ID als ID-Attributwert weiterverwendet werden   |
| UNSELECTABLE     | Selektierbarkeit z.B. mit Maus<br><b>wird nicht an Kinder vererbt</b>   |
| .value           | bei Wechsel: vorhergehende vorhandene Selektion wird nicht verändert<br>Wert eines Objekt-Attributes<br>Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar  |
| <b>Methoden:</b> |   |
| .add()           | ein bereits erzeugtes Element einer Collection hinzufügen<br>hinzufügen erst nach dem kompletten Laden des Dokumentes<br>Element erzeugen per Methode .createElement()  |
| .addBehavior()   | DHTML-Verhaltenseigenschaft einem Element hinzufügen<br>Empfehlung: Standard-IE-Eigenschaften nutzen, da diese mit "#default#behaviorName" komplett erfasst werden und bereits im Browser implementiert sind (keine HTC-Datei nötig).<br>ab IE 5.x bis unter IE 5.5   |
| .applyElement()  | Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern<br>DOM wird geändert  |



|                                       |  |
|---------------------------------------|--|
|                                       | Element kann selbst Kinder haben<br>Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde<br>Achtung: Wenn Element per Methode <code>.createElement()</code> erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft <code>.innerHTML</code> gelöscht !   |
| <code>.attachEvent()</code>           | Einschalten des Registrieren eines Events durch Eventhandler<br>Hinweis: Abschalten mit Methode <code>.detachEvent()</code><br>Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in <b>Zufallsfolge</b> , es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)  |
| <code>.blur()</code>                  | Element den Focus wegnehmen und Event onblur auslösen<br>Der Focus wird nicht automatisch auf irgend ein anderes Element gesetzt !<br>vor IE 5.0 TABINDEX-Attribut muss kodiert sein<br>ab IE 5.0 TABINDEX-Attribut muss nicht kodiert sein  |
| <code>.clearAttributes()</code>       | alle HTML-Attribute eines Objektes entfernen<br>außer ID, STYLE und per Script definierte Attribute<br>Script-erzeugte Attribute nicht entfernbar  |
| <code>.click()</code>                 | DOM wird geändert<br>simuliert einen Klick auf das Element und löst onclick-Event aus<br>manipuliert nicht den Focus   |
| <code>.cloneNode()</code>             | Objekt klonen und Referenz des erzeugten Klone liefern<br>DOM wird nicht geändert, da Klone nicht in DOM eingebunden wird (reines Neu-Instanzieren eines DOM-Objektes im Hauptspeicher außerhalb des DOM)  |
| <code>.componentFromPoint()</code>    | Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt<br>auch für CSS-Layout<br>onmouseover-Event hat <b>nicht die Pixelgenauigkeit</b> wie die Angaben der Methode <code>.componentFromPoint()</code><br>also wenn Event erzeugt, muss die Maus noch lange nicht genau die obengenannte Pixelpos erreicht haben<br>Overbereich der Maus ist mehr als 1 Pixel gross<br>beachte Einstellungen der Maus zur Cursorgeschwindigkeit etc.   |
| <code>.contains()</code>              | prüfen ob Element innerhalb eines Elementes liegt, also ob das innere, eingeschlossene Element Eltern (Eltern-Objekt, Container) hat und somit ein Kind-Objekt ist<br>DOM nicht geändert   |
| <code>.detachEvent()</code>           | Abschalten des Registrieren eines Events durch Eventhandler<br>wobei Registrierung mit Methode <code>.attachEvent()</code> aktiviert wurde<br>Abschalten = ordnet dem Window-Objekt das Event laut Parameter <code>event_</code> bezeichner zu, das <b>nicht</b> behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)  |
| <code>.dragDrop()</code>              | prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element   |
| <code>.fireEvent()</code>             | ein Event auslösen   |
| <code>.focus()</code>                 | Focus setzen und Focus-Event auslösen<br>nur nach dem kompletten Laden des Dokumentes<br>vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen  |
| <code>.getAdjacentText()</code>       | Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann<br>Text kann HTML-Tags enthalten, muss aber nicht<br>DOM nicht geändert  |
| <code>.getAttribute()</code>          | Wert eines per HTML erzeugten Attributes liefern<br>DOM nicht geändert   |
| <code>.getAttributeNode()</code>      | Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf <code>attribute.name</code> Eigenschaft.<br>Eigenschaft kann mit HTML-Anweisung erzeugt worden sein, muss aber nicht<br>Eigenschaft ist selbst ein Knoten in der Attribute-Objekt-Hierarchie zum Objekt<br>Wert des Attributes wird somit über die Referenz laut DOM erreichbar<br>DOM nicht geändert   |
| <code>.getBoundingClientRect()</code> | Referenz auf TextRectangle-Objekt im Element holen   |
| <code>.getClientRects()</code>        | Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster<br>Feld mit Index als Integer ab 0<br>pro Eintrag ein Rectangle  |
| <code>.getElementsByName()</code>     | Referenz auf ein Feld (Collection) aller im Objekt befindlichen Kinder-Objekte mit gemeinsamen Tagnamen liefern, inklusive aller Kinder und Unterkinder etc.<br>Hinweis: Natürlich kann auch das document-Objekt so verarbeitet werden (beachte dabei <code>document.all</code> Collection)<br>Achtung: Kinder-Objekte, die keinen Tag-Name besitzen, werden nicht erfasst !<br>Für Verwaltung per ID (analog zum ID-Attribut):<br>siehe Methode <code>getElementsById()</code><br>Für Verwaltung per NAME (analog zum NAME-Attribut):<br>siehe Methode <code>getElementsByName()</code> |
| <code>.getExpression()</code>         | DOM nicht geändert<br>Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern<br>Style-Eigenschaft ist per Methoden<br><code>expression()</code> oder <code>setExpression()</code><br>zu definieren<br>DOM wird nicht verändert (nur Werteveränderung), aber das Dokument-Layout (nach dem eventuellen expliziten Dokument-Refresh)   |
| <code>.hasChildNodes()</code>         | prüfen auf Existenz von Kinder(n) als HTML-Elemente oder Textknoten  |



