

input Objekt und seine Varianten im Internet Explorer

- input Objekt und seine Varianten im Internet Explorer1**
- input button Objekt 2**
- input checkbox Objekt..... 6**
- input file Objekt 11**
- input hidden Objekt..... 15**
- input image Objekt 18**
- input password Objekt 23**
- input radio Objekt..... 27**
- input reset Objekt 32**
- input submit Objekt..... 36**
- input text Objekt 41**

Varianten des Input Controls (Eingabe-Steuerelementes) z.B. im Formular festlegbar per TYPE-Attribut:

- button
- checkbox
- file
- hidden
- image
- password
- radio
- reset
- submit
- text

wobei der Wert des TYPE-Attributes **wahlweise** in " " bzw. ' ' oder ohne " " bzw. ' ' kodiert werden kann.

Das input Objekt ist Prototyp für das Objekt document.form.input nutzt weitere Objekte wie button und img.

Sämtliche Input-Varianten besitzen das ID-Attribut, über das der Script-Zugriff möglich ist.

Nachfolgende Syntax-Beschreibungen zeigen die HTML-Verwendung der Input-Varianten, wobei aus Gründen der Übersichtlichkeit auf die Kodierung des ID-Attributes verzichtet wurde. Eigenschaften und Methoden **nur** beim Internet Explorers.

Erzeugung in HTML:

Beispiel Alle Elemente im Formular müssen **NAME**-Attribut erhalten, wenn die Daten der Elemente gesendet werden sollen !

```

<FORM ACTION="http://intranet/test" METHOD=POST>
  <INPUT TYPE="TEXT"      NAME="CONTROL1" VALUE="Ihr Name">
  <BR>
  <P>Password</P>
  <INPUT TYPE="PASSWORD"      NAME="CONTROL2">
  <P>Farbe-Auswahl</P>
  <BR>
  <INPUT TYPE="RADIO"      NAME="CONTROL3" VALUE="0" CHECKED>Rot
  <INPUT TYPE="RADIO"      NAME="CONTROL3" VALUE="1">Gelb
  <INPUT TYPE="RADIO"      NAME="CONTROL3" VALUE="2">Blau
  <P>Ihr Kommentar</P>
  <BR>
  <INPUT TYPE="TEXT"      NAME="CONTROL4" SIZE="20,5" MAXLENGTH="250">
  <BR>
  <INPUT TYPE="CHECKBOX"      NAME="CONTROL5" CHECKED>Senden
  <BR>
  <INPUT TYPE="SUBMIT" VALUE="OK">
  <INPUT TYPE="RESET" VALUE="Reset">
</FORM>

```

Eigenschaften beim Internet Explorer:

Diese Eigenschaften sind nicht für alle INPUT-Varianten sinnvoll anwendbar Bsp.: Loop für ein Image-Map



- .accept Liste von Multipurpose Internet Mail Extensions (MIME)-Typen
- .align Ausrichtung
- .alt Alternativer Text als Tooltip
- .complete Zustand des Ladens des Objektes
- .dynsrc Adresse von Videoclip oder VRML
- .hspace horizontaler Abstand in Pixel zum Elternobjekt
- .lowsrc Url des Images in geringerer Auflösung
- .start Start eines Videoclips oder VRML-Datei
- .useMap Url oder Anker für client-seitige Image-Map
- .value Wert eines Objekt-Attributes
Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
- .vspace vertikaler Abstand in Pixel zum Elternobjekt

Methoden beim Internet Explorer:

keine

input button Objekt

Taste-Control-Element (Button-Control-Element)

Hinweis: Es gibt noch das button Objekt (siehe dort)

Erzeugung in HTML:

<INPUT TYPE=button> auch als String "button" kodierbar

Beispiel

```
<INPUT TYPE=button
VALUE="betaetigen!"
NAME="button1"
onClick="alert('Das input button Objekt in Aktion')"
>
```

Eigenschaften beim Internet Explorer:

- .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
- ATOMICSELECTION Selektierbarkeit des Objektes einstellen
- .begin 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
- .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.
- .dataFld Datenquelle-Name vergeben (ID)
- .dataFormatAs Datenquelle-Anzeigeart
- .dataSrc Datenquelle als Anker festlegen
- .defaultValue Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
- .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
- .form Zeiger auf das Formular (Formular als Container)
ab IE 6.x für Elemente fieldSet, label, legend
- .hasMedia Objekt ist HTML-Media-Objekt
siehe Objekt currTimeState und Behavior .style.time2
- .hideFocus Focussierbarkeit
- .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).



.isContentEditable	Zeiger aus ID bilden var Zeiger = eval(object.id); 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
.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
.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 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
.size	Dimension eines Input-Objektes
.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 currentTimeState 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
- .timeContainer Typ der Timeline des Objektes
siehe Objekt currTimeState und Behavior .style.time2
- .title Tooltip-Text bei Mouse over über Objekt
- .type Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
- .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
- .value Wert eines Objekt-Attributes
Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
- .width Breite des Objektes in Pixel

Methoden beim Internet Explorer:

- .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
- .appendChild() Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern
DOM wird geändert
Zeiger wird zugleich immer am Ende der Collection childNodes angehängen
Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag
(falls vorhanden) des Knoten geparkt wurde
- .applyElement() Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz
laut DOM liefern
DOM wird geändert
Element kann selbst Kinder haben
Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde
Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im
im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !
- .attachEvent() Einschalten des Registrieren eines Events durch Eventhandler
Hinweis: Abschalten mit Methode .detachEvent()
Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider
NICHT verkettet sondern in **Zufallsfolge**, es sei denn
die Handler prüfen ihre Aufruffolge (muss programmiert werden)
- .blur() 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
- .clearAttributes() alle HTML-Attribute eines Objektes entfernen
außer ID, STYLE und per Script definierte Attribute
Script-erzeugte Attribute nicht entfernbar
DOM wird geändert
- .click() simuliert einen Klick auf das Element und löst onclick-Event aus
manipuliert nicht den Focus
- .cloneNode() 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)
- .componentFromPoint() Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt
auch für CSS-Layout
onmouseover-Event hat **nicht die Pixelgenauigkeit** wie die Angaben der Methode .componentFromPoint()
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.
- .contains() 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
- .createTextRange() Textbereich erzeugen
- .detachEvent() Abschalten des Registrieren eines Events durch Eventhandler
wobei Registrierung mit Methode .attachEvent() aktiviert wurde
Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichner
zu, das **nicht** behandelt werden soll, falls es für das Window-Objekt auftritt
(also Standardbehandlung aktiv)
- .dragDrop() prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element
- .fireEvent() ein Event auslösen
- .focus() Focus setzen und Focus-Event auslösen



nur nach dem kompletten Laden des Dokumentes
vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen

.getAdjacentText() Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann
Text kann HTML-Tags enthalten, muss aber nicht
DOM nicht geändert

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.getAttributeNode() Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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

.getBoundingClientRect() Referenz auf TextRectangle-Objekt im Element holen

.getClientRects() Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster
Feld mit Index als Integer ab 0
pro Eintrag ein Rectangle

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren
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
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, onmousemove, onclick, ondblclick,
onmouseover und onmouseout.
Hinweis: einschalten per Methode .setCapture()

.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



.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
.scrollIntoView()	DOM wird geändert Objekt derart scrollen, dass es im Fenster für User sichtbar wird Objekt muss an sich schon renderbar sein
.select()	Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode .focus() verwenden)
.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
.swapNode()	DOM wird nicht geändert Positionen von 2 Knoten im DOM tauschen (Zeigertausch) nur sichtbar wenn Endetag geparkt DOM wird geändert

input checkbox Objekt

Checkbox-Control
Standardwert ist "on"

Style-Wert zu height und width ab IE 5.x (innere Höhe bzw. Breite): wenn height bzw. width
>= 20 Pixel

	Dicke Padding um die Checkbox	4 Pixel
	innere Höhe bzw innere Breite	8 Pixel
< 20 Pixel	und > 13 Pixel	
	Dicke Padding um die Checkbox	(height -13) / 2 Pixel bzw. (width -13) / 2 Pixel
	innere Höhe bzw innere Breite	unverändert übernommen
< 13 Pixel		
	Dicke Padding um die Checkbox	0 Pixel
	innere Höhe bzw innere Breite	unverändert übernommen

Erzeugung in HTML:

<INPUT TYPE=checkbox> auch als String "checkbox" kodierbar

Beispiel

```

<SCRIPT>
function Anzeige1()
{ID_Span.insertAdjacentHTML("Checkbox 1 aktiv");}

function Anzeige2()
{ID_Span.insertAdjacentHTML("Checkbox 2 aktiv");}
</SCRIPT>
<INPUT TYPE=checkbox
NAME="checkbox1"
CHECKED
onclick="Anzeige1()"
>Aktion 1
<INPUT TYPE=checkbox
NAME="checkbox2"

```



```

onclick="Anzeige2()"
>Aktion 2
<SPAN ID="ID_Span">
  Test-Text
</SPAN>

```

Eigenschaften beim Internet Explorer:

.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
ATOMICSELECTION	Selektierbarkeit des Objektes einstellen
.begin	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
.checked	Selektionsstatus des Checkbox-Control bzw. Radio Button-Control bezüglich Selektion durch User
.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
.dataFld	Datenquelle-Name vergeben (ID)
.dataSrc	Datenquelle als Anker festlegen
.defaultChecked	Selektionsstatus des Checkbox-Control bzw. Radio Button-Control bzw. Radio Button-Control bezüglich Standard-Selektion
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.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);
.indeterminate	Grauzustand (Dimmed) und Selektiertheit des Checkbox-Control
.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
.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 parse 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
.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 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
.size	Dimension eines Input-Objektes
.sourceIndex	Index des Objektes in der Collection document.all
.status	Selektionsstatus eines Control-Elementes Control-Element: kann durch User interaktiv verändert werden Hinweis: für Scripting ist das Style-Objekt zu nutzen
STYLE	direkt im HTML-Element kodierter Style (Inline-Style) 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
.syncMaster	
.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
.timeContainer	Typ der Timeline des Objektes siehe Objekt currTimeState und Behavior .style.time2
.title	Tooltip-Text bei Mouse over über Objekt
.type	Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
.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
.value	Wert eines Objekt-Attributes Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
.width	Breite des Objektes in Pixel

Methoden beim Internet Explorer:



