

## a Objekt des Internet Explorer

Hypertext-Link (Linkt mit Textangabe für User) also Anker:

Es kann auch ein Image im Text angegeben werden (zusätzlich oder anstelle des Textes).

Bei Image: Border unsichtbar durch Eigenschaft .border auf 0 setzen.

Achtung: keine Tabelle einbindbar (kein Objekt table) !!

Script einbindbar

benötigt **immer** die kodierte Eigenschaft .href oder .name

nicht verschachtelbar

Objekt mit Timeline nur wirksam, wenn diese aktiv ist.

siehe auch link Objekt

### Erzeugung:

Beispiele:

```
<A ID="ID_A" HREF="http://www.test.de.com">gehe zu test.de</A>

<A ID="ID_A" HREF="home.htm">home.htm</A>

<A ID="ID_A" TARGET="name_des_fensters" HREF="test.htm">im Fenster oeffnen</A>

<A ID="ID_A" HREF="http://www.test.de "><IMG SRC="test.gif">Link mit Image und Text</A>

<A ID="ID_A" NAME="anchor_name">Eine Anker als Sprungziel</A>

<A ID="ID_A" HREF="#anchor_name">Anker ausführen </A>

<A ID="ID_A" HREF="javascript:window.open()">Javascript</A>

<A HREF="text.doc">Den Text als Word-Dokument herunterladen</A>

<HEAD>
<STYLE>
.an {text-decoration: underline overline; color:blue;}
.aus {text-decoration: none; color:black;}
</STYLE>
</HEAD>
<BODY>
<A
    HREF="test.htm"
    CLASS="aus"
    onmouseover="this.className='an';"
    onmouseout="this.className='aus';"
>
</A>
</BODY>
```

### Zugriff:

**ID\_A**.eigenschaft

**ID\_A**.methode()

Beispiel 1:

```
<HTML XMLNS:IE>
<HEAD>
<SCRIPT>
    function NachDemDownload(DownLoadedFileContent)
    {alert (DownLoadedFileContent);}
</SCRIPT>
</HEAD>
<BODY>
<IE:Download ID="ID_IETag" STYLE="behavior:url(#default#download)" >
<P>
Click
<A HREF="javascript:ID_IETag.startDownload('download.htm', NachDemDownload)">
hier
</A>
zum Start des Downloads dieser Seite.
</BODY>
</HTML>
```

Beispiel 2 für Link ohne Linkrahmen bei IE  
NS ab 6.x

Achtung: Eventuell wird Navigation per TAB-Taste nicht mehr möglich sein !

Dies ist ein <A HREF="#" onfocus="if(this.blur){this.blur();}">Link ohne Rahmen bei Focuserhalt</A>



Dies ist ein <A HREF="seite.html" onclick="this.blur()">Link ohne Rahmen nach klick</A>

Beispiel 3 für Link ohne Unterstreichug:

```

<HTML>
<HEAD>
  <STYLE TYPE="text/css">
    A { text-decoration: none; }
  </STYLE>
</HEAD>
<BODY>
  <A HREF="http://www.test.de/">Ein Link ohne Unterstreichug</A>
</BODY>
</HTML>

```

**Alternative:**

Beispiel:

```

<FORM ACTION="test.htm">
  <INPUT TYPE="submit" VALUE="Gehe zu test.htm">
</FORM>

```

**Eigenschaften:**

.accessKey	Tastaturzugriff auf ein Objekt per Alt + Taste bei Ausführung der Tastenkombination wird das Sprungziel wird fokussiert die Sprungquelle defokussiert das Focus-Ereignis ausgelöst
ATOMICSELECTION	vor IE 5.x funktioniert Tastenkombination nicht mit jedem Objekt
.begin	Selektierbarkeit des Objektes einstellen Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction 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
.charset	Zeichensatz zum Encoden eines Objektes im Dokument
.className	Klassenreferenz, Klassenname
.clientHeight	Objekthöhe in Pixel ohne Abstand zum Umgebungsobjekt ohne Rahmen 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 ohne Rahmen ohne Scrollbalken
.contentEditable	Objekt-Content-Editierbarkeit (auch wenn Objekt kein Layout hat) Content = Beziehung des Objektes zum Umfeld z.B. bezüglich Layout etc.
.coords	Koordinaten eines Objektes in Abhängigkeit zur Eigenschaft .shape
.dataFld	Datenquelle-Name vergeben (ID)
.dataSrc	Datenquelle als Anker festlegen
.dir	Umflussrichtung
.disabled	Interaktionsfähigkeit nur wenn sichtbar so User-Interaktion möglich
.end	Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur siehe Objekt currTimeState und Behavior .style.time2
.firstChild	Zeiger auf das ERSTE Kind laut childNodes-Collection eines Objektes
.hash	Teil des Wertes der Eigenschaft .href also Teil der Url als Anker Teil folgt direkt hinter dem Nummernkreuz # und ohne Nummernkreuz
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Fokussierbarkeit
.host	<b>Hostname</b> und <b>Port</b> einer Location oder Url in der Form " <b>hostname:port</b> "
.hostname	<b>Hostname</b> einer Location oder Url in der Form " <b>hostname:port</b> " kann Domain oder IP sein
.href	Ziel-Url oder Anker als Sprungziel (z.B. <A>-Tag) siehe auch Eigenschaften .rel und .rev
.hreflang	Sprachcode des Objektes laut RFC1766
.id	Bezeichner des Objektes für Referenzierung des Objektes (Zeiger, ID, logischer Objektname) Hinweis: Browser erzeugt pro Objekt ein internes ID, das per Eigenschaft .uniqueID ermittelt und anstelle der Eigenschaft .id verwendet werden kann (falls Browser und betroffenes Objekt die Eigenschaft .uniqueID kennen). Zeiger aus ID bilden var Zeiger = eval(object.id);
.innerHTML	Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag



	ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B. <B> dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes, also nach dem Laden des Objektes aber wirksam erst mit parsen des HTML-Endetag						
.innerText	Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B. <B> dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes, also nach dem Laden des Objektes aber wirksam erst mit parsen des HTML-Endetag						
.isContentEditable	Editierbarkeit des Objekt-Content (auch wenn kein Layout hat) Content = Beziehung des Objektes zum Umfeld z.B. bezüglich Layout etc.						
.isDisabled	Interaktionsfähigkeit 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						
.Methods	Liste der vom Objekt unterstützten HTTP-Methoden						
.name	Name des Objektes (nicht ID !!!) muss beim Formular für alle zu sendenden Felder kodiert sein !! 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) also TAG-Bezeichner, Attribut-Name; #text für Anker						
.nodeType	Knotentyp laut attributes Collection						
.nodeValue	Knotenwert (Wert des Kindes, Node, Elementes) nur für Text- und Attribut-Elemente 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 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 Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound						
.outerHTML	Referenz auf Bereich ab inklusive HTML-Start bis hinter -Ende-Tag wirksam mit parsen des Ende-Tag nur nach kompletten Einlesen des Dokumentes nutzbar						
.outerText	Referenz auf den gesamten Plain-Text im Objekt 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						
.pathname	Datei und Pfad eines Objektes						
.port	<b>Port</b> einer Location oder Url in der Form "hostname: <b>port</b> " basierend auf dem aktuellen <b>Protokoll</b> laut Eigenschaft .protocol z.B. <table border="0" style="display: inline-table; vertical-align: middle;"> <tr> <td style="border-bottom: 1px solid black; padding: 0 5px;">Standard-Ports</td> <td style="border-bottom: 1px solid black; padding: 0 5px;">Protokoll</td> </tr> <tr> <td style="padding: 0 5px;">21</td> <td style="padding: 0 5px;">FTP</td> </tr> <tr> <td style="padding: 0 5px;">80</td> <td style="padding: 0 5px;">HTTP</td> </tr> </table>	Standard-Ports	Protokoll	21	FTP	80	HTTP
Standard-Ports	Protokoll						
21	FTP						
80	HTTP						
.previousSibling	Referenz auf das Vorgängerkind						
.protocol	Protokoll-Teil einer Url inklusive http und ftp liefern						
.readyState	aktueller Status des Objektes beim Füllen des Objektes mit Daten						
.recordNumber	Datenquelle-Satznummer eines Datenfeldes						
.rel	Beziehung zwischen Objekt in Quellseite und Nachfolger-Zielseite laut Objekt Objekt ist ein Link z.B. A-Tag oder Link-Tag nur wichtig für Suchmaschine: dient als Steuerungshinweis für Suchmaschinen beim durchforsten der Seiten und Nachfolgerseiten vorrangig kodiert in <A> oder <LINK> es <b>muss</b> zugleich die Eigenschaft .href kodiert und mit <b>gültigen</b> Wert belegt sein nur für einen Anker ist zugleich Eigenschaft .rev kodierbar (falls sinnvoll)						
.rev	Beziehung zwischen Objekt in Quellseite und Vorgänger-Zielseite laut Objekt Objekt ist ein Link z.B. A-Tag oder Link-Tag nur wichtig für Suchmaschine: dient als Steuerungshinweis für Suchmaschinen beim durchforsten der Seiten und Vorgängerseiten vorrangig kodiert in <A> oder <LINK> es <b>muss</b> zugleich die Eigenschaft .href kodiert und mit <b>gültigen</b> Wert belegt sein						



nur für einen Anker ist zugleich Eigenschaft .rev kodierbar (falls sinnvoll)