- (Textelemente) in einem Objekt  
DOM nicht geändert

.insertAdjacentElement()  
Objekt in eine Objekt einfügen und Referenz liefern, wobei die Lage definiert werden kann  
wenn Element bereits eingefügt vorhanden, so wird dieses nur verschoben laut Lage des Objektes im DOM  
nur nach dem kompletten Laden des Dokumentes möglich  
DOM wird geändert
- .insertAdjacentHTML()  
HTML-Code und/oder Script-Code in ein Element einfügen, wobei die Lage definiert sein kann  
nur nach dem kompletten Laden des Dokumentes möglich  
HTML- und Script-Code müssen syntaktisch korrekt sein  
wenn nicht, so wird das Einfügen **nicht** ausgeführt  
eingefügter Code wird **nur** dann sofort geparkt und ausgeführt, wenn syntaktisch korrekt ist  
bei Script-Code: <SCRIPT DEFER .....> muss kodiert werden  
DOM wird geändert
- .insertAdjacentText()  
Plain-Text (ohne HTML und Script) in ein Element einfügen, wobei die Lage definiert werden kann  
nur nach dem kompletten Laden des Dokumentes  
DOM wird geändert
- .insertBefore()  
Objekt als Kindknoten VOR dem einem anderen Kind-Objekt einfügen und Zeiger liefern  
einzufügendes Objekt muss mit Methode createElement() erzeugt worden sein  
Achtung: NICHT anwenden für einfügen VON bzw. VOR obersten Kindknoten  
Sichtbarkeit erst wenn Ende-Tag geparkt wurde  
DOM wird geändert
- .mergeAttributes()  
alle Attribute eines Elementes in ein anderes Element kopieren und eventuell die Attribute im Ziel mischen  
Attribute sind: HTML  
Events  
Styles  
ab IE 5.01 auch ID, NAME  
Achtung: Diese Methode ist mir Vorsicht zu geniessen !!  
DOM wird geändert
- .normalize()  
Normalisierung des DOM zur Erreichung einer konsistenten Struktur  
Achtung: CDATA-Sections dürfen nicht enthalten sein, da diese immer Inkonsistenz erzeugen
- .releaseCapture()  
Maus-Überwachung ausschalten für ein Objekt  
Maus-Events sind : onmousedown, onmouseup, onmouseover, onclick, ondblclick,  
onmouseover und onmouseout.
- .remove()  
Hinweis: einschalten per Methode .setCapture()  
ein Element aus einer Collection entfernen
- .removeAttribute()  
entfernen eines per HTML erzeugten Attributes  
Achtung: Der Browser unterscheidet zwischen HTML-erzeugte oder mit dieser Methode erzeugte Attribute!  
per Methode .createAttribute() erzeugte Attribute werden nicht erfasst  
DOM wird geändert
- .removeAttributeNode()  
entfernen von Attribut, egal ob es mit oder ohne HTML-Anweisung erzeugt wurde, und Referenz auf das  
entfernte Attribut liefern  
DOM wird geändert
- .removeBehavior()  
per Methode .addBehavior() einem Element hinzugefügte Verhaltenseigenschaft entfernen (stets VOR dem  
Entfernen des Elementes mit der zugeordneten Eigenschaft aus der Dokument-Hierarchie)  
DOM wird geändert
- .removeChild()  
Kind-Objekt aus einem Objekt entfernen aus DOM und Referenz auf das entfernte Kind liefern  
Sichtbarkeit erst, wenn Ende-Tag geparkt wurde, also das Dokument neu geladen wurde  
DOM wird geändert
- .removeExpression()  
Ausdruck entfernen, der für die Berechnung des Wertes einer Style-Eigenschaft als Objektreferenz der  
Form objekt.style.eigenschaft.dient.  
Ausdruck muss mit der Methode .setExpression() gesetzt worden sein  
DOM wird nicht geändert
- .removeNode()  
Knoten entfernen aus DOM und Referenz auf den entfernten Knoten liefern  
Sichtbarkeit erst wenn Ende-Tag geparkt wurde  
DOM wird geändert
- .replaceAdjacentText()  
Plain-Text (ohne HTML und Script) eines Elementes durch anderen Text ersetzen und Referenz auf den zu  
ersetzenden Text liefern  
DOM wird nicht geändert
- .replaceChild()  
Kind-Objekt ersetzen durch ein Objekt  
ersetzende Objekt muss per Methode .createElement() erzeugt worden sein  
Sichtbarkeit erst wenn Ende-Tag geparkt wurde  
DOM wird geändert
- .replaceNode()  
Objekt durch anderes Objekt komplett ersetzen und Referenz auf das komplett ersetzte Objekt liefern  
sichtbar erst mit parsen des Endetags  
DOM wird geändert
- .scrollIntoView()  
Objekt derart scrollen, dass es im Fenster für User sichtbar wird  
Objekt muss an sich schon renderbar sein
- .setActive()  
Objekt für die Eventdurchreichung aktivieren  
aber ohne es zu fokussieren  
und ohne es scrollbar zu machen
- .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



|                     |   |
|---------------------|---|
| .setAttributeNode() | Attribut einem Knoten zuweisen und Referenz liefern<br>DOM wird geändert  |
| .setCapture()       | Maus-Überwachung einschalten für ein Objekt<br>Maus-Events sind : onmousedown, onmouseup, onmousemove, onclick, ondblclick,<br>onmouseover und onmouseout.<br>ab IE 5.5<br>Hinweis: ausschalten per Methode .releaseCapture()               |
| .setExpression()    | Wert definieren, der als Ausdruck für die Methode .getExpression() zur Berechnung einer Style-Eigenschaft als Objektreferenz der Form<br>objekt.style.eigenschaft.<br>dient<br>Ausdruck nur als Script kodierbar<br>DOM wird nicht geändert |
| .swapNode()         | Positionen von 2 Knoten im DOM tauschen (Zeigertausch)<br>nur sichtbar wenn Endetag geparkt<br>DOM wird geändert  |
| .urns()             | Referenz auf Feld aller Elemente mit gemeinsamer URN liefern  |

## **select.option Objekt des Internet Explorer**

Objekt einer Option aus einer Selektion per document.select Objekt

Option ist anwählbar

Die Zeiger aller Optionen werden in der Collection document.select.options gesammelt.

**Ab IE 5.5** werden option Objekte zusätzlich auch über die Collection **document.all** referenzierbar.

Zum Hinzufügen eines option Objektes müssen das select und option Objekt in einem **gemeinsamen** Fenster liegen.

Beispiel 1:

```
<SELECT NAME="Katzen" SIZE="1">
  <OPTION VALUE="1">Mischling
  <OPTION VALUE="2">Siamese
  <OPTION VALUE="3" SELECTED>Kratzkegel
</SELECT>
```

Beispiel 2:

```
<SELECT NAME="Autos" SIZE="3" MULTIPLE>
  <OPTION VALUE="1" SELECTED>BMW
  <OPTION VALUE="2">Porsche
  <OPTION VALUE="3" SELECTED>Mercedes
</SELECT>
```

Beispiel 3:

```
<SCRIPT LANGUAGE="JScript">
  var OptionObjekt = document.createElement("OPTION");
  OptionObjekt.text="Ferrari";
  OptionObjekt.value="4";

  // Annahme: selection Objekt sei vorhanden
  SelektionObjekt.add(OptionObjekt);
</SCRIPT>
```

Beispiel 4: Leere Textarea per Textwert einer selektierten Option aus Drop-Down-Liste füllen:

```
<SCRIPT LANGUAGE="JScript">
  function TextAreaErweitern()
  {
    // Index der selektierte Option holen
    var Index = ID_Selelect.selectedIndex;

    // Text-Wert der selektieren Option holen
    var TextWert = ID_Selelect.options[Index].text;

    // Textwert an Textarea anhängen
    ID_Textarea.value+= TextWert + "\n";
  }
</SCRIPT>
<SELECT ID="ID_Selelect" SIZE="1" onchange="TextAreaErweitern()">
  <OPTION VALUE="1">BMW
  <OPTION VALUE="2">PORSCHE
  <OPTION VALUE="3" SELECTED>MERCEDES
</SELECT>
<TEXTAREA ID="ID_Textarea"></TEXTAREA>
```

Beispiel 5:





```

<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>

```

Beispiel 6:

```

<SELECT DATASRC="#Anker" DATAFLD="KartenTyp">
    <OPTION>Visa
    <OPTION>Mastercard
    <OPTION>American Express
    <OPTION>Diner's Club
    <OPTION>Sparkasse
</SELECT>