.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
.appendChild()	Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern DOM wird geändert Zeiger wird zugleich immer am Ende der Collection childNodes angehängen Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde
.applyElement()	Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern DOM wird geändert Element kann selbst Kinder haben Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !
.attachEvent()	Einschalten des Registrieren eines Events durch Eventhandler Hinweis: Abschalten mit Methode .detachEvent() Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in Zufallsfolge , es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)
.blur()	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
.clearAttributes()	alle HTML-Attribute eines Objektes entfernen außer ID, STYLE und per Script definierte Attribute Script-erzeugte Attribute nicht entfernen DOM wird geändert
.click()	simuliert einen Klick auf das Element und löst onclick-Event aus manipuliert nicht den Focus
.cloneNode()	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)
.componentFromPoint()	Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt auch für CSS-Layout onmouseover-Event hat nicht die Pixelgenauigkeit wie die Angaben der Methode .componentFromPoint() also wenn Event erzeugt, muss die Maus noch lange nicht genau die obengenannte Pixelposition erreicht haben Overbereich der Maus ist mehr als 1 Pixel gross beachte Einstellungen der Maus zur Cursorgeschwindigkeit etc.
.contains()	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
.detachEvent()	Abschalten des Registrieren eines Events durch Eventhandler wobei Registrierung mit Methode .attachEvent() aktiviert wurde Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichnung zu, das nicht behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)
.dragDrop()	prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element
.fireEvent()	ein Event auslösen
.focus()	Focus setzen und Focus-Event auslösen nur nach dem kompletten Laden des Dokumentes vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen
.getAdjacentText()	Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann Text kann HTML-Tags enthalten, muss aber nicht DOM nicht geändert
.getAttribute()	Wert eines per HTML erzeugten Attributes liefern DOM nicht geändert
.getAttributeNode()	Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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
.getBoundingClientRect()	Referenz auf TextRectangle-Objekt im Element holen
.getClientRects()	Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster Feld mit Index als Integer ab 0 pro Eintrag ein Rectangle
.getExpression()	Wert einer Style-Eigenschaft anhand des Ausdrucks berechnen und liefern Style-Eigenschaft ist per Methoden expression() oder setExpression() zu definieren 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 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, 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
.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
.select()	Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode .focus() verwenden)
.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



.setAttributeNode() DOM wird nur bei Erzeugung geändert
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
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

.swapNode() DOM wird nicht geändert
Positionen von 2 Knoten im DOM tauschen (Zeigertausch)
nur sichtbar wenn Endetag geparkt
DOM wird geändert

input file Objekt

File Upload-Control mit Dialogbox

ab IE 4.x

folgende Voraussetzungen **müssen** erfüllt werden:

Control muss in einem Formular liegen

NAME-Attribut muss kodiert sein

Formular mit Attribut METHOD auf "post" gesetzt

ENCTYPE auf "multipart/form-data"

Server muss den Enctype "multipart/form-data" verarbeiten können (eventuell CGI- oder ASP-Script dort nötig)

Erzeugung in HTML:

Beispiel mit Serverscript per ASP:

HTML-Dokument:

```
<FORM NAME="Formular"
ACTION="script.asp"
ENCTYPE="multipart/form-data"
METHOD="post"
>
  <INPUT TYPE="file" NAME="testdatei">
  <INPUT TYPE="submit" VALUE="Upload der Datei">
</FORM>
```

script.asp:

```
<% @ LANGUAGE = JScript %>
<%
  Response.buffer=true;
%>
<HTML>
<BODY>
  <H1>Upload Status</H1>
  <P>
    Zielort: <% Response.Write(Request.Form("TargetURL")) %>
  </P>
  <%
    Response.write("<P>Dateiname: " + Request.Form("FileName") + "</P>");
    Response.write("<P>Dateigroesse: " + Request.Form("FileSize") + "</P>");
    Response.write("<P>Dateipfad: " + Request.Form("FilePath") + "</P>");
  %>
</BODY>
</HTML>
```

Eigenschaften beim Internet Explorer:

.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

ATOMICSELECTION
.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

.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
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.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);
.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
.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
.outerHTML	Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound 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
.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 nur nutzbar nach dem kompletten Laden des Dokumentes
.scrollTop	Abstand von oberen sichtbaren Randes des Umgebungsobjektes zum oberen Rand des Objektes



.scrollWidth	nur nutzbar nach dem kompletten Laden des Dokumentes Breite des horizontalen Scrollbereiches, also Abstand von linken und rechten sichtbaren Rand des Umgebungsobjektes
.size	Dimension eines Input-Objektes
.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
.timeContainer	Typ der Timeline des Objektes siehe Objekt currTimeState und Behavior .style.time2
.title	Tooltip-Text bei Mouse over über Objekt
.type	Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
.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
.value	Wert eines Objekt-Attributes Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
.width	Breite des Objektes in Pixel
Methoden beim Internet Explorer:	
.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
.appendChild()	Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern DOM wird geändert Zeiger wird zugleich immer am Ende der Collection childNodes angehängen Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde
.applyElement()	Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern DOM wird geändert Element kann selbst Kinder haben Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !
.attachEvent()	Einschalten des Registrieren eines Events durch Eventhandler Hinweis: Abschalten mit Methode .detachEvent() Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in Zufallsfolge , es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)
.blur()	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
.clearAttributes()	alle HTML-Attribute eines Objektes entfernen außer ID, STYLE und per Script definierte Attribute Script-erzeugte Attribute nicht entfernbar DOM wird geändert



.click() simuliert einen Klick auf das Element und löst onclick-Event aus
manipuliert nicht den Focus

.cloneNode() 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)

.componentFromPoint() Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt
auch für CSS-Layout
onmouseover-Event hat **nicht die Pixelgenauigkeit** wie die Angaben der Methode .componentFromPoint()
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.

.contains() 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

.detachEvent() Abschalten des Registrieren eines Events durch Eventhandler
wobei Registrierung mit Methode .attachEvent() aktiviert wurde
Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichnung zu, das **nicht** behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)

.dragDrop() prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element

.fireEvent() ein Event auslösen

.focus() Focus setzen und Focus-Event auslösen
nur nach dem kompletten Laden des Dokumentes
vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen

.getAdjacentText() Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann
Text kann HTML-Tags enthalten, muss aber nicht
DOM nicht geändert

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.getAttributeNode() Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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

.getBoundingClientRect() Referenz auf TextRectangle-Objekt im Element holen

.getClientRects() Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster
Feld mit Index als Integer ab 0
pro Eintrag ein Rectangle

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren
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
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, onmousemove, onclick, ondblclick, onmouseover und onmouseout. Hinweis: einschalten per Methode .setCapture()
.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
.select()	Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode .focus() verwenden)
.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

input hidden Objekt

ist **kein** Control-Element

Ersatz für Cookies: Übertragung von für den User nicht sichtbarer Daten zum Server
gesendet wird der Wert
zu sender Wert durch Programmierer festlegbar und nicht durch User beeinflussbar
Achtung: Dieses Objekt kann missbräuchlich benutzt werden !

Erzeugung in HTML:

<INPUT TYPE=hidden ...> auch als "hidden" kodierbar



Eigenschaften beim Internet Explorer:

ATOMICSELECTION	Selektierbarkeit des Objektes einstellen
.begin	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction siehe Objekt currTimeState und Behavior .style.time2
.canHaveHTML	prüfen ob Objekt HTML-Tags enthalten darf
.className	Klassenreferenz, Klassennamen
.dataFld	Datenquelle-Name vergeben (ID)
.dataSrc	Datenquelle als Anker festlegen
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.end	Objektaktivitäten laut Eigenschaft .timeAction beenden ab IE 6.x alternativ: Eigenschaft .dur siehe Objekt currTimeState und Behavior .style.time2
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.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);
.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
.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)
.offsetParent	Referenz der Eltern für Nutzung von .offsetHeight, .offsetLeft, .offsetTop und .offsetWidth
.outerHTML	Referenz auf Bereich ab inklusive HTML-Start bis hinter -Ende-Tag wirksam mit parse 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
.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
.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
.tagName	Tag-Bezeichner des Objektes
.tagUrn	Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut
.timeContainer	Typ der Timeline des Objektes siehe Objekt currTimeState und Behavior .style.time2
.type	Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)



.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
.value	Wert eines Objekt-Attributes Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar

Methoden beim Internet Explorer:

.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 Element kann selbst Kinder haben Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !
.attachEvent()	Einschalten des Registrieren eines Events durch Eventhandler Hinweis: Abschalten mit Methode .detachEvent() Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in Zufallsfolge , es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)
.clearAttributes()	alle HTML-Attribute eines Objektes entfernen außer ID, STYLE und per Script definierte Attribute Script-erzeugte Attribute nicht entfernbar DOM wird geändert
.cloneNode()	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)
.componentFromPoint()	Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt auch für CSS-Layout onmouseover-Event hat nicht die Pixelgenauigkeit wie die Angaben der Methode .componentFromPoint() 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.
.createTextRange()	Textbereich erzeugen
.detachEvent()	Abschalten des Registrieren eines Events durch Eventhandler wobei Registrierung mit Methode .attachEvent() aktiviert wurde Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichnet zu, das nicht behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)
.fireEvent()	ein Event auslösen
.getAdjacentText()	Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann Text kann HTML-Tags enthalten, muss aber nicht DOM nicht geändert
.getAttribute()	Wert eines per HTML erzeugten Attributes liefern DOM nicht geändert
.getAttributeNode()	Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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
.getExpression()	Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern Style-Eigenschaft ist per Methoden expression() oder setExpression() zu definieren DOM wird nicht verändert (nur Werteveränderung), aber das Dokument-Layout (nach dem eventuellen expliziten Dokument-Refresh)
.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



.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. Hinweis: einschalten per Methode .setCapture()
.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
.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
.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 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

input image Objekt

Image-Control für Senden eines Formulars durch Klick auf das Bild (interaktives Bild **nur** zum Zweck des Submit) gesendet werden **nur** die Koordinaten der linken oberen Ecke (in Pixel) gesendet

wobei es sich darin um interne Eigenschaften handelt, die nicht manipulierbar sind

- .x für Spalte
- .y für Zeile

ab IE 5.0 werden folgende Image-Medien unterstützt:

- .avi Audio-Visual Interleaved (AVI)
- .bmp Windows Bitmap (BMP)
- .emf Windows Enhanced Metafile (EMF)
- .gif Graphics Interchange Format (GIF)
- .jpg, .jpeg Joint Photographic Experts Group (JPEG)



.mov	Apple QuickTime Movie (MOV)
.mpg, .mpeg	Motion Picture Experts Group (MPEG)
.png	Portable Network Graphics (PNG)
.wmf	Windows Metafile (WMF)
.xbm	X Bitmap (XBM)

basiert auf dem img Objekt

Erzeugung in HTML:

<INPUT TYPE=image> auch als "image" kodierbar

Eigenschaften beim Internet Explorer:

.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
.alt	Alternativer Text als Tooltip
ATOMICSELECTION	Selektierbarkeit des Objektes einstellen
.begin	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
.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
.complete	Zustand des Ladens des Objektes
.dataFld	Datenquelle-Name vergeben (ID)
.dataSrc	Datenquelle als Anker festlegen
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.dir	Umflussrichtung
.disabled	Interaktionsfähigkeit nur wenn sichtbar so User-Interaktion möglich
.dynsrc	Adresse von Videoclip oder VRML
.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.hspace	horizontaler Abstand in Pixel zum Elternobjekt
.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);
.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
.loop	Anzahl der Wiederholungen nur Objekt BGSOUND bzw. IMG mit Sound bzw. INPUT-Element mit Sound
.lowsrc	Url des Images in geringerer Auflösung
.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
.outerHTML	Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound Referenz auf Bereich ab inklusive HTML-Start bis hinter -Ende-Tag wirksam mit parse 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
.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 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
.size	Dimension eines Input-Objektes
.sourceIndex	Index des Objektes in der Collection document.all
.src	Url der Daten z.B. vom Image in normaler Auflösung
.start	Start eines Videoclips oder VRML-Datei
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
.timeContainer	Typ der Timeline des Objektes siehe Objekt currTimeState und Behavior .style.time2
.title	Tooltip-Text bei Mouse over über Objekt
.type	Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)