.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 nur nutzbar nach dem kompletten Laden des Dokumentes

.scrollTop Abstand von oberen sichtbaren Randes des Umgebungsobjektes zum oberen Rand des Objektes nur nutzbar nach dem kompletten Laden des Dokumentes

.scrollWidth Breite des horizontalen Scrollbereiches, also Abstand von linken und rechten sichtbaren Rand des Umgebungsobjektes

.search Teil des Wertes des Eigenschaft .href  
 Teil folgt direkt dem Fragezeichen  
 wird als Querystring oder Formdata bezeichnet  
 hat nichts mit der Suche auf Webseiten zu tun  
 Hinweis: Fragezeichen-Anhang an der HREF dient zur Übergabe von String-Werten einer Webseite an eine andere:  
 Quellseite besitzt HREF mit " ...?....."  
 ruft Zielseite mit diesem HREF auf  
 Zielseite wurde von Quellseite aufgerufen  
 liest den Teil von HREF hinter dem ? als Zeichenkettendaten

.shape Form/Gestalt eines Objektes  
 siehe auch Eigenschaft .coords

.sourceIndex Index des Objektes in der Collection document.all

STYLE direkt im HTML-Element kodierter Style (Inline-Style)  
 Hinweis: für Scripting ist das Style-Objekt zu nutzen

.syncMaster Synchronisierung der Animation des im Container liegenden Elementes auf Timeline (Container ist Master in der Synchronisierung), wobei ein Master **nur** genau ein zu synchronisierendes Element haben darf (Eindeutigkeit).  
 nur sinnvoll bei Zwangssynchronisierung (siehe .syncBehavior)  
 ersetzt die Eigenschaft clockSource, da diese deprecated ist und nicht mehr verwendet werden darf !  
 siehe Objekt currTimeState und Behavior .style.time2  
 siehe .syncTolerance und .syncBehavior

.systemBitrate wird hier nicht erklärt

.systemCaptions wird hier nicht erklärt

.systemLanguage Sprache festlegen für das Objekt

.systemOverdubOrSubtitle wird hier nicht erklärt

.tabIndex Index des Elementes in der Tab-Tasten-Folge für Anspringen des Dokumentes  
 Anspringen verbunden mit Focus erhalten  
 --> Ereignisse werden ausgelöst !!  
 unter IE 5.x onblur, onfocus  
 ab IE 5.x onblur, onfocus, onkeydown, onkeypress, onkeyup

Anspringen default per TAB-Taste für A, BODY, BUTTON, FRAME, IFRAME, IMG, INPUT, SELECT, TEXTAREA

Anspringen default nicht per TAB-Taste für APPLET, DIV, FRAMESET, SPAN, TABLE, TD

.tagName Tag-Bezeichner des Objektes

.tagUrn Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut

.target Name des Ziel-Fenster bzw. Ziel-Frame

.timeContainer Typ der Timeline des Objektes  
 siehe Objekt currTimeState und Behavior .style.time2

.title Tooltip-Text bei Mouse over über Objekt

.type MIME-Typ des Objektes (Multipurpose Internet Mail Extension)  
 wenn die Eigenschaft .classid nicht kodiert wird, dann immer .type kodieren bzw. die im Browserstandard enthaltenen MIME verwenden  
 Hinweis: MIME wird von Simple Mail Transfer Protocol (SMTP) benutzt  
 z.B. bei Audio, Images, Video, Texten

.uniqueID durch den Browser automatisch generiertes ID des Objektes  
 Browser generiert zu verschiedenen Zeitpunkten auch verschiedene ID, wenn Objekt mehrmals geladen wurde kann anstelle eines privat vergebenen ID als ID-Attributwert weiterverwendet werden

UNSELECTABLE Selektionsfähigkeit eines Objektes

.urn Uniform Resource Name (URN) des Dokumentes

**Methoden:**

.addBehavior() DHTML-Verhaltenseigenschaft einem Element hinzufügen  
 Empfehlung: Standard-IE-Eigenschaften nutzen, da diese mit "#default#behaviorName" komplett erfasst werden und bereits im Browser implementiert sind (keine HTC-Datei nötig).  
 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  
 DOM wird geändert



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



	(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 <b>nicht</b> ausgeführt eingefügter Code wird <b>nur</b> 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
.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, 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 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
.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
.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
.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 DOM wird geändert
.setCapture()	Maus-Überwachung einschalten für ein Objekt Maus-Events sind : onmousedown, onmouseup, onmousemove, onclick, ondblclick, onmouseover und onmouseout. ab IE 5.5
.setExpression()	Hinweis: ausschalten per Methode .releaseCapture() 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