```

Beispiel 7:

```

<SELECT ID="ID_select" MULTIPLE>
    <OPTION>Auswahl 1</OPTION>
    <OPTION> Auswahl 2</OPTION>
    <OPTION> Auswahl 3</OPTION>
</SELECT>
<BUTTON onclick="ID_select.multiple=false">keine multiple Auswahl</BUTTON>
<BUTTON onclick="ID_select.multiple=true">multiple Auswahl</BUTTON>

```

Beispiel 8:

```

<SCRIPT>
var ZeigerAufNeueOption = document.createElement("OPTION");
ID_Select.options.add(ZeigerAufNeueOption);
ZeigerAufNeueOption.innerText = "Option 2";
ZeigerAufNeueOption.Value = "2";
</SCRIPT>
<SELECT ID="ID_Select" >
    <OPTION VALUE="1">Option 1</OPTION>
</SELECT>

```

Beispiel 9:

```

<SCRIPT>
function Erzeugen()
{
    ID_Span.innerHTML="";
    var FeldDerZeigerAufOption = ID_Select.options[ID_Select.selectedIndex];

    if(FeldDerZeigerAufOption.text.length>0)
    {
        var ZeigerAufElement= document.createElement(FeldDerZeigerAufOption.text);
        eval(
            " ZeigerAufElement."
            + FeldDerZeigerAufOption.value
            + "="
            + ID_Input.value
            + ""
        );

        if(FeldDerZeigerAufOption.text=="A")
        { ZeigerAufElement.href="javascript:alert('A link.')"; }
    }
    ID_Span.appendChild(ZeigerAufElement);
}
</SCRIPT>
<SELECT ID="ID_Select" onchange="Erzeugen(">
    <OPTION VALUE="innerText">A

```





```

<OPTION VALUE="value">&It;INPUT TYPE="button"&gt;
</SELECT>
<INPUT TYPE="text" ID="ID_Input" VALUE="Beispiel Text">
<SPAN ID="ID_Span" ></SPAN>

```

**Zugriff:**

```

var Zeiger = document.all.zeiger_auf_select_objekt
Zeiger.zeiger_auf_option_objekt.eigenschaft
Zeiger.zeiger_auf_option_objekt.methode()

```

zeiger\_auf\_select\_objekt z.B. laut ID-Attribut  
zeiger\_auf\_option\_objekt z.B. laut ID-Attribut

```
var ZeigerAufFeldElement = document.zeiger_auf_select_objekt.options [Index [, SubIndex]]
```

Index Integer ab 0  
oder String Name oder ID des Elementes  
(analog zu NAME und ID-Attribut)  
muss in [ ] kodiert sein

SubIndex optional  
nur kodieren wenn Index ein String ist  
Integer als Unterindex also Unterelement eines Elementes

zeiger\_auf\_select\_objekt z.B. laut ID-Attribut

**Eigenschaften:**

|                    |   |
|--------------------|---|
| .begin             | Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction<br>siehe Objekt currTimeState und Behavior .style.time2  |
| .canHaveChildren   | prüfen ob Kind möglich ist, also ob Objekt Parent sein kann   |
| .canHaveHTML       | prüfen ob Objekt HTML-Tags enthalten darf   |
| .className         | Klassenreferenz   |
| .clientHeight      | Objekthöhe in Pixel ohne Abstand zum Umgebungsobjekt<br>ohne Rahmen<br>ohne Scrollbalken  |
| .clientLeft        | Abstand in Pixel zum linken Rand des Fensters   |
| .clientTop         | Abstand in Pixel zum oberen Rand des Fensters   |
| .clientWidth       | Objekt-Breite in Pixel ohne Abstand zum Umgebungsobjekt<br>ohne Rahmen<br>ohne Scrollbalken   |
| .defaultSelected   | Default-Selektionsstatus der Option, also Objektes document.select.option des Objektes document.select<br>standardgemäß ist immer die oberste Option selektiert<br>es kann aus einer Optionengruppe immer nur genau 1 Option selektiert sein  |
| .dir               | Umflussrichtung   |
| .disabled          | Interaktionsfähigkeit<br>nur wenn sichtbar so User-Interaktion möglich  |
| .end               | Objektaktivitäten laut Eigenschaft .timeAction beenden<br>ab IE 6.x<br>alternativ: Eigenschaft .dur   |
| .firstChild        | siehe Objekt currTimeState und Behavior .style.time2  |
| .form              | Zeiger auf das ERSTE Kind laut childNodes-Collection eines Objektes<br>Zeiger auf das Formular (Formular als Container)<br>ab IE 6.x für Elemente fieldSet, label, legend   |
| .hasMedia          | Objekt ist HTML-Media-Objekt<br>siehe Objekt currTimeState und Behavior .style.time2  |
| .id                | Bezeichner des Objektes für Referenzierung des Objektes (Zeiger, ID)  |
| .index             | laufende Nummer der Option, also Objektes document.select.option, in einer List-Box als<br>document.select Objekt   |
| .innerHTML         | Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag<br>ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B. <B><br>dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes,<br>also nach dem Laden des Objektes<br>aber wirksam erst mit parsen des HTML-Endetag |
| .innerText         | Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag<br>ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B. <B><br>dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes,<br>also nach dem Laden des Objektes<br>aber wirksam erst mit parsen des HTML-Endetag |
| .isContentEditable | Editierbarkeit des Objekt-Content (auch wenn kein Layout hat)<br>Content = Beziehung des Objektes zum Umfeld z.B. bezüglich Layout etc.   |
| .isDisabled        | Interaktionsfähigkeit<br>nur wenn sichtbar so User-Interaktion möglich  |
| .isMultiLine       | Mehrzeiligkeit des Objektinhaltes   |
| .isTextEdit        | Erzeugbarkeit eines Textbereiches   |
| .label             | Label einer Option, also Objektes document.select.option des Objektes document.select   |