.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
.value	Wert eines Objekt-Attributes Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
.width	Breite des Objektes in Pixel
<u>Methoden beim Internet Explorer:</u>	
.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
.appendChild()	Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern DOM wird geändert Zeiger wird zugleich immer am Ende der Collection childNodes angehängen Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde
.applyElement()	Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern DOM wird geändert Element kann selbst Kinder haben Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !
.attachEvent()	Einschalten des Registrieren eines Events durch Eventhandler Hinweis: Abschalten mit Methode .detachEvent() Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in Zufallsfolge , es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)
.blur()	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
.clearAttributes()	alle HTML-Attribute eines Objektes entfernen außer ID, STYLE und per Script definierte Attribute Script-erzeugte Attribute nicht entfernen DOM wird geändert
.click()	simuliert einen Klick auf das Element und löst onclick-Event aus manipuliert nicht den Focus
.cloneNode()	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)
.componentFromPoint()	Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt auch für CSS-Layout onmouseover-Event hat nicht die Pixelgenauigkeit wie die Angaben der Methode .componentFromPoint() 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.
.contains()	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
.detachEvent()	Abschalten des Registrieren eines Events durch Eventhandler wobei Registrierung mit Methode .attachEvent() aktiviert wurde Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichner zu, das nicht behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)
.dragDrop()	prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element
.fireEvent()	ein Event auslösen
.focus()	Focus setzen und Focus-Event auslösen nur nach dem kompletten Laden des Dokumentes vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen
.getAdjacentText()	Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann Text kann HTML-Tags enthalten, muss aber nicht DOM nicht geändert
.getAttribute()	Wert eines per HTML erzeugten Attributes liefern DOM nicht geändert
.getAttributeNode()	Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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
.getBoundingClientRect()	Referenz auf TextRectangle-Objekt im Element holen
.getClientRects()	Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster



Feld mit Index als Integer ab 0
pro Eintrag ein Rectangle

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdrucks berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren
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
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.

.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

.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



.select()	Objekt muss an sich schon renderbar sein Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode .focus() verwenden)
.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
.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
.swapNode()	DOM wird nicht geändert Positionen von 2 Knoten im DOM tauschen (Zeigertausch) nur sichtbar wenn Endetag geparst DOM wird geändert

input password Objekt

Text-Control aus einer Zeile mit Dummy-Anzeige während der Eingabe der Zeichen durch User

Erzeugung in HTML:

<INPUT TYPE=password> auch als "password" kodierbar

```

Beispiel <SCRIPT>
function Pruefen()
{
    if (ID_InputButton.value == "Otto")
    {
        if (ID_InputPassword.value == "Waalkes")
        {alert("Password korrekt");}
    }
}
</SCRIPT>

```

```

Username <INPUT TYPE=button ID="ID_InputButton" onclick="Pruefen()">
Password <INPUT TYPE=password ID="ID_InputPassword" onclick=Pruefen()">

```

Eigenschaften beim Internet Explorer:

.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
ATOMICSELECTION	Selektierbarkeit des Objektes einstellen
.autocomplete	Status des Autovervollständigung zum Formular
.begin	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
.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.
.dataFld	Datenquelle-Name vergeben (ID)
.dataSrc	Datenquelle als Anker festlegen
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars



.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.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);
.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
.maxLength	Maximale Anzahl der durch User eingebbaren Zeichen in einem Text-Control
.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 parse 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
.previousSibling	Referenz auf das Vorgängerkind
.readOnly	Editierbarkeit eines Text-Controls Editierung bedeutet Focuserhalt ! aktueller Status des Objektes beim Füllen des Objektes mit Daten
.readyState	
.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 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
.size	Dimension eines Input-Objektes



.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
.timeContainer Typ der Timeline des Objektes
siehe Objekt currTimeState und Behavior .style.time2
.title Tooltip-Text bei Mouse over über Objekt
.type Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
.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
.value Wert eines Objekt-Attributes
Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
.vcard_name Werte für vCard für Autocomplete (Autovervollständigung) bei Formular per Text-Control
.width Breite des Objektes in Pixel

Methoden beim Internet Explorer:

.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

.appendChild() Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern
DOM wird geändert
Zeiger wird zugleich immer am Ende der Collection childNodes angehängen
Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde

.applyElement() Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern
DOM wird geändert
Element kann selbst Kinder haben
Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde
Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !

.attachEvent() Einschalten des Registrieren eines Events durch Eventhandler
Hinweis: Abschalten mit Methode .detachEvent()
Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in **Zufallsfolge**, es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)

.blur() 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

.clearAttributes() alle HTML-Attribute eines Objektes entfernen
außer ID, STYLE und per Script definierte Attribute
Script-erzeugte Attribute nicht entfernbar
DOM wird geändert

.click() simuliert einen Klick auf das Element und löst onclick-Event aus
manipuliert nicht den Focus



.cloneNode()	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)
.componentFromPoint()	Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt auch für CSS-Layout onmouseover-Event hat nicht die Pixelgenauigkeit wie die Angaben der Methode .componentFromPoint() 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.
.contains()	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
.createTextRange() .detachEvent()	Textbereich erzeugen Abschalten des Registrieren eines Events durch Eventhandler wobei Registrierung mit Methode .attachEvent() aktiviert wurde Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichner zu, das nicht behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)
.dragDrop() .fireEvent() .focus()	prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element ein Event auslösen Focus setzen und Focus-Event auslösen nur nach dem kompletten Laden des Dokumentes vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen
.getAdjacentText()	Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann Text kann HTML-Tags enthalten, muss aber nicht DOM nicht geändert
.getAttribute()	Wert eines per HTML erzeugten Attributes liefern DOM nicht geändert
.getAttributeNode()	Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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
.getBoundingClientRect() .getClientRects()	Referenz auf TextRectangle-Objekt im Element holen Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster Feld mit Index als Integer ab 0 pro Eintrag ein Rectangle
.getExpression()	Wert einer Style-Eigenschaft anhand des Ausdrucks berechnen und liefern Style-Eigenschaft ist per Methoden expression() oder setExpression() zu definieren 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 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



.releaseCapture()	Achtung: CDATA-Sections dürfen nicht enthalten sein, da diese immer Inkonsistenz erzeugen 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
.scrollIntoView()	DOM wird geändert Objekt derart scrollen, dass es im Fenster für User sichtbar wird Objekt muss an sich schon renderbar sein
.select()	Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode .focus() verwenden)
.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
.swapNode()	Ausdruck nur als Script kodierbar DOM wird nicht geändert Positionen von 2 Knoten im DOM tauschen (Zeigertausch) nur sichtbar wenn Endetag geparkt DOM wird geändert

input radio Objekt

Radio-Button-Control z.B. für Formular
ermöglicht Auswahl (Selektion) genau eines Eintrages aus Menge von Einträgen:

jeder Eintrag ist Radio-Button-Control
kann nur selektiert werden wenn NAME-Attribut kodiert wurde
zu jedem Zeitpunkt kann nur ein Eintrag ausgewählt sein

Menge von Einträgen ist Gruppe:
alle Elemente der Gruppe haben identischen Wert im NAME-Attribut
mehrere Gruppen zulässig mit je eigenem Namen



Standard-Element markierbar per Eigenschaft .checked (siehe dort)

senden beim Formular: nur das ausgewählte (selektierte) Element der Gruppe:
gesendet wird der Wert und NAME-Attribut-Wert

Erzeugung in HTML:

Beispiel

```

<SCRIPT>
  function Anzeigen()
  {
    if (radio[0].checked)
    {alert("Eintrag 1");}
    else
    {
      if (radio[1].checked)
      {alert("Eintrag 2");}
      else
      {alert("Eintrag 3");}
    }
  }
</SCRIPT>
Gruppe <B>GemeinsamerName<B>
<BR>
<INPUT TYPE=radio NAME="GemeinsamerName" CHECKED>Eintrag 1
<INPUT TYPE=radio NAME="GemeinsamerName">Eintrag 2
<INPUT TYPE=radio NAME="GemeinsamerName">Eintrag 3
<INPUT TYPE=button
  VALUE="Anzeige des aktuell selektierten Eintrages"
  onClick="Anzeigen()"
>

```

Eigenschaften beim Internet Explorer:

.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
ATOMICSELECTION	Selektierbarkeit des Objektes einstellen
.begin	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
.checked	Selektionsstatus des Checkbox-Control bzw. Radio Button-Control bezüglich Selektion durch User
.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.
.dataFld	Datenquelle-Name vergeben (ID)
.dataSrc	Datenquelle als Anker festlegen
.defaultChecked	Selektionsstatus des Checkbox-Control bzw. Radio Button-Control bzw. Radio Button-Control bezüglich Standard-Selektion
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit



.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).
.isContentEditable	Zeiger aus ID bilden <code>var Zeiger = eval(object.id);</code> 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
.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
.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 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
.size	Dimension eines Input-Objektes
.sourceIndex	Index des Objektes in der Collection document.all
.status	Selektionsstatus eines Control-Elementes Control-Element: kann durch User interaktiv verändert werden direkt im HTML-Element kodierter Style (Inline-Style) Hinweis: für Scripting ist das Style-Objekt zu nutzen
STYLE	
.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
 .timeContainer Typ der Timeline des Objektes
 siehe Objekt currTimeState und Behavior .style.time2
 .title Tooltip-Text bei Mouse over über Objekt
 .type Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
 .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
 .value Wert eines Objekt-Attributes
 Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
 .width Breite des Objektes in Pixel

Methoden beim Internet Explorer:

.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

.appendChild() Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern
 DOM wird geändert
 Zeiger wird zugleich immer am Ende der Collection childNodes angehängen
 Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde

.applyElement() Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern
 DOM wird geändert
 Element kann selbst Kinder haben
 Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde
 Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !

.attachEvent() Einschalten des Registrieren eines Events durch Eventhandler
 Hinweis: Abschalten mit Methode .detachEvent()
 Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in **Zufallsfolge**, es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)

.blur() 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

.clearAttributes() alle HTML-Attribute eines Objektes entfernen
 außer ID, STYLE und per Script definierte Attribute
 Script-erzeugte Attribute nicht entfernbar
 DOM wird geändert

.click() simuliert einen Klick auf das Element und löst onclick-Event aus
 manipuliert nicht den Focus

.cloneNode() 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)

.componentFromPoint() Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt
 auch für CSS-Layout
 onmouseover-Event hat **nicht die Pixelgenauigkeit** wie die Angaben der Methode .componentFromPoint() 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.

.contains() 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

.detachEvent() Abschalten des Registrieren eines Events durch Eventhandler
 wobei Registrierung mit Methode .attachEvent() aktiviert wurde



Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichner zu, das **nicht** behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)

.dragDrop() prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element

.fireEvent() ein Event auslösen

.focus() Focus setzen und Focus-Event auslösen

.getAdjacentText() nur nach dem kompletten Laden des Dokumentes
vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen
Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann
Text kann HTML-Tags enthalten, muss aber nicht

.getAttribute() DOM nicht geändert
Wert eines per HTML erzeugten Attributes liefern

.getAttributeNode() DOM nicht geändert
Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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

.getBoundingClientRect() DOM nicht geändert
Referenz auf TextRectangle-Objekt im Element holen

.getClientRects() Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster
Feld mit Index als Integer ab 0
pro Eintrag ein Rectangle

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren

.hasChildNodes() DOM wird nicht verändert (nur Werteveränderung), aber das Dokument-Layout
(nach dem eventuellen expliziten Dokument-Refresh)
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 DOM wird geändert
.removeExpression()	Ausdruck entfernen, der für die Berechnung des Wertes einer Style-Eigenschaft als Objektreferenz der Form <code>objekt.style.eigenschaft.dient</code> . Ausdruck muss mit der Methode <code>.setExpression()</code> 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 <code>.createElement()</code> 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
.select()	Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode <code>.focus()</code> verwenden)
.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 : <code>onmousedown</code> , <code>onmouseup</code> , <code>onmousemove</code> , <code>onclick</code> , <code>ondblclick</code> , <code>onmouseover</code> und <code>onmouseout</code> . ab IE 5.5
.setExpression()	Hinweis: ausschalten per Methode <code>.releaseCapture()</code> Wert definieren, der als Ausdruck für die Methode <code>.getExpression()</code> zur Berechnung einer Style-Eigenschaft als Objektreferenz der Form <code>objekt.style.eigenschaft.dient</code> . 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

input reset Objekt

Reset-Button für Formular-Reset: Standardgemäß alle Formular-Elemente auf Initialwert setzen.
VALUE-Attribut für Titel (Label) des Reset-Button verwenden
Wert wird nicht gesendet

Erzeugung in HTML:

<INPUT TYPE=reset> auch als "reset" kodierbar

Eigenschaften beim Internet Explorer:

.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
ATOMICSELECTION	Selektierbarkeit des Objektes einstellen
.begin	Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft <code>.timeAction</code> siehe Objekt <code>currTimeState</code> und Behavior <code>.style.time2</code>
.canHaveChildren	prüfen ob Kind möglich ist, also ob Objekt Parent sein kann
.canHaveHTML	prüfen ob Objekt HTML-Tags enthalten darf
.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.
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.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);
.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
.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 Knotentyp laut attributes Collection
.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
.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 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
.size	Dimension eines Input-Objektes
.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 (Eineindeutigkeit). 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
.timeContainer	Typ der Timeline des Objektes siehe Objekt currTimeState und Behavior .style.time2
.title	Tooltip-Text bei Mouse over über Objekt
.type	Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
.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
.value	Wert eines Objekt-Attributes Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
.width	Breite des Objektes in Pixel
<u>Methoden beim Internet Explorer:</u>	
.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
.appendChild()	Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern DOM wird geändert Zeiger wird zugleich immer am Ende der Collection childNodes angehängen Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde
.applyElement()	Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern DOM wird geändert Element kann selbst Kinder haben Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !
.attachEvent()	Einschalten des Registrieren eines Events durch Eventhandler Hinweis: Abschalten mit Methode .detachEvent() Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider NICHT verkettet sondern in Zufallsfolge , es sei denn die Handler prüfen ihre Aufruffolge (muss programmiert werden)
.blur()	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
.clearAttributes()	alle HTML-Attribute eines Objektes entfernen außer ID, STYLE und per Script definierte Attribute Script-erzeugte Attribute nicht entfernbar DOM wird geändert



.click() simuliert einen Klick auf das Element und löst onclick-Event aus
manipuliert nicht den Focus

.cloneNode() 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)

.componentFromPoint() Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt
auch für CSS-Layout
onmouseover-Event hat **nicht die Pixelgenauigkeit** wie die Angaben der Methode .componentFromPoint()
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.

.contains() 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

.createTextRange() Textbereich erzeugen

.detachEvent() Abschalten des Registrieren eines Events durch Eventhandler
wobei Registrierung mit Methode .attachEvent() aktiviert wurde
Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichner zu, das **nicht** behandelt werden soll, falls es für das Window-Objekt auftritt (also Standardbehandlung aktiv)

.dragDrop() prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element

.fireEvent() ein Event auslösen

.focus() Focus setzen und Focus-Event auslösen
nur nach dem kompletten Laden des Dokumentes
vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen

.getAdjacentText() Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann
Text kann HTML-Tags enthalten, muss aber nicht
DOM nicht geändert

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.getAttributeNode() Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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

.getBoundingClientRect() Referenz auf TextRectangle-Objekt im Element holen

.getClientRects() Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster
Feld mit Index als Integer ab 0
pro Eintrag ein Rectangle

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren
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
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 !!



.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. Hinweis: einschalten per Methode .setCapture()
.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
.select()	Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode .focus() verwenden)
.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 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

input submit Objekt

Submit-Button für Formular

Label im VALUE-Attribut kodieren (Standardtext ist "Submit Query")

Senden nur möglich, wenn NAME-Attribut kodiert wurde:

gesendet wird Wert von NAME- und VALUE-Attribut

Erzeugung in HTML:

<INPUT TYPE=submit> auch als "submit" kodierbar



Eigenschaften beim Internet Explorer:

.accessKey	Tastaturzugriff auf ein Objekt per Alt + Taste bei Ausführung der Tastenkombination wird das Sprungziel wird focussiert die Sprungquelle defocussiert das Focus-Ereignis ausgelöst
ATOMICSELECTION	vor IE 5.x funktioniert Tastenkombination nicht mit jedem Objekt Selektierbarkeit des Objektes einstellen
.begin	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
.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.
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.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);
.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
.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 Knotentyp laut attributes Collection
.nodeType	Knotenwert (Wert des Kindes, Node, Elementes) nur für Text- und Attribut-Elemente nicht für Element-Knoten (Knotentyp 1)
.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
.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
	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
.size	Dimension eines Input-Objektes
.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
.timeContainer	Typ der Timeline des Objektes
	siehe Objekt currTimeState und Behavior .style.time2
.title	Tooltip-Text bei Mouse over über Objekt
.type	Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
.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
.value	Wert eines Objekt-Attributes
	Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
.width	Breite des Objektes in Pixel
<u>Methoden beim Internet Explorer:</u>	
.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
.appendChild()	Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern
	DOM wird geändert
	Zeiger wird zugleich immer am Ende der Collection childNodes angehängen
	Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag (falls vorhanden) des Knoten geparkt wurde
.applyElement()	Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern
	DOM wird geändert
	Element kann selbst Kinder haben



Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde
Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im
im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !

.attachEvent() Einschalten des Registrieren eines Events durch Eventhandler
Hinweis: Abschalten mit Methode .detachEvent()
Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider
NICHT verkettet sondern in **Zufallsfolge**, es sei denn
die Handler prüfen ihre Aufruffolge (muss programmiert werden)

.blur() 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

.clearAttributes() alle HTML-Attribute eines Objektes entfernen
außer ID, STYLE und per Script definierte Attribute
Script-erzeugte Attribute nicht entferbar
DOM wird geändert

.click() simuliert einen Klick auf das Element und löst onclick-Event aus
manipuliert nicht den Focus

.cloneNode() 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)

.componentFromPoint() Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt
auch für CSS-Layout
onmouseover-Event hat **nicht die Pixelgenauigkeit** wie die Angaben der Methode .componentFromPoint()
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.

.contains() 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

.createTextRange() Textbereich erzeugen

.detachEvent() Abschalten des Registrieren eines Events durch Eventhandler
wobei Registrierung mit Methode .attachEvent() aktiviert wurde
Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_ bezeichner
zu, das **nicht** behandelt werden soll, falls es für das Window-Objekt auftritt
(also Standardbehandlung aktiv)

.dragDrop() prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element

.fireEvent() ein Event auslösen

.focus() Focus setzen und Focus-Event auslösen
nur nach dem kompletten Laden des Dokumentes
vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen

.getAdjacentText() Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann
Text kann HTML-Tags enthalten, muss aber nicht
DOM nicht geändert

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.getAttributeNode() Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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

.getBoundingClientRect() Referenz auf TextRectangle-Objekt im Element holen

.getClientRects() Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster
Feld mit Index als Integer ab 0
pro Eintrag ein Rectangle

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren
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
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, onmousemove, onclick, ondblclick,
 onmouseover und onmouseout.
 Hinweis: einschalten per Methode .setCapture()

.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

.select()
 Bereich des Input-Objektes im Formular markieren
 setzt **nicht** den Focus (dafür Methode .focus() verwenden)

.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
 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



.swapNode() Ausdruck nur als Script kodierbar
 DOM wird nicht geändert
 Positionen von 2 Knoten im DOM tauschen (Zeigertausch)
 nur sichtbar wenn Endetag geparkt
 DOM wird geändert

input text Objekt

Einzeilige Textbox, in der User eingeben kann
 Breite der Textbox in Zeichen im SIZE-Attribut kodieren.
 Maximale Anzahl der eingebbaren Zeichen in Attribut MAXLENGTHs kodieren.
 Wenn SIZE < MAXLENGTH so wird automatisch horizontaler Scrollbalken erzeugt.

Erzeugung in HTML:

Beispiel

```
<SCRIPT>
function Anzeige()
{alert("Es wurde eingegeben" + textbox.value); }
</SCRIPT>

<INPUT TYPE=text VALUE="" NAME="textbox" SIZE=15 onclick="Anzeige()">
```

Eigenschaften beim Internet Explorer:

.accessKey	Tastaturzugriff auf ein Objekt per Alt + Taste bei Ausführung der Tastenkombination wird das Sprungziel wird focussiert die Sprungquelle defocussiert das Focus-Ereignis ausgelöst
ATOMICSELECTION	vor IE 5.x funktioniert Tastenkombination nicht mit jedem Objekt Selektierbarkeit des Objektes einstellen
.autocomplete	Status des Autovervollständigung zum Formular
.begin	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
.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.
.dataFld	Datenquelle-Name vergeben (ID)
.dataSrc	Datenquelle als Anker festlegen
.defaultValue	Standardwert laut Objektinitialisierung für Element innerhalb eines Formulars
.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
.form	Zeiger auf das Formular (Formular als Container) ab IE 6.x für Elemente fieldSet, label, legend
.hasMedia	Objekt ist HTML-Media-Objekt siehe Objekt currTimeState und Behavior .style.time2
.hideFocus	Focussierbarkeit
.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);
.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
.maxLength	Maximale Anzahl der durch User eingebbaren Zeichen in einem Text-Control
.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	deprectated 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
.previousSibling	Referenz auf das Vorgängerkind
.readOnly	Editierbarkeit eines Text-Controls Editierung bedeutet Focuserhalt ! aktueller Status des Objektes beim Füllen des Objektes mit Daten
.readyState	
.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 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
.size	Dimension eines Input-Objektes
.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
 .timeContainer Typ der Timeline des Objektes
 siehe Objekt currTimeState und Behavior .style.time2
 .title Tooltip-Text bei Mouse over über Objekt
 .type Typ (Variante) des Input-Objektes (Art der Control-Elemente vom gemeinsamen Typ INPUT)
 .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
 .value Wert eines Objekt-Attributes
 Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar
 .vcard_name Werte für vCard für Autocomplete (Autovervollständigung) bei Formular per Text-Control
 .width Breite des Objektes in Pixel

Methoden beim Internet Explorer:

.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

.appendChild() Knoten als Kind an die DOM-Hierarchie anhängen und danach den Zeiger laut DOM liefern
 DOM wird geändert
 Zeiger wird zugleich immer am Ende der Collection childNodes angehängen
 Anhängen wird danach nur sichtbar, wenn zusätzlich dem BODY-Objekt angehängen wird UND Ende-Tag
 (falls vorhanden) des Knoten geparkt wurde

.applyElement() Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz
 laut DOM liefern
 DOM wird geändert
 Element kann selbst Kinder haben
 Element erst sichtbar, wenn Endetag (falls vorhanden) des Elementes geparkt wurde
 Achtung: Wenn Element per Methode .createElement() erzeugt wurde, aber nicht im
 im Dokumentenbaum eingebunden ist, so wird die Eigenschaft .innerHTML gelöscht !