|                          |   |
|--------------------------|---|
| .lang                    | Sprache für Anzeige von Sonderzeichen etc.  |
| .language                | Sprache für Script festlegen  |
| .lastChild               | Zeiger auf das LETZTE Kind laut childNodes collection eines Objektes  |
| .nextSibling             | Zeiger auf das NACHFOLGENDE Kind laut childNodes collection eines Objektes  |
| .nodeName                | String als Name des Kindes (Knoten, Node, Element)<br>also TAG-Bezeichner, Attribut-Name; #text für Anker   |
| .nodeType                | Knotentyp laut attributes Collection  |
| .nodeValue               | Knotenwert (Wert des Kindes, Node, Elementes)<br>nur für Text- und Attribut-Elemente<br>nicht für Element-Knoten (Knotentyp 1)  |
| .offsetHeight            | Y-Koordinate der rechten unteren Ecke des Objektes bezüglich Koordinatensystem des Elternobjektes (offsetParent)  |
| .offsetLeft              | X-Koordinate der linken oberen Ecke Objektes bezüglich Koordinatensystem des Elternobjektes (offsetParent)  |
| .offsetParent            | Referenz der Eltern<br>für Nutzung von offsetHeight, offsetLeft, offsetTop und offsetWidth  |
| .offsetTop               | Y-Koordinate der linken oberen Ecke des Objektes bezüglich Koordinatensystem des Elternobjektes (offsetParent)  |
| .offsetWidth             | X-Koordinate der rechten unteren Ecke des Objektes bezüglich Koordinatensystem des Elternobjektes (offsetParent)  |
| .onOffBehavior           | deprecated ab IE 5.x<br>Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound  |
| .ownerDocument           | Referenz auf das document Objekt zu dem der Knoten gehört, also in dem der Knoten erzeugt wurde   |
| .parentElement           | Referenz auf das Elternobjekt, also nicht Elternknoten innerhalb DOM  |
| .parentNode              | Referenz auf Elternknoten innerhalb der DOM-Hierarchie  |
| .parentTextEdit          | Textbereich des Elternobjektes referenzieren  |
| .previousSibling         | Referenz auf das Vorgängerkind  |
| .readyState              | aktueller Status des Objektes beim Füllen des Objektes mit Daten  |
| .scopeName               | Namensraum laut XMLNS-Attribut  |
| .scrollHeight            | Höhe des vertikalen Scrollbereiches, also Abstand von oberen und unteren sichtbaren Rand des Umgebungsobjektes  |
| .scrollLeft              | Abstand von linken sichtbaren Randes des Umgebungsobjektes zum linken Rand des Objektes<br>nur nutzbar nach dem kompletten Laden des Dokumentes   |
| .scrollTop               | Abstand von oberen sichtbaren Randes des Umgebungsobjektes zum oberen Rand des Objektes<br>nur nutzbar nach dem kompletten Laden des Dokumentes   |
| .scrollWidth             | Breite des horizontalen Scrollbereiches, also Abstand von linken und rechten sichtbaren Rand des Umgebungsobjektes  |
| .selected                | Selektionsstatus der Option, also Objektes document.select.option des Objektes document.select<br>standardgemäß ist immer die oberste Option selektiert<br>es kann aus einer Optionengruppe immer nur genau 1 Option selektiert sein<br>direkt im HTML-Element kodierter Style (Inline-Style)<br>Hinweis: für Scripting ist das Style-Objekt zu nutzen<br>Achtung: Bezüglich der Style-Eigenschaften .background-color und .color<br>kann nur genau 1 Style deklariert werden für<br><b>alle</b> Objekte document.select.option<br><b>und</b> das document.select Objektes<br>insgesamt.<br>Es wird der Style des ersten in der Kodierungsfolge gefundenen Objektes document.select.option für alle anderen <b>und</b> das Objekt document.select selbst verwendet.<br>Es wird angeraten, <b>nur</b> im document.select das STYLE-Attribut zu verwenden.<br>Das Objekt document.select.option kann <b>nur</b> genau folgende Style-Eigenschaften erhalten: .background-color und .color |
| STYLE                    |   |
| .syncMaster              | Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master <b>nur</b> genau ein zu synchronisierendes Element haben darf (Eindeutigkeit).<br>nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior)<br>ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf!<br>siehe Objekt currTimeState und Behavior .style.time2<br>siehe .syncTolerance und .syncBehavior   |
| .systemBitrate           | wird hier nicht erklärt   |
| .systemCaptions          | wird hier nicht erklärt   |
| .systemLanguage          | Sprache festlegen für das Objekt  |
| .systemLanguage          | Standard-Sprache des Betriebssystems (Sprache der Installation) per navigator Objekt  |
| .systemOverdubOrSubtitle | wird hier nicht erklärt   |
| .tagName                 | Tag-Bezeichner des Objektes   |
| .tagUrn                  | Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut   |
| .text                    | interner Kettenwert einer Option, , also Objektes document.select.option des Objektes document.select<br>ändert nicht den angezeigten Text hinter <OPTION ...><br>ist nur ein interner Wert und beim Formular der gesendete Wert der Option   |
| .timeContainer           | Typ der Timeline des Objektes<br>siehe Objekt currTimeState und Behavior .style.time2   |
| .uniqueID                | durch den Browser automatisch-geniertes ID des Objektes   |



|                          |   |
|--------------------------|---|
| .value                   | <p>Browser generiert zu verschiedenen Zeitpunkten auch verschiedene ID, wenn Objekt mehrmals geladen wurde kann anstelle eines privat vergebenen ID als ID-Attributwert weiterverwendet werden</p> <p>Wert eines Objekt-Attributes</p> <p>Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar</p> <p>im Formular: wenn Eigenschaft .text kodiert, so wird deren Wert gesendet und der Wert von .value bleibt nur angezeigt</p> <p>wenn Eigenschaft .text nicht kodiert, so wird der Wert von .value gesendet</p> |
| <b>Methoden:</b>         |   |
| .addBehavior()           | <p>DHTML-Verhaltenseigenschaft einem Element hinzufügen</p> <p>Empfehlung: Standard-IE-Eigenschaften nutzen, da diese mit "#default#behaviorName" komplett erfasst werden und bereits im Browser implementiert sind (keine HTC-Datei nötig).</p> <p>ab IE 5.x bis unter IE 5.5</p>  |
| .appendChild()           | <p>Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern</p> <p>DOM wird geändert</p> <p>Zeiger wird zugleich immer am Ende der Collection childNodes angehängen</p> <p>Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde</p>  |
| .applyElement()          | <p>Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern</p> <p>DOM wird geändert</p> <p>Element kann selbst Kinder haben</p> <p>Element erst sichtbar, wenn Endtag (falls vorhanden) des Elementes geparkt wurde</p> <p>Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !</p>  |
| .attachEvent()           | <p>Einschalten des Registrieren eines Events durch Eventhandler</p> <p>Hinweis: Abschalten mit Methode .detachEvent()</p> <p>Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in <b>Zufallsfolge</b>, es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)</p>  |
| .clearAttributes()       | <p>alle HTML-Attribute eines Objektes entfernen</p> <p>außer ID, STYLE und per Script definierte Attribute</p> <p>Script-erzeugte Attribute nicht entfernbar</p> <p>DOM wird geändert</p>   |
| .click()                 | <p>simuliert einen Klick auf das Element und löst onclick-Event aus</p> <p>manipuliert nicht den Focus</p>  |
| .cloneNode()             | <p>Objekt klonen und Referenz des erzeugten Klone liefern</p> <p>DOM wird nicht geändert, da Klone nicht in DOM eingebunden wird (reines Neu-Instanzieren eines DOM-Objektes im Hauptspeicher außerhalb des DOM)</p>  |
| .componentFromPoint()    | <p>Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt</p> <p>auch für CSS-Layout</p> <p>onmouseover-Event hat <b>nicht die Pixelgenauigkeit</b> wie die Angaben der Methode .componentFromPoint() also wenn Event erzeugt, muss die Maus noch lange nicht genau die obengenannte Pixelposition erreicht haben</p> <p>Overbereich der Maus ist mehr als 1 Pixel gross</p> <p>beachte Einstellungen der Maus zur Cursorgeschwindigkeit etc.</p>  |
| .contains()              | <p>prüfen ob Element innerhalb eines Elementes liegt, also ob das innere, eingeschlossene Element Eltern (Eltern-Objekt, Container) hat und somit ein Kind-Objekt ist</p> <p>DOM nicht geändert</p>   |
| .detachEvent()           | <p>Abschalten des Registrieren eines Events durch Eventhandler</p> <p>wobei Registrierung mit Methode .attachEvent() aktiviert wurde</p> <p>Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichner zu, das <b>nicht</b> behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)</p>   |
| .dragDrop()              | <p>prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element</p>   |
| .fireEvent()             | <p>ein Event auslösen</p>   |
| .getAdjacentText()       | <p>Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann</p> <p>Text kann HTML-Tags enthalten, muss aber nicht</p> <p>DOM nicht geändert</p>  |
| .getAttribute()          | <p>Wert eines per HTML erzeugten Attributes liefern</p> <p>DOM nicht geändert</p>   |
| .getAttributeNode()      | <p>Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name Eigenschaft.</p> <p>Eigenschaft kann mit HTML-Anweisung erzeugt worden sein, muss aber nicht</p> <p>Eigenschaft ist selbst ein Knoten in der Attribute-Objekt-Hierarchie zum Objekt</p> <p>Wert des Attributes wird somit über die Referenz laut DOM erreichbar</p> <p>DOM nicht geändert</p>  |
| .getBoundingClientRect() | <p>Referenz auf TextRectangle-Objekt im Element holen</p>   |
| .getClientRects()        | <p>Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster</p> <p>Feld mit Index als Integer ab 0</p> <p>pro Eintrag ein Rectangle</p>  |
| .getExpression()         | <p>Wert einer Style-Eigenschaft anhand des Ausdrucks berechnen und liefern</p> <p>Style-Eigenschaft ist per Methoden</p> <p style="text-align: center;">expression() oder setExpression()<br/>zu definieren</p>   |