.attachEvent() Einschalten des Registrieren eines Events durch Eventhandler
 Hinweis: Abschalten mit Methode .detachEvent()
 Achtung: Wenn mehrere Eventhandler zum Event, so Aufruf der Handler leider
 NICHT verkettet sondern in **Zufallsfolge**, es sei denn
 die Handler prüfen ihre Aufruffolge (muss programmiert werden)

.blur() 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

.clearAttributes() alle HTML-Attribute eines Objektes entfernen
 außer ID, STYLE und per Script definierte Attribute
 Script-erzeugte Attribute nicht entfernbar
 DOM wird geändert

.click() simuliert einen Klick auf das Element und löst onclick-Event aus
 manipuliert nicht den Focus

.cloneNode() 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)

.componentFromPoint() Layout-Komponente eines Objektes ermitteln, die an einer Koordinate liegt
 auch für CSS-Layout
 onmouseover-Event hat **nicht die Pixelgenauigkeit** wie die Angaben der Methode .componentFromPoint()
 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.

.contains() 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

.createTextRange() Textbereich erzeugen

.detachEvent() Abschalten des Registrieren eines Events durch Eventhandler
 wobei Registrierung mit Methode .attachEvent() aktiviert wurde
 Abschalten = ordnet dem Window-Objekt das Event laut Parameter event_bezeichner
 zu, das **nicht** behandelt werden soll, falls es für das Window-Objekt auftritt
 (also Standardbehandlung aktiv)

.dragDrop() prüfen des Status der letzten Drag-Manipulation (anklicken, ziehen, ablegen) auf Element

.fireEvent() ein Event auslösen

.focus() Focus setzen und Focus-Event auslösen
 nur nach dem kompletten Laden des Dokumentes
 vor IE 5.x: Objekt muss TABINDEX-Attribut besitzen

.getAdjacentText() Text eines Objektes liefern, wobei Textlage im Objekt definiert werden kann



Text kann HTML-Tags enthalten, muss aber nicht
DOM nicht geändert

.getAttribute() Wert eines per HTML erzeugten Attributes liefern
DOM nicht geändert

.getAttributeNode() Referenz auf Eigenschaft des attribute-Objektes liefern, also Zeiger auf attribute.name 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

.getBoundingClientRect() Referenz auf TextRectangle-Objekt im Element holen

.getClientRects() Referenz auf Feld der Zeiger auf TextRectangle-Objekte im Fenster
Feld mit Index als Integer ab 0
pro Eintrag ein Rectangle

.getExpression() Wert einer Style-Eigenschaft anhand des Ausdruckes berechnen und liefern
Style-Eigenschaft ist per Methoden
expression() oder setExpression()
zu definieren

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
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, 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

.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
.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
.scrollIntoView()	DOM wird geändert Objekt derart scrollen, dass es im Fenster für User sichtbar wird Objekt muss an sich schon renderbar sein
.select()	Bereich des Input-Objektes im Formular markieren setzt nicht den Focus (dafür Methode .focus() verwenden)
.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 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
.swapNode()	DOM wird nicht geändert Positionen von 2 Knoten im DOM tauschen (Zeigertausch) nur sichtbar wenn Endetag geparkt DOM wird geändert



#default#behaviorName 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
#text 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
.accept 2
.accessKey 2, 7, 11, 19, 23, 28, 32, 37, 41
.addBehavior() 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
.align 2, 19
.alt 2, 19
.appendChild() 4, 9, 13, 21, 25, 30, 34, 38, 43
.applyElement() 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
.attachEvent() 4, 9, 13, 14, 17, 21, 25, 26, 30, 34, 35, 39, 43
.autocomplete 23, 41
.avi 18
.begin 2, 7, 11, 16, 19, 23, 28, 32, 37, 41
.blur() 4, 9, 13, 21, 25, 30, 34, 39, 43
.bmp 18
.canHaveChildren 2, 7, 11, 19, 23, 28, 32, 37, 41
.checked 7, 28
.className 2, 7, 11, 16, 19, 23, 28, 32, 37, 41
.clearAttributes() 4, 9, 13, 17, 21, 25, 30, 34, 39, 43
.click() 4, 9, 14, 21, 25, 30, 35, 39, 43
.clientHeight 2, 7, 11, 19, 23, 28, 32, 37, 41
.clientLeft 2, 7, 12, 19, 23, 28, 32, 37, 41
.clientTop 2, 7, 12, 19, 23, 28, 32, 37, 41
.clientWidth 2, 7, 12, 19, 23, 28, 32, 37, 41
.cloneNode() 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
.complete 2, 19
.componentFromPoint() 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
.contains() 4, 9, 14, 21, 26, 30, 35, 39, 43
.contentEditable 2, 23, 28, 33, 37, 41
.createElement() 6, 10, 15, 18, 22, 27, 32, 36, 40, 45
.dataFld 2, 7, 16, 19, 23, 28, 41
.dataFormatAs 2
.dataSrc 2, 7, 16, 19, 23, 28, 41
.defaultChecked 7, 28
.defaultValue 2, 7, 12, 16, 19, 23, 28, 33, 37, 41
.detachEvent() 4, 9, 13, 14, 17, 21, 25, 26, 30, 34, 35, 39, 43
.dir 2, 7, 12, 19, 24, 28, 33, 37, 41
.disabled 2, 7, 12, 19, 24, 28, 33, 37, 41
.dragDrop() 4, 9, 14, 21, 26, 31, 35, 39, 43
.dynsrc 2, 19
.emf 18
.end 2, 7, 12, 16, 19, 24, 28, 33, 37, 41
.fireEvent() 4, 9, 14, 17, 21, 26, 31, 35, 39, 43
.firstChild 2, 7, 12, 19, 24, 28, 33, 37, 41
.focus() 4, 6, 9, 10, 14, 15, 21, 23, 26, 27, 31, 32, 35, 36, 39, 40, 43, 45
.form 2, 7, 12, 16, 19, 24, 28, 33, 37, 41
.getAdjacentText() 5, 9, 14, 17, 21, 26, 31, 35, 39, 43
.getAttribute() 5, 9, 14, 17, 21, 26, 31, 35, 39, 44
.getAttributeNode() 5, 9, 14, 17, 21, 26, 31, 35, 39, 44
.getBoundingClientRect() 5, 9, 14, 21, 26, 31, 35, 39, 44
.getClientRects() 5, 9, 14, 21, 26, 31, 35, 39, 44
.getExpression() 5, 9, 14, 17, 22, 26, 31, 35, 39, 44
.gif 18
.hasChildNodes() 5, 10, 14, 22, 26, 31, 35, 39, 44
.hasMedia 2, 7, 12, 16, 19, 24, 28, 33, 37, 41
.hideFocus 2, 7, 12, 16, 19, 24, 28, 33, 37, 41
.hspace 2, 19
.id 2, 7, 12, 16, 19, 24, 29, 33, 37, 41
.indeterminate 7
.insertAdjacentElement() 5, 10, 14, 17, 22, 26, 31, 35, 39, 44
.insertAdjacentHTML() 5, 10, 14, 17, 22, 26, 31, 35, 39, 44
.insertAdjacentText() 5, 10, 14, 17, 22, 26, 31, 35, 40, 44
.insertBefore() 5, 10, 14, 22, 26, 31, 35, 40, 44
.isContentEditable 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
.isDisabled 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
.isMultiLine 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
.isTextEdit 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
.jpeg 18
.jpg 18
.lang 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
.language 3, 7, 12, 16, 19, 24, 29, 33, 37, 42
.lastChild 3, 7, 12, 19, 24, 29, 33, 37, 42

.loop 19
.lowsrc 2, 19
.maxLength 24, 42
.mergeAttributes() 5, 10, 14, 18, 22, 26, 31, 35, 40, 44
.mov 19
.mpeg 19
.mpg 19
.name 3, 7, 12, 16, 19, 24, 29, 33, 37, 42
.nextSibling 3, 7, 12, 16, 19, 24, 29, 33, 37, 42
.nodeName 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
.nodeType 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
.nodeValue 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
.normalize() 5, 10, 15, 18, 22, 26, 31, 36, 40, 44
.offsetParent 3, 8, 12, 20, 24, 29, 33, 37, 42
.offsetLeft 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
.offsetHeight 3, 7, 12, 20, 24, 29, 33, 37, 42
.offsetHeight 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
.offsetLeft 3, 7, 12, 20, 24, 29, 33, 37, 42
.offsetParent 3, 7, 8, 12, 16, 20, 24, 29, 33, 37, 42
.offsetTop 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
.offsetWidth 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
.onOffBehavior 3, 8, 12, 20, 24, 29, 33, 37, 42
.outerHTML 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
.outerText 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
.ownerDocument 3, 8, 12, 20, 24, 29, 33, 38, 42
.parentElement 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
.parentNode 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
.parentTextEdit 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
.png 19
.previousSibling 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
.readOnly 24, 42
.readyState 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
.recordNumber 8, 16, 24, 29, 42
.releaseCapture() 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
.removeAttribute() 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
.removeAttributeNode() 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
.removeBehavior() 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
.removeChild() 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
.removeExpression() 5, 10, 15, 18, 22, 27, 32, 36, 40, 44
.removeNode() 6, 10, 15, 22, 27, 32, 36, 40, 44
.replaceAdjacentText() 6, 10, 15, 18, 22, 27, 32, 36, 40, 45
.replaceChild() 6, 10, 15, 18, 22, 27, 32, 36, 40, 45
.replaceNode() 6, 10, 15, 22, 27, 32, 36, 40, 45
.scopeName 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
.scrollHeight 3, 8, 12, 20, 24, 29, 33, 38, 42
.scrollIntoView() 6, 10, 15, 22, 27, 32, 36, 40, 45
.scrollLeft 3, 8, 12, 20, 24, 29, 33, 38, 42
.scrollTop 3, 8, 12, 20, 24, 29, 33, 38, 42
.scrollWidth 3, 8, 13, 20, 24, 29, 34, 38, 42
.select() 6, 10, 15, 23, 27, 32, 36, 40, 45
.setActive() 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
.setAttribute() 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
.setAttributeNode() 6, 11, 15, 18, 23, 27, 32, 36, 40, 45
.setCapture() 6, 11, 15, 18, 23, 27, 32, 36, 40, 45
.setExpression() 6, 11, 15, 18, 23, 27, 32, 36, 40, 45
.size 3, 8, 13, 20, 24, 29, 34, 38, 42
.sourceIndex 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.src 20
.start 2, 20
.status 8, 29
.style.time2 Behavior .. 2, 3, 4, 7, 8, 11, 12, 13, 16, 19, 20, 23, 24, 25, 28, 29, 30, 32, 33, 34, 37, 38, 41, 42, 43
.swapNode() 6, 11, 15, 18, 23, 27, 32, 36, 41, 45
.syncBehavior 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.syncMaster 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.syncTolerance 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.systemBitrate 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.systemCaptions 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.systemLanguage 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.systemOverdubOrSubtitle 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
.tabIndex 3, 8, 13, 20, 25, 30, 34, 38, 42
.tagName 4, 8, 13, 16, 20, 25, 30, 34, 38, 43
.tagUrn 4, 8, 13, 16, 20, 25, 30, 34, 38, 43



.timeContainer 4, 8, 13, 16, 20, 25, 30, 34, 38, 43
.title 4, 8, 13, 20, 25, 30, 34, 38, 43
.uniqueID .. 2, 4, 7, 8, 12, 13, 16, 17, 19, 21, 24, 25, 29, 30, 33, 34, 37, 38, 41, 43
.useMap 2
.value 2, 4, 8, 13, 17, 21, 25, 30, 34, 38, 43
.vcard_name 25, 43
.vspace 2
.width 4, 8, 13, 21, 25, 30, 34, 38, 43
.wmf 19
.xbm 19
Abstand Objekt zum linken Rand des Fensters 2, 7, 12, 19, 23, 28, 32, 37, 41
Abstand Objekt zum oberen Rand des Fensters.... 2, 7, 12, 19, 23, 28, 33, 37, 41
ACTION 11
Adresse von Videoclip 2, 19
aktivieren Eventdurchreichung.... 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Alt + Taste 2, 7, 11, 19, 23, 28, 32, 37, 41
alternativer Text als Tooltip 2, 19
anhängen Kind 4, 9, 13, 21, 25, 30, 34, 38, 43
anhängen Knoten 4, 9, 13, 21, 25, 30, 34, 38, 43
Anker 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
Anzeigesprache..... 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
Apple QuickTime Movie..... 19
ASP-Script 11
ATOMICSELECTION..... 2, 7, 11, 16, 19, 23, 28, 32, 37, 41
Attribut automatisch erzeugen und mit Wert belegen 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Attribut eines Knoten 6, 11, 15, 18, 23, 27, 32, 36, 40, 45
Attribut erzeugen und mit Wert belegen . 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Attribut Wert5, 6, 9, 10, 14, 15, 17, 18, 21, 23, 26, 27, 31, 32, 35, 36, 39, 40, 44, 45
Attribute eines Elementes..... 5, 10, 14, 18, 22, 26, 31, 35, 40, 44
Attribute HTML 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
attribute Objekt..... 5, 9, 14, 17, 21, 26, 31, 35, 39, 44
attributes Collection..... 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
Attribut-Name..... 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
Attributwert Objekt 2, 4, 13, 17, 21, 25, 30, 34, 38, 43
Audio-Visual Interleaved 18
auslösen Event 4, 9, 14, 17, 21, 26, 31, 35, 39, 43
Ausrichtung 19
Autocomplete..... 25, 43
automatisch-genriertes ID des Objektes.... 4, 8, 13, 17, 21, 25, 30, 34, 38, 43
Autovervollständigung 25, 43
Autovervollständigung zum Formular 23, 41
AVI 18
Behavior .style.time2... 2, 3, 4, 7, 8, 11, 12, 13, 16, 19, 20, 23, 24, 25, 28, 29, 30, 32, 33, 34, 37, 38, 41, 42, 43
Bereich ab inklusive HTML-Start bis hinter -Ende-Tag .. 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
Bereich des Input-Objektes im Formular markieren. 6, 10, 15, 23, 27, 32, 36, 40, 45
Bezeichner des Objektes..... 2, 7, 12, 16, 19, 24, 29, 33, 37, 41
Bezeichner des Tag eines Objektes 4, 8, 13, 16, 20, 25, 30, 34, 38, 43
BMP 18
Body Eltern 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Body Referenz auf das Elternobjekt.... 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Breite des horizontalen Scrollbereiches3, 8, 13, 20, 24, 29, 34, 38, 42
Breite Objekt..... 2, 7, 12, 19, 23, 28, 33, 37, 41
button 1
button Objekt 1, 2
Button Submit für Formular 36
Button-Control..... 2
canHaveHTML..... 2, 7, 11, 16, 19, 23, 28, 32, 37, 41
CGI-Script..... 11
checkbox 1
Checkbox-Control 6
Checkbox-Control Grauzustand (Dimmed) und Selektiertheit..... 7
Checkbox-Control Selektionsstatus 7, 28
CHECKED 28

childNodes Collection..... 2, 7, 12, 19, 24, 28, 33, 37, 41
Collection attributes 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
Collection childNodes..... 2, 7, 12, 19, 24, 28, 33, 37, 41
Collection document.all 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
Container 4, 9, 14, 21, 26, 30, 35, 39, 43
Content vom Objekt Editierbarkeit 2, 23, 28, 33, 37, 41
Control-Element 4, 8, 13, 15, 16, 20, 25, 30, 34, 38, 43
Control-Element Selektionsstatus 8, 29
Cookie-Ersatz durch input hidden Objekt 15
Cookies..... 15
createAttribute() 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
createTextRange() 4, 17, 26, 35, 39, 43
currTimeState Objekt .. 2, 3, 4, 7, 8, 11, 12, 13, 16, 19, 20, 23, 24, 25, 28, 29, 30, 32, 33, 34, 37, 38, 41, 42, 43
Cursorgeschwindigkeit..... 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
Daten Url 20
Datenfeld 8, 16, 24, 29, 42
Datenquelle als Anker festlegen 2, 7, 16, 19, 23, 28, 41
Datenquelle-Anzeigart..... 2
Datenquelle-Name vergeben..... 2, 7, 16, 19, 23, 28, 41
Datenquelle-Satznummer 8, 16, 24, 29, 42
Datenübertragung 15
DEFER 5, 10, 14, 17, 22, 26, 31, 35, 40, 44
DHTML-Eigenschaft hinzufügen .. 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
document Objekt des Knoten 3, 8, 12, 20, 24, 29, 33, 38, 42
document.all Collection 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
document.form.input Objekt 1
DOM 6, 10, 15, 22, 27, 32, 36, 40, 44
DOM Elementeigenschaft..... 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
DOM Normalisierung 5, 10, 15, 18, 22, 26, 31, 36, 40, 44
Drag-Manipulation Status 4, 9, 14, 21, 26, 31, 35, 39, 43
durchreichen Event..... 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Editierbarkeit Content vom Objekt 2, 23, 28, 33, 37, 41
Editierbarkeit des Objekt-Content . 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
Eigenschaft des attribute-Objektes 5, 9, 14, 17, 21, 26, 31, 35, 39, 44
Eigenschaft Element..... 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
Eigenschaft hinzufügen..... 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
Eigenschaften IE Standard..... 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
einzeilige Textbox 41
Element Attribute 5, 10, 14, 18, 22, 26, 31, 35, 40, 44
Element den Focus wegnehmen..... 4, 9, 13, 21, 25, 30, 34, 39, 43
Element Eigenschaft..... 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
Element eines Formulars..... 2, 7, 12, 16, 19, 23, 28, 33, 37, 41
Element in einem Formular..... 2, 7, 12, 16, 19, 24, 28, 33, 37, 41
Element innerhalb eines Elementes 4, 9, 14, 21, 26, 30, 35, 39, 43
Elementeigenschaft im DOM..... 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
Elementes Tab-Tasten-Folge 3, 8, 13, 20, 25, 30, 34, 38, 42
Eltern 4, 9, 13, 17, 21, 25, 30, 34, 38, 43
Eltern Body 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Eltern Referenz..... 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
Elternknoten 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Elternobjekt Textbereich..... 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
EMF 18
ENCTYPE..... 11
Ereignis Standardbehandlung 4, 9, 14, 17, 21, 26, 31, 35, 39, 43
Ersatz von Cookies durch input hidden Objekt 15
erstes Kind Referenz 2, 7, 12, 19, 24, 28, 33, 37, 41
erstes Kind Zeiger 2, 7, 12, 19, 24, 28, 33, 37, 41
erzeugen Attribut automatisch und mit Wert belegen 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
erzeugen Attribut und mit Wert belegen. 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Event auslösen..... 4, 9, 14, 17, 21, 26, 31, 35, 39, 43
Event durchreichen..... 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Event onblur 4, 9, 13, 21, 25, 30, 34, 39, 43
Event onfocus 4, 9, 14, 21, 26, 31, 35, 39, 43
Event onmouseover Pixelgenauigkeit. 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
Event registrieren 4, 9, 13, 17, 21, 25, 30, 34, 39, 43
Eventdurchreichung aktivieren 6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Events registrieren..... 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
Existenz Kind möglich..... 2, 7, 11, 19, 23, 28, 32, 37, 41
Existenz von Kinder 5, 10, 14, 22, 26, 31, 35, 39, 44