DOM wird nicht verändert (nur Werteveränderung), aber das Dokument-Layout (nach dem eventuellen expliziten Dokument-Refresh)

.hasChildNodes() prüfen auf Existenz von Kinder(n) als HTML-Elemente oder Textknoten (Textelemente) in einem Objekt

.insertAdjacentElement() DOM nicht geändert  
Objekt in eine Objekt einfügen und Referenz liefern, wobei die Lage definiert werden kann wenn Element bereits eingefügt vorhanden, so wird dieses nur verschoben laut Lage des Objektes im DOM nur nach dem kompletten Laden des Dokumentes möglich

.insertAdjacentHTML() DOM wird geändert  
HTML-Code und/oder Script-Code in ein Element einfügen, wobei die Lage definiert sein kann nur nach dem kompletten Laden des Dokumentes möglich  
HTML- und Script-Code müssen syntaktisch korrekt sein  
wenn nicht, so wird das Einfügen **nicht** ausgeführt  
eingefügter Code wird **nur** dann sofort geparkt und ausgeführt, wenn syntaktisch korrekt ist  
bei Script-Code: <SCRIPT DEFER .....> muss kodiert werden

.insertAdjacentText() DOM wird geändert  
Plain-Text (ohne HTML und Script) in ein Element einfügen, wobei die Lage definiert werden kann nur nach dem kompletten Laden des Dokumentes

.insertBefore() DOM wird geändert  
Objekt als Kindknoten VOR dem einem anderen Kind-Objekt einfügen und Zeiger liefern  
einzufügendes Objekt muss mit Methode createElement() erzeugt worden sein  
Achtung: NICHT anwenden für einfügen VON bzw. VOR obersten Kindknoten  
Sichtbarkeit erst wenn Ende-Tag geparkt wurde

.mergeAttributes() DOM wird geändert  
alle Attribute eines Elementes in ein anderes Element kopieren und eventuell die Attribute im Ziel mischen  
Attribute sind: HTML  
Events  
Styles  
ab IE 5.01 auch ID, NAME  
Achtung: Diese Methode ist mir Vorsicht zu geniessen !!

.normalize() DOM wird geändert  
Normalisierung des DOM zur Erreichung einer konsistenten Struktur  
Achtung: CDATA-Sections dürfen nicht enthalten sein, da diese immer Inkonsistenz erzeugen

.releaseCapture() Maus-Überwachung ausschalten für ein Objekt  
Maus-Events sind : onmousedown, onmouseup, onmousemove, onclick, ondblclick, onmouseover und onmouseout.

.removeAttribute() Hinweis: einschalten per Methode .setCapture()  
entfernen eines per HTML erzeugten Attributes  
Achtung: Der Browser unterscheidet zwischen HTML-erzeugte oder mit dieser Methode erzeugte Attribute!  
per Methode .createAttribute() erzeugte Attribute werden nicht erfasst

.removeAttributeNode() DOM wird geändert  
entfernen von Attribut, egal ob es mit oder ohne HTML-Anweisung erzeugt wurde, und Referenz auf das entfernte Attribut liefern

.removeBehavior() DOM wird geändert  
per Methode .addBehavior() einem Element hinzugefügte Verhaltenseigenschaft entfernen (stets VOR dem Entfernen des Elementes mit der zugeordneten Eigenschaft aus der Dokument-Hierarchie)

.removeChild() DOM wird geändert  
Kind-Objekt aus einem Objekt entfernen aus DOM und Referenz auf das entfernte Kind liefern  
Sichtbarkeit erst, wenn Ende-Tag geparkt wurde, also das Dokument neu geladen wurde

.removeExpression() DOM wird geändert  
Ausdruck entfernen, der für die Berechnung des Wertes einer Style-Eigenschaft als Objektreferenz der Form objekt.style.eigenschaft.dient.  
Ausdruck muss mit der Methode .setExpression() gesetzt worden sein

.removeNode() DOM wird nicht geändert  
Knoten entfernen aus DOM und Referenz auf den entfernten Knoten liefern  
Sichtbarkeit erst wenn Ende-Tag geparkt wurde

.replaceAdjacentText() DOM wird geändert  
Plain-Text (ohne HTML und Script) eines Elementes durch anderen Text ersetzen und Referenz auf den zu ersetzenden Text liefern

.replaceChild() DOM wird nicht geändert  
Kind-Objekt ersetzen durch ein Objekt  
ersetzende Objekt muss per Methode .createElement() erzeugt worden sein  
Sichtbarkeit erst wenn Ende-Tag geparkt wurde

.replaceNode() DOM wird geändert  
Objekt durch anderes Objekt komplett ersetzen und Referenz auf das komplett ersetzte Objekt liefern  
sichtbar erst mit parsen des Endetags

.setAttribute() DOM wird geändert  
Wert von vorhandenem Attribut setzen  
wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt

.setAttributeNode() DOM wird nur bei Erzeugung geändert  
Attribut einem Knoten zuweisen und Referenz liefern

.setCapture() DOM wird geändert  
Maus-Überwachung einschalten für ein Objekt



Maus-Events sind : onmousedown, onmouseup, onmousemove, onclick, ondblclick, onmouseover und onmouseout.

ab IE 5.5

Hinweis: ausschalten per Methode .releaseCapture()

.setExpression()

Wert definieren, der als Ausdruck für die Methode .getExpression() zur Berechnung einer Style-Eigenschaft als Objektreferenz der Form objekt.style.eigenschaft.

dient

Ausdruck nur als Script kodierbar

DOM wird nicht geändert

.swapNode()