Feld der Zeiger auf TextRectangle-Objekte.... 5, 9, 14, 21, 26, 31, 35, 39, 44

Fenster linker Rand.....2, 7, 12, 19, 23, 28, 32, 37, 41

Fenster oberer Rand.....2, 7, 12, 19, 23, 28, 33, 37, 41

file 1

File Upload-Control..... 11

Focus.....4, 9, 14, 21, 25, 30, 35, 39, 43

Focus input Objekt6, 10, 15, 23, 27, 32, 36, 40, 45

Focus setzen.....4, 9, 14, 21, 26, 31, 35, 39, 43

Focus wegnehmen4, 9, 13, 21, 25, 30, 34, 39, 43

Focus-Event.....4, 9, 14, 21, 26, 31, 35, 39, 43

Focussierbarkeit Objekt.....2, 7, 12, 16, 19, 24, 28, 33, 37, 41

form Objekt.....2, 7, 12, 16, 19, 23, 28, 33, 37, 41

Formular2, 7, 11, 12, 16, 19, 23, 27, 28, 33, 37, 41

Formular Autovervollständigung.....23, 25, 41, 43

Formular Button Submit.....36

Formular Eingabe41

Formular Passwort.....23

Formular Reset-Button32

Formular senden18

Formular Submit-Button36

Formular Usereingabe41

Formular Zeiger auf Element2, 7, 12, 16, 19, 24, 28, 33, 37, 41

GIF18

Graphics Interchange Format18

Grauzustand (Dimmed) und Selektiertheit des Checkbox-Control ... 7

hidden1

Hintergrundmusik Wiederholungen.....19

hinzufügen DHTML-Eigenschaft ..4, 9, 13, 17, 21, 25, 30, 34, 38, 43

hinzufügen Eigenschaft4, 9, 13, 17, 21, 25, 30, 34, 38, 43

Höhe des vertikalen Scrollbereiches3, 8, 12, 20, 24, 29, 33, 38, 42

Höhe Objekt.....2, 7, 11, 19, 23, 28, 32, 37, 41

horizontaler Scrollbereich3, 8, 13, 20, 24, 29, 34, 38, 42

HTML-Attribut Wert.....5, 9, 14, 17, 21, 26, 31, 35, 39, 44

HTML-Attribute5, 10, 15, 18, 22, 27, 31, 36, 40, 44

HTML-Attribute eines Objektes4, 9, 13, 17, 21, 25, 30, 34, 39, 43

HTML-Code und/oder Script-Code einfügen5, 10, 14, 17, 22, 26, 31, 35, 39, 44

HTML-Tags im Objekt.....2, 7, 11, 16, 19, 23, 28, 32, 37, 41

ID2, 3, 7, 12, 16, 19, 23, 24, 28, 29, 33, 37, 41, 42

ID des Objektes4, 8, 13, 17, 21, 25, 30, 34, 38, 43

ID internes2, 7, 12, 16, 19, 24, 29, 33, 37, 41

ID Objekt2, 7, 12, 16, 19, 24, 29, 33, 37, 41

IE Standard-Eigenschaften4, 9, 13, 17, 21, 25, 30, 34, 38, 43

image.....1

Image in geringerer Auflösung Url.....19

Image Tooltip2, 19

Image-Control.....18

Image-Map client-seitig.....2

Image-Medien.....18

IMG mit Sound.....19

img Objekt1, 19

Index des Elementes in der Tab-Tasten-Folge 3, 8, 13, 20, 25, 30, 34, 38, 42

Index des Objektes in der Collection document.all3, 8, 13, 16, 20, 25, 29, 34, 38, 42

Inkonsistenz.....5, 10, 15, 18, 22, 27, 31, 36, 40, 44

Inline-Style3, 8, 13, 16, 20, 25, 29, 34, 38, 42

INPUT.....4, 8, 13, 16, 20, 25, 30, 34, 38, 43

input button Objekt.....2

input checkbox Objekt.....6

input checkbox Objekt Grauzustand (Dimmed) und Selektiertheit... 7

input checkbox Objekt Selektionsstatus.....7, 28

input checkbox Objekt Style6

input file Objekt.....11

input hidden Objekt15

input image Objekt18, 38

input Objekt1

input Objekt Focus6, 10, 15, 23, 27, 32, 36, 40, 45

input XE "Objekt input im Formular markieren" XE "Objekt input markieren" XE "input Objekt markieren" Objekt im Formular markieren6, 10, 15, 23, 27, 32, 36, 40, 45

input Objekt markieren.....6, 10, 15, 23, 27, 32, 36, 40, 45

input Objekt Typ.....4, 8, 13, 16, 20, 25, 30, 34, 38, 43

input Objekt Varinate.....4, 8, 13, 16, 20, 25, 30, 34, 38, 43

input password Objekt.....23

input radio Objekt27

input radio Objekt Selektionsstatus7, 28

input reset Objekt32

input submit Objekt.....36

input text Objekt.....41

INPUT-Element mit Sound.....19

INPUT-Varianten1

Interaktionsfähigkeit Objekt 2, 3, 7, 12, 16, 19, 24, 28, 29, 33, 37, 41

internes ID2, 7, 12, 16, 19, 24, 29, 33, 37, 41

Joint Photographic Experts Group18

JPEG18

JPG18

Kind4, 9, 13, 17, 21, 25, 30, 34, 38, 43

Kind anhängen.....4, 9, 13, 21, 25, 30, 34, 38, 43

Kind erstes2, 7, 12, 19, 24, 28, 33, 37, 41

Kind letztes.....3, 7, 12, 19, 24, 29, 33, 37, 42

Kind nachfolgendes.....3, 7, 12, 16, 19, 24, 29, 33, 37, 42

Kind Name3, 7, 12, 16, 20, 24, 29, 33, 37, 42

Kind Objekt5, 10, 15, 18, 22, 27, 31, 36, 40, 44

Kind Vorgänger.....3, 8, 12, 16, 20, 24, 29, 33, 38, 42

Kinder Existenz5, 10, 14, 22, 26, 31, 35, 39, 44

Kind-Existenz möglich2, 7, 11, 19, 23, 28, 32, 37, 41

Kindknoten5, 10, 14, 22, 26, 31, 35, 40, 44

Klassenname.....2, 7, 11, 16, 19, 23, 28, 32, 37, 41

Klassenreferenz2, 7, 11, 16, 19, 23, 28, 32, 37, 41

Klick auf das Element4, 9, 14, 21, 25, 30, 35, 39, 43

klonen Objekt4, 9, 14, 17, 21, 26, 30, 35, 39, 43

Knoten als Kind anhängen4, 9, 13, 21, 25, 30, 34, 38, 43

Knoten Attribut6, 11, 15, 18, 23, 27, 32, 36, 40, 45

Knoten Eltern3, 8, 12, 16, 20, 24, 29, 33, 38, 42

Knoten Elternobjekt3, 8, 12, 20, 24, 29, 33, 38, 42

Knoten entfernen.....6, 10, 15, 22, 27, 32, 36, 40, 44

Knoten Existenz5, 10, 14, 22, 26, 31, 35, 39, 44

Knoten im DOM tauschen6, 11, 15, 18, 23, 27, 32, 36, 41, 45

Knoten Kind5, 10, 14, 22, 26, 31, 35, 40, 44

Knotentyp3, 7, 12, 16, 20, 24, 29, 33, 37, 42

Knotenwert3, 7, 12, 16, 20, 24, 29, 33, 37, 42

konsistente Struktur.....5, 10, 15, 18, 22, 26, 31, 36, 40, 44

-Koordinate der linken oberen Ecke Objektes 3, 7, 12, 20, 24, 29, 33, 37, 42

Label36

Label des Reset-Button32

Ladezustand Objekt.....2, 19

Länge Textbox24, 42

Länge Text-Control.....24, 42

Länge Textfeld24, 42

Layout.....2, 3, 7, 12, 16, 19, 23, 24, 28, 29, 33, 37, 41

Layout-Komponente eines Objektes... 4, 9, 14, 17, 21, 26, 30, 35, 39, 43

letztes Kind Zeiger3, 7, 12, 19, 24, 29, 33, 37, 42

linke obere Ecke des Objektes3, 8, 12, 20, 24, 29, 33, 37, 42

linke obere Ecke Objektes.....3, 7, 12, 20, 24, 29, 33, 37, 42

linker Rand des Fensters2, 7, 12, 19, 23, 28, 32, 37, 41

linker Rand des Objektes3, 8, 12, 20, 24, 29, 33, 38, 42

logischer Objektname.....2, 7, 12, 16, 19, 24, 29, 33, 37, 41

Maus-Überwachung ausschalten für ein Objekt..5, 10, 15, 18, 22, 27, 31, 36, 40, 44

Maus-Überwachung einschalten für ein Objekt ..6, 11, 15, 18, 23, 27, 32, 36, 40, 45

Maximale Anzahl der durch User eingebbaren Zeichen in einem Text-Control.....24, 42

MAXLENGTH.....41

Medien.....18

METHOD.....11

MIME2

MIME-Typen2

Motion Picture Experts Group19

MOV19

MPEG19

multipart/form-data11

Multipurpose Internet Mail Extensions2

nachfolgendes Kind Zeiger3, 7, 12, 16, 19, 24, 29, 33, 37, 42



NAME..... 1, 11, 27
Name des Kindes 3, 7, 12, 16, 20, 24, 29, 33, 37, 42
Name des Objektes 3, 7, 12, 16, 19, 24, 29, 33, 37, 42
Normalisierung des DOM 5, 10, 15, 18, 22, 26, 31, 36, 40, 44
oberer Rand des Fensters 2, 7, 12, 19, 23, 28, 33, 37, 41
oberer Rand des Objektes 3, 8, 12, 20, 24, 29, 33, 38, 42
Objekt Abstand zum linken Rand des Fensters 2, 7, 12, 19, 23, 28, 32, 37, 41
Objekt Abstand zum oberen Rand des Fensters.... 2, 7, 12, 19, 23, 28, 33, 37, 41
Objekt aktivieren 2, 7, 11, 16, 19, 23, 28, 32, 37, 41
Objekt attribute 5, 9, 14, 17, 21, 26, 31, 35, 39, 44
Objekt Attributwert 2, 4, 13, 17, 21, 25, 30, 34, 38, 43
Objekt aus einem Objekt entfernen... 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
Objekt Bezeichner 2, 7, 12, 16, 19, 24, 29, 33, 37, 41
Objekt button 1, 2
Objekt currTimeState .. 2, 3, 4, 7, 8, 11, 12, 13, 16, 19, 20, 23, 24, 25, 28, 29, 30, 32, 33, 34, 37, 38, 41, 42, 43
Objekt document.form.input 1
Objekt durch anderes Objekt komplett ersetzen . 6, 10, 15, 22, 27, 32, 36, 40, 45
Objekt ersetzen durch ein Objekt . 6, 10, 15, 18, 22, 27, 32, 36, 40, 45
Objekt Focussierbarkeit..... 2, 7, 12, 16, 19, 24, 28, 33, 37, 41
Objekt form..... 2, 7, 12, 16, 19, 23, 28, 33, 37, 41
Objekt füllen mit Daten 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Objekt HTML-Attribute 4, 9, 13, 17, 21, 25, 30, 34, 39, 43
Objekt ID . 2, 4, 7, 8, 12, 13, 16, 17, 19, 21, 24, 25, 29, 30, 33, 34, 37, 38, 41, 43
Objekt img 1, 19
Objekt in eine Objekt einfügen 5, 10, 14, 17, 22, 26, 31, 35, 39, 44
Objekt Index in der Collection document.all . 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
Objekt Initialisierung mit Standardwert.... 2, 7, 12, 16, 19, 23, 28, 33, 37, 41
Objekt input 1
Objekt input button..... 2
Objekt input checkbox..... 6
Objekt input checkbox Grauzustand (Dimmed) und Selektiertheit... 7
Objekt input checkbox Selektiertheit 7
Objekt input checkbox Selektionsstatus..... 7, 28
Objekt input checkbox Style 6
Objekt input file..... 11
Objekt input Focus 6, 10, 15, 23, 27, 32, 36, 40, 45
Objekt input hidden 15
Objekt input im Formular markieren . 6, 10, 15, 23, 27, 32, 36, 40, 45
Objekt input image 18, 38
Objekt input markieren..... 6, 10, 15, 23, 27, 32, 36, 40, 45
Objekt input password..... 23
Objekt input radio..... 27
Objekt input radio Selektionsstatus..... 7, 28
Objekt input reset..... 32
Objekt input submit 36
Objekt input text 41
Objekt input Typ..... 4, 8, 13, 16, 20, 25, 30, 34, 38, 43
Objekt input Varinate 4, 8, 13, 16, 20, 25, 30, 34, 38, 43
Objekt Interaktionsfähigkeit . 2, 3, 7, 12, 16, 19, 24, 28, 29, 33, 37, 41
Objekt internes ID 2, 7, 12, 16, 19, 24, 29, 33, 37, 41
Objekt Kind 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
Objekt klonen 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
Objekt Ladezustand 2, 19
Objekt Layout-Komponente 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
Objekt linke obere Ecke 3, 8, 12, 20, 24, 29, 33, 37, 42
Objekt linke obere Ecke 3, 7, 12, 20, 24, 29, 33, 37, 42
Objekt linker Rand 3, 8, 12, 20, 24, 29, 33, 38, 42
Objekt Maus-Überwachung ausschalten . 5, 10, 15, 18, 22, 27, 31, 36, 40, 44
Objekt Maus-Überwachung einschalten . 6, 11, 15, 18, 23, 27, 32, 36, 40, 45
Objekt mit HTML-Tags 2, 7, 11, 16, 19, 23, 28, 32, 37, 41
Objekt Name 3, 7, 12, 16, 19, 24, 29, 33, 37, 42
Objekt oberer Rand..... 3, 8, 12, 20, 24, 29, 33, 38, 42
Objekt Plain-Text 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Objekt rechte untere Ecke 3, 8, 12, 20, 24, 29, 33, 37, 42