Positionen von 2 Knoten im DOM tauschen (Zeigertausch)

nur sichtbar wenn Endetag geparkt

DOM wird geändert



"#default#behaviorName 4, 11  
 #text 3, 10  
 .accessKey 2  
 .add() 4  
 .addBehavior() 4, 11  
 .align 3  
 .appendChild() 11  
 .applyElement() 4, 11  
 .attachEvent() 5, 11  
 .begin 9  
 .blur() 5  
 .canHaveChildren 3, 9  
 .className 3, 9  
 .clearAttributes() 5, 11  
 .click() 5, 11  
 .clientHeight 3, 9  
 .clientLeft 3, 9  
 .clientTop 3, 9  
 .clientWidth 3, 9  
 .cloneNode() 5, 11  
 .componentFromPoint() 5, 11  
 .contains() 5, 11  
 .createAttribute() 6, 12  
 .createElement() 4, 6, 12  
 .dataFld 3  
 .dataSrc 3  
 .defaultSelected 9  
 .detachEvent() 5, 11  
 .dir 3, 9  
 .disabled 3, 9  
 .dragDrop() 5, 11  
 .end 9  
 .fireEvent() 5, 11  
 .firstChild 3, 9  
 .focus() 5  
 .form 3, 9  
 .getAdjacentText() 5, 11  
 .getAttribute() 5, 11  
 .getAttributeNode() 5, 11  
 .getBoundingClientRect() 5, 11  
 .getClientRects() 5, 11  
 .getElementsByName() 5  
 .getElementsByTagName() 5  
 .getExpression() 5, 11  
 .hasChildNodes() 5, 12  
 .hasMedia 9  
 .hideFocus 3  
 .id 3, 9  
 .index 9  
 .innerHTML 3, 9  
 .innerText 3, 9  
 .insertAdjacentElement() 6, 12  
 .insertAdjacentHTML() 6, 12  
 .insertAdjacentText() 6, 12  
 .insertBefore() 6, 12  
 .isContentEditable 3, 9  
 .isDisabled 3, 9  
 .isMultiLine 3, 9  
 .isTextEdit 3, 9  
 .label 9  
 .lang 3, 10  
 .language 3, 10  
 .lastChild 3, 10  
 .length 3  
 .mergeAttributes() 6, 12  
 .multiple 3  
 .name 3  
 .nextSibling 3, 10  
 .nodeName 3, 10  
 .nodeType 3, 10  
 .nodeValue 3, 10  
 .normalize() 6, 12  
 .offsetHeight 3, 10  
 .offsetLeft 3, 10  
 .offsetParent 3, 10  
 .offsetTop 3, 10  
 .offsetWidth 3, 10  
 .onOffBehavior 3, 10  
 .options 3  
 .outerHTML 3  
 .outerText 4  
 .ownerDocument 4, 10  
 .parentElement 4, 10  
 .parentNode 4, 10  
 .parentTextEdit 4, 10  
 .previousSibling 4, 10  
 .readyState 4, 10  
 .recordNumber 4  
 .releaseCapture() 6, 12  
 .remove() 6  
 .removeAttribute() 6, 12  
 .removeAttributeNode() 6, 12  
 .removeBehavior() 6, 12  
 .removeChild() 6, 12  
 .removeExpression() 6, 12  
 .removeNode() 6, 12  
 .replaceAdjacentText() 6, 12  
 .replaceChild() 6, 12  
 .replaceNode() 6, 12  
 .scopeName 4, 10  
 .scrollHeight 4, 10  
 .scrollIntoView() 6  
 .scrollLeft 4, 10  
 .scrollTop 4, 10  
 .scrollWidth 4, 10  
 .selected 10  
 .selectedIndex 4  
 .setActive() 6  
 .setAttribute() 6, 12  
 .setAttributeNode() 7, 12  
 .setCapture() 7, 12  
 .setExpression() 7, 13  
 .size 4  
 .sourceIndex 4  
 .style.time2 Behavior 9, 10  
 .swapNode() 7, 13  
 .syncBehavior 10  
 .syncMaster 10  
 .syncTolerance 10  
 .systemBitrate 10  
 .systemCaptions 10  
 .systemLanguage 10  
 .systemOverdubOrSubtitle 10  
 .tabIndex 4  
 .tagName 4, 10  
 .tagUrn 4, 10  
 .text 10  
 .timeContainer 10





- .type 4
- .uniqueID 4, 10
- .urns() 7
- .value 4, 11
- .z-index 1
- Abstand Objekt zum linken Rand des Fensters 3, 9
- Abstand Objekt zum oberen Rand des Fensters 3, 9
- Abstand von linken sichtbaren Randes des Umgebungsobjektes Body 4, 10
- Abstand von oberen sichtbaren Randes des Umgebungsobjektes zum oberen Body 4, 10
- aktivieren Eventdurchreichung 6
- aktueller Status Body 4, 10
- Alt + Taste 2
- anhängen Kind 11
- anhängen Knoten 11
- Anker 3, 10
- Anzahl der Feldelemente also Feldlänge 3
- Anzahl der Zeilen in einer List-Box 4
- ATOMICSELECTION 3
- Attribut automatisch erzeugen und mit Wert belegen 6, 12
- Attribut eines Knoten 7, 12
- Attribut erzeugen und mit Wert belegen 6, 12
- Attribut Wert 5, 6, 11, 12
- Attribute eines Elementes 6, 12
- Attribute HTML 6, 12
- attribute Objekt 5, 11
- attributes Collection 3, 10
- Attribut-Name 3, 10
- Attributwert Objekt 4, 11
- auslösen Event 5, 11
- Ausrichtung 3
- automatisch-genriertes ID des Body 4, 10
- Behavior .style.time2 9, 10
- Bereich ab inklusive HTML-Start bis hinter -Ende-Tag 3
- Bereich zwischen HTML-Start und -Ende-Tag Body 3, 9
- Betriebssystem Sprache Installation 10
- Betriebssystem Standard-Sprache laut Installation 10
- Bezeichner des Tag eines Objektes 4, 10
- Body Abstand von linken sichtbaren Randes des Umgebungsobjektes 4, 10
- Body Abstand von oberen sichtbaren Randes des Umgebungsobjektes zum oberen 4, 10
- Body aktueller Status 4, 10
- Body Breite des horizontalen Scrollbereiches 4, 10
- Body Content-Editierbarkeit 3, 9
- Body durch den Browser automatisch-genriertes ID 4, 10
- Body Eltern 3, 4, 10
- Body Focussierbarkeit 3
- Body Höhe des vertikalen Scrollbereiches 4, 10
- Body ID 3, 9
- Body Index des Elementes in der Tab-Tasten-Folge 4
- Body Index in der Collection document.all 4
- Body Inline-Style 4, 10
- Body Klassenreferenz 3, 9
- Body Knotentyp 3, 10
- Body Knotenwert 3, 10
- Body Mehrzeiligkeit Inhalt 3, 9
- Body Referenz auf das Elternobjekt 4, 10
- Body Referenz auf das Vorgängerkind 4, 10
- Body Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag 3, 9
- Body Referenz der Eltern 3, 10
- Body Selektierbarkeit 3, 4
- Body Sprache für Anzeige von Sonderzeichen etc. 3, 10
- Body Sprache für Script 3, 10
- Body Status 4, 10
- Body Tag-Bezeichner 4, 10
- Body Tastaturzugriff 2
- Body Textbereich des Elternobjektes 4, 10
- Body Umflussrichtung 3, 9
- Body X-Koordinate der linken oberen Ecke 3, 10
- Body X-Koordinate der rechten unteren Ecke 3, 10
- Body Y-Koordinate der linken oberen Ecke 3, 10
- Body Y-Koordinate der rechten unteren Ecke 3, 10
- Body Zeiger auf das NACHFOLGENDE Kind 3, 10
- Breite des horizontalen Scrollbereiches Body 4, 10
- Breite Objekt 3, 9
- canHaveHTML 3, 9
- childNodes Collection 3, 9
- Collection attributes 3, 10
- Collection childNodes 3, 9
- Collection document.all 1, 7
- Collection document.select.options 1, 3, 4, 7
- Collection Element entfernen 6
- Container 5, 11
- Content-Editierbarkeit Body 3, 9
- currTimeState Objekt 9, 10
- Cursorgeschwindigkeit 5, 11
- Datenfeld 4
- Datenquelle als Anker festlegen 3
- Datenquelle-Name vergeben 3
- Datenquelle-Satznummer 4
- Default-Selektionsstatus document.select.option Objekt 9
- Default-Selektionsstatus Objekt document.select.option 9
- Default-Selektionsstatus Option 9
- DEFER 6, 12
- DHTML-Eigenschaft hinzufügen 4, 11
- document Objekt des Knoten 4, 10
- document.all Collection 1, 7
- document.select Objekt 1, 3, 4, 7, 9, 10
- document.select Objekt Listbox 9
- document.select.option Objekt 7
- document.select.option Objekt Default-Selektionsstatus 9
- document.select.option Objekt interner Wert 10
- document.select.option Objekt Label 9
- document.select.option Objekt laufende Nummer in List-Box 9
- document.select.option Objekt Selektionsstatus 10
- document.select.options Collection 1, 3, 4, 7
- DOM 6, 12
- DOM Elementeigenschaft 4, 11
- DOM Normalisierung 6, 12
- Drag-Manipulation Status 5, 11
- Drop-Down-Liste 1
- durchreichen Event 6
- Eigenschaft des attribute-Objektes 5, 11
- Eigenschaft Element 6, 12
- Eigenschaft hinzufügen 4, 11
- Eigenschaften IE Standard 4, 11
- Element Attribute 6, 12
- Element den Focus wegnehmen 5
- Element Eigenschaft 6, 12
- Element entfernen aus einer Collection 6
- Element in einem Formular 3, 9
- Element innerhalb eines Elementes 5, 11
- Elementeigenschaft im DOM 4, 11
- Eltern 4, 11
- Eltern Body 3, 4, 10
- Elternknoten 4, 10





Elternobjekt Textbereich 4, 10  
 Ereignis Standardbehandlung 5, 11  
 erstes Kind Referenz 3, 9  
 erstes Kind Zeiger 3, 9  
 erzeugen Attribut automatisch und mit Wert belegen 6, 12  
 erzeugen Attribut und mit Wert belegen 6, 12  
 Event auslösen 5, 11  
 Event durchreichen 6  
 Event onblur 5  
 Event onmouseover Pixelgenauigkeit 5, 11  
 Event registrieren 5, 11  
 Eventdurchreichung aktivieren 6  
 Events registrieren 5, 11  
 Existenz Kind möglich 3, 9  
 Existenz von Kinder 5, 12  
 Feld aller Elemente mit gemeinsamer URN 7  
 Feld der Zeiger auf TextRectangle-Objekte 5, 11  
 Feldelemente Anzahl 3  
 Feldlänge 3  
 Fenster linker Rand 3, 9  
 Fenster oberer Rand 3, 9  
 Focus 5, 11  
 Focus setzen 5  
 Focus wegnehmen 5  
 Focus-Event 5  
 Focussierbarkeit Body 3  
 Formular 10  
 Formular Optionswert zum Senden 10  
 Formular senden Optionswert 10  
 Formular Zeiger auf Element 3, 9  
 hinzufügen DHTML-Eigenschaft 4, 11  
 hinzufügen Eigenschaft 4, 11  
 Höhe des vertikalen Scrollbereiches Body 4, 10  
 Höhe Objekt 3, 9  
 horizontaler Scrollbereich 4, 10  
 HTML-Attribut Wert 5, 11  
 HTML-Attribute 6, 12  
 HTML-Attribute eines Objektes 5, 11  
 HTML-Code und/oder Script-Code einfügen 6, 12  
 HTML-Tags im Objekt 3, 9  
 ID 3, 5  
 ID Body 3, 9  
 ID des Objektes 4, 10  
 IE Standard-Eigenschaften 4, 11  
 Index der Option im select Objekt 4  
 Index des Elementes in der Tab-Tasten-Folge Body 4  
 Index in der Collection document.all Body 4  
 Inkonsistenz 6, 12  
 Inline-Style Body 4, 10  
 Interaktionsfähigkeit Objekt 3, 9  
 interner Wert document.select.option Objekt 10  
 interner Wert document.select.option Objekt 10  
 interner Wert Objekt document.select.option 10  
 interner Wert Option 10  
 Kind 4, 11  
 Kind anhängen 11  
 Kind erstes 3, 9  
 Kind letztes 3, 10  
 Kind nachfolgendes 3, 10  
 Kind Name 3, 10  
 Kind Objekt 6, 12  
 Kind Vorgänger 4, 10  
 Kinder Existenz 5, 12  
 Kind-Existenz möglich 3, 9  
 Kindknoten 6, 12

Klassenreferenz Body 3, 9  
 Klick auf das Element 5, 11  
 klonen Objekt 5, 11  
 Knoten als Kind anhängen 11  
 Knoten Attribut 7, 12  
 Knoten Eltern 4, 10  
 Knoten Elternobjekt 4, 10  
 Knoten entfernen 6, 12  
 Knoten Existenz 5, 12  
 Knoten im DOM tauschen 7, 13  
 Knoten Kind 6, 12  
 Knotentyp Body 3, 10  
 Knotenwert Body 3, 10  
 konsistenten Struktur 6, 12  
 Label document.select.option Objekt 9  
 Label einer Option 9  
 Label Objekt document.select.option 9  
 laufende Nummer der Option in List-Box 9  
 Layout-Komponente eines Objektes 5, 11  
 letztes Kind Zeiger 3, 10  
 linker Rand des Fensters 3, 9  
 linker Rand des Objektes 4, 10  
 List-Box 1  
 List-Box Anzahl der Zeilen 4  
 List-Box laufende Nummer der Option 9  
 List-Box Option laufende Nummer 9  
 List-Box Zeilenanzahl 4  
 Maus-Überwachung ausschalten für ein Objekt 6, 12  
 Maus-Überwachung einschalten für ein Objekt 7, 12  
 Mehrzeiligkeit Inhalt Body 3, 9  
 multiple Auswahl select objekt 3, 4  
 MULTIPLE 1, 7  
 nachfolgendes Kind Zeiger 3, 10  
 Name des Kindes 3, 10  
 Name des Objektes 3  
 Namensraum laut XMLNS-Attribut 4, 10  
 navigator Objekt 10  
 Normalisierung des DOM 6, 12  
 oberer Rand des Fensters 3, 9  
 oberer Rand des Objektes 4, 10  
 Objekt Abstand zum linken Rand des Fensters 3, 9  
 Objekt Abstand zum oberen Rand des Fensters 3, 9  
 Objekt aktivieren 9  
 Objekt attribute 5, 11  
 Objekt Attributwert 4, 11  
 Objekt aus einem Objekt entfernen 6, 12  
 Objekt currTimeState 9, 10  
 Objekt document.select 1, 3, 4, 7, 9, 10  
 Objekt document.select Listbox 9  
 Objekt document.select.option 7  
 Objekt document.select.option Default-Selektionsstatus 9  
 Objekt document.select.option interner Wert 10  
 Objekt document.select.option Label 9  
 Objekt document.select.option laufende Nummer in List-Box 9  
 Objekt document.select.option Selektionsstatus 10  
 Objekt durch anderes Objekt komplett ersetzen 6, 12  
 Objekt ersetzen durch ein Objekt 6, 12  
 Objekt HTML-Attribute 5, 11  
 Objekt ID 4, 10  
 Objekt in eine Objekt einfügen 6, 12  
 Objekt Interaktionsfähigkeit 3, 9  
 Objekt Kind 6, 12  
 Objekt klonen 5, 11  
 Objekt Layout-Komponente 5, 11



Objekt linker Rand 4, 10  
 Objekt Maus-Überwachung ausschalten 6, 12  
 Objekt Maus-Überwachung einschalten 7, 12  
 Objekt mit HTML-Tags 3, 9  
 Objekt Name 3  
 Objekt navigator 10  
 Objekt oberer Rand 4, 10  
 Objekt Plain-Text 4  
 Objekt scrollen 6  
 objekt select Auswahl multiple 3, 4  
 Objekt select Index der Option 4  
 Objekt select List-Box 4  
 objekt select multiple Auswahl 3, 4  
 Objekt select Optionindex 4  
 Objekt sichtbar 6  
 Objekt Sprache 10  
 Objekt Status 4, 10  
 Objekt Tag-Bezeichner 4, 10  
 Objekt Text 5, 11  
 Objekt Textbereich 4, 10  
 Objekt Umgebung 4, 10  
 Objekt Viewbereich 6  
 Objekt Zeiger auf TextRectangle 5, 11  
 Objekt Zugriff per Alt + Taste 2  
 Objektaktivität 9  
 Objekt-Breite 3, 9  
 Objekthöhe 3, 9  
 onblur 5  
 onclick 5, 7, 11, 13  
 ondblclick 7, 13  
 onmousedown 6, 7, 12, 13  
 onmousemove 6, 7, 12, 13  
 onmouseout 6, 7, 12, 13  
 onmouseover 6, 7, 12, 13  
 onmouseover Pixelgenauigkeit 5, 11  
 onmouseup 6, 7, 12, 13  
 OPTION 1, 7, 10  
 Option aus einer Selektion 7  
 Option innerhalb einer Optionsgruppe 4  
 Option Label 9  
 Option laufende Nummer 9  
 Option laufende Nummer in List-Box 9  
 Optionen 1, 7  
 Optionindex im select Objekt 4  
 Optionsgruppe 4  
 Plain-Text 6, 12  
 Plain-Text im Objekt 4  
 Positionen von 2 Knoten im DOM tauschen 7, 13  
 Rahmen 3, 9  
 Referenz auf Feld der Zeiger auf TextRectangle-Objekte 5, 11  
 Referenz auf Bereich ab inklusive HTML-Start bis hinter -  
 Ende-Tag 3  
 Referenz auf das document Objekt des Knoten 4, 10  
 Referenz auf das Elternobjekt Body 4, 10  
 Referenz auf das ERSTE Kind 3, 9  
 Referenz auf das LETZTE Kind 3, 10  
 Referenz auf das NACHFOLGENDE Kind 3, 10  
 Referenz auf das Vorgängerkind 4, 10  
 Referenz auf das Vorgängerkind Body 4, 10  
 Referenz auf den gesamten Plain-Text im Objekt 4  
 Referenz auf Elternknoten 4, 10  
 Referenz auf Feld aller Elemente mit gemeinsamer URN 7  
 Referenz auf TextRectangle 5, 11  
 Referenz der Eltern Body 3, 10

registrieren eines Events 5, 11  
 registrieren eines Events 5, 11  
 Script-Code einfügen 6, 12  
 Scrollbalken 3, 9  
 Scrollbereich horizontal 4, 10  
 Scrollbereich vertikal 4, 10  
 SELECT 1, 7  
 select objekt Auswahl multiple 3, 4  
 select Objekt Index der Option 4  
 select Objekt List-Box 4  
 select objekt multiple Auswahl 3, 4  
 select Objekt Optionindex 4  
 SELECTED 1, 7  
 Selektierbarkeit Body 3, 4  
 Selektion und Option 7  
 Selektionsstatus Default document.select.option Objekt 9  
 Selektionsstatus Default Objekt document.select.option 9  
 Selektionsstatus document.select.option Objekt 10  
 Selektionsstatus Objekt document.select.option 10  
 Selektionsstatus Option 10  
 setCapture() 6, 12  
 setExpression() 6, 12  
 setzen Focus 5  
 Sprache für Anzeige von Sonderzeichen etc. Body 3, 10  
 Sprache für Script Body 3, 10  
 Sprache Installation Betriebssystem 10  
 Sprache zum Objekt 10  
 Standardbehandlung Ereignis 5, 11  
 Standard-IE-Eigenschaften 4, 11  
 Standard-Sprache des Betriebssystems 10  
 Status Body 4, 10  
 Status des Objektes 4, 10  
 Status Drag-Manipulation 5, 11  
 STYLE 4, 10  
 Style-Eigenschaft Wert 5, 6, 7, 11, 12, 13  
 TABINDEX 5  
 TAG-Bezeichner 3, 10  
 Tag-Bezeichner Body 4, 10  
 Tag-Bezeichner des Objektes 4, 10  
 Tastaturzugriff auf Body 2  
 Tastaturzugriff auf ein Objekt per Alt + Taste 2  
 Tastenkombination 2  
 Text eines Objektes 5, 11  
 Textbereich des Elternobjektes 4, 10  
 Textbereich des Elternobjektes Body 4, 10  
 TextRectangle 5, 11  
 TIMEACTION 9  
 Timeline 9  
 Timeline Synchronisierung der Animation des im Container  
 liegenden Elementes auf Timeline 10  
 Umflussrichtung Body 3, 9  
 Umgebungsobjekt 4, 10  
 Uniform Resource Name 4, 10  
 UNSELECTABLE 4  
 URN 4, 10  
 VALUE 4, 11  
 vertikaler Scrollbereich 4, 10  
 Viewbereich Objekt 6  
 Vorgängerkind 4, 10  
 Wert einer Style-Eigenschaft 5, 6, 7, 11, 12, 13  
 Wert eines Attributes belegen 6, 12  
 Wert eines Objekt-Attributes 4, 11  
 Wert eines per HTML-Attributes 5, 11  
 Wert vom Attribut 6, 12  
 X-Koordinate der linken oberen Ecke Body 3, 10



X-Koordinate der rechten unteren Ecke Body 3, 10  
XMLNS 4, 10  
Y-Koordinate der linken oberen Ecke Body 3, 10  
Y-Koordinate der rechten unteren Ecke Body 3, 10  
Zeiger auf Element in einem Formular 3, 9  
Zeiger auf das ERSTE Kind 3, 9  
Zeiger auf das LETZTE Kind 3, 10

Zeiger auf das NACHFOLGENDE Kind Body 3, 10  
Zeiger auf das Vorgängerkind 4, 10  
Zeiger auf Elternknoten 4, 10  
Zeiger auf TextRectangle-Objekt 5, 11  
Zeigertausch 7, 13  
Zeilenanzahl List-Box 4  
Zugriff per Alt + Taste 2