Objekt rechte unteren Ecke 3, 7, 12, 20, 24, 29, 33, 37, 42
Objekt Referenz auf TextRectangle..... 5, 9, 14, 21, 26, 31, 35, 39, 44
Objekt scrollen 6, 10, 15, 22, 27, 32, 36, 40, 45
Objekt Selektierbarkeit.... 2, 4, 7, 8, 11, 13, 16, 17, 19, 21, 23, 25, 28, 30, 32, 34, 37, 38, 41, 43
Objekt sichtbar 6, 10, 15, 22, 27, 32, 36, 40, 45
Objekt Sprache 3, 8, 13, 16, 20, 25, 29, 34, 38, 42
Objekt Standardwert..... 2, 7, 12, 16, 19, 23, 28, 33, 37, 41
Objekt Status 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Objekt Tag-Bezeichner 4, 8, 13, 16, 20, 25, 30, 34, 38, 43
Objekt Text..... 5, 9, 14, 17, 21, 26, 31, 35, 39, 43
Objekt Textbereich..... 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Objekt Timeline Typ 25
Objekt Umflussrichtung 2, 7, 12, 19, 24, 28, 33, 37, 41
Objekt Umgebung 3, 8, 12, 20, 24, 29, 33, 38, 42
Objekt Viewbereich 6, 10, 15, 22, 27, 32, 36, 40, 45
Objekt Zeiger auf TextRectangle 5, 9, 14, 21, 26, 31, 35, 39, 44
Objekt Zugriff per Alt + Taste 2, 7, 11, 19, 23, 28, 32, 37, 41
Objekt Zustand des Ladens 2, 19
Objektaktivität 2, 7, 12, 16, 19, 24, 28, 33, 37, 41
Objekt-Breite..... 2, 7, 12, 19, 23, 28, 33, 37, 41
Objekt-Content Editierbarkeit..... 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
Objekt-Content-Editierbarkeit 2, 23, 28, 33, 37, 41
Objekthöhe 2, 7, 11, 19, 23, 28, 32, 37, 41
Objektinhalt Mehrzeiligkeit 3, 7, 12, 16, 19, 24, 29, 33, 37, 41
Objektname logischer..... 2, 7, 12, 16, 19, 24, 29, 33, 37, 41
onblur 4, 9, 13, 21, 25, 30, 34, 39, 43
onclick ... 4, 6, 9, 11, 14, 15, 18, 21, 23, 25, 27, 30, 32, 35, 36, 39, 40, 43, 45
ondblclick 6, 11, 15, 18, 23, 27, 32, 36, 40, 45
onmousedown... 5, 6, 10, 11, 15, 18, 22, 23, 27, 31, 32, 36, 40, 44, 45
onmousemove... 5, 6, 10, 11, 15, 18, 22, 23, 27, 31, 32, 36, 40, 44, 45
onmouseout 5, 6, 10, 11, 15, 18, 22, 23, 27, 31, 32, 36, 40, 44, 45
onmouseover 5, 6, 10, 11, 15, 18, 22, 23, 27, 31, 32, 36, 40, 44, 45
onmouseover Pixelgenauigkeit 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
onmouseup 5, 6, 10, 11, 15, 18, 22, 23, 27, 31, 32, 36, 40, 44, 45
Padding..... 6
password..... 1
Password..... 23
Plain-Text .. 5, 6, 10, 14, 15, 17, 18, 22, 26, 27, 31, 32, 35, 36, 40, 44, 45
Plain-Text im Objekt..... 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
PNG 19
Portable Network Graphics 19
Positionen von 2 Knoten im DOM tauschen 6, 11, 15, 18, 23, 27, 32, 36, 41, 45
radio..... 1
Radio Button-Control Selektionsstatus..... 7, 28
Radio-Button-Control 27
Rahmen..... 2, 7, 12, 19, 23, 28, 32, 37, 41
rechte untere Ecke des Objektes 3, 8, 12, 20, 24, 29, 33, 37, 42
rechte unteren Ecke des Objektes 3, 7, 12, 20, 24, 29, 33, 37, 42
Rectangle..... 5, 9, 14, 22, 26, 31, 35, 39, 44
Referenz auf Feld der Zeiger auf TextRectangle-Objekte 5, 9, 14, 21, 26, 31, 35, 39, 44
Referenz auf Bereich ab inklusive HTML-Start bis hinter -Ende-Tag 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
Referenz auf das document Objekt des Knoten 3, 8, 12, 20, 24, 29, 33, 38, 42
Referenz auf das Elternobjekt Body ... 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Referenz auf das ERSTE Kind 2, 7, 12, 19, 24, 28, 33, 37, 41
Referenz auf das LETZTE Kind 3, 7, 12, 19, 24, 29, 33, 37, 42
Referenz auf das NACHFOLGENDE Kind ... 3, 7, 12, 16, 19, 24, 29, 33, 37, 42
Referenz auf das Vorgängerkind ... 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Referenz auf den gesamten Plain-Text im Objekt. 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Referenz auf Elternknoten 3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Referenz auf TextRectangle-Objekt 5, 9, 14, 21, 26, 31, 35, 39, 44
Referenz der Eltern 3, 8, 12, 16, 20, 24, 29, 33, 37, 42
regisitrieren eines Events 4, 9, 14, 17, 21, 26, 30, 35, 39, 43
registrieren eines Events 4, 9, 13, 17, 21, 25, 30, 34, 39, 43
reset 1



Reset-Button für Formular	32
Reset-Button Label	32
Reset-Button Titel	32
Script-Code einfügen	5, 10, 14, 17, 22, 26, 31, 35, 39, 44
Scriptsprache	3, 7, 12, 16, 19, 24, 29, 33, 37, 42
Scrollbalken	2, 7, 12, 19, 23, 28, 32, 37, 41
Scrollbereich horizontal	3, 8, 13, 20, 24, 29, 34, 38, 42
Scrollbereich vertikal	3, 8, 12, 20, 24, 29, 33, 38, 42
Selektierbarkeit des Objektes	2, 7, 11, 16, 19, 23, 28, 32, 37, 41
Selektierbarkeit eines Objektes	4, 8, 13, 17, 21, 25, 30, 34, 38, 43
Selektiertheit des Checkbox-Control	7
Selektionsfähigkeit eines Objektes	4, 8, 13, 17, 21, 25, 30, 34, 38, 43
Selektionsstatus Checkbox-Control	7, 28
Selektionsstatus eines Control-Elementes	8, 29
Selektionsstatus Radio Button-Control	7, 28
Senden eines Formulars	18
Server	11, 15
setCapture()	5, 10, 15, 18, 22, 27, 31, 36, 40, 44
setExpression()	5, 10, 15, 18, 22, 27, 32, 36, 40, 44
SIZE	41
Sonderzeichen	3, 7, 12, 16, 19, 24, 29, 33, 37, 41
Sound	19
Sound Wiederholungen	19
Sprache für Anzeige von Sonderzeichen	3, 7, 12, 16, 19, 24, 29, 33, 37, 41
Sprache für Script	3, 7, 12, 16, 19, 24, 29, 33, 37, 42
Sprache zum Objekt	3, 8, 13, 16, 20, 25, 29, 34, 38, 42
Standardbehandlung Ereignis	4, 9, 14, 17, 21, 26, 31, 35, 39, 43
Standard-IE-Eigenschaften	4, 9, 13, 17, 21, 25, 30, 34, 38, 43
Standardwert laut Objektinitialisierung	2, 7, 12, 16, 19, 23, 28, 33, 37, 41
Start einer VRML-Datei	2, 20
Start eines Videoclips	2, 20
Status des Autovervollständigung zum Formular	23, 41
Status des Objektes	3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Status Drag-Manipulation	4, 9, 14, 21, 26, 31, 35, 39, 43
STYLE	3, 8, 13, 16, 20, 25, 29, 34, 38, 42
Style Checkbox-Control	6
Style input checkbox Objekt	6
Style-Eigenschaft Wert	5, 6, 9, 10, 11, 14, 15, 17, 18, 22, 23, 26, 27, 31, 32, 35, 36, 39, 40, 44, 45
submit	1
Submit Query	36
Submit-Button für Formular	36
TABINDEX	4, 5, 9, 13, 14, 21, 25, 26, 30, 31, 34, 35, 39, 43
Tab-Tasten-Folge	3, 8, 13, 20, 25, 30, 34, 38, 42
TAG-Bezeichner	3, 7, 12, 16, 20, 24, 29, 33, 37, 42
Tag-Bezeichner des Objektes	4, 8, 13, 16, 20, 25, 30, 34, 38, 43
Tastaturzugriff auf ein Objekt per Alt + Taste	2, 7, 11, 19, 23, 28, 32, 37, 41
Taste-Control	2
Tastenkombination	2, 7, 11, 19, 23, 28, 32, 37, 41
text	1
Text als Tooltip	2, 19
Text eines Objektes	5, 9, 14, 17, 21, 26, 31, 35, 39, 43
Textbereich	4, 17, 26, 35, 39, 43
Textbereich des Elternobjektes	3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Textbox	25, 43
Textbox einzeilige	41
Textbox Länge	24, 42
Text-Control	23, 25, 43
Textcontrol einzeilig	41
Text-Control Länge	24, 42
Text-Control maximale Anzahl der durch User eingebbarer Zeichen	24, 42
Textfeld	25, 43
Textfeld Länge	24, 42
TextRectangle-Objekt	5, 9, 14, 21, 26, 31, 35, 39, 44
TIMEACTION	2, 7, 11, 12, 16, 19, 23, 24, 28, 32, 33, 37, 41
Timeline	2, 7, 11, 16, 19, 23, 28, 32, 37, 41
Timeline Synchronisierung der Animation des im Container liegenden Elementes auf Timeline	3, 8, 13, 16, 20, 25, 29, 34, 38, 42
Timeline Typ	25
Titel des Reset-Button	32
Tooltip	2, 19
Tooltip-Text	4, 8, 13, 20, 25, 30, 34, 38, 43
Typ der Timeline des Objektes	25
Typ input Objekt	4, 8, 13, 16, 20, 25, 30, 34, 38, 43
TYPE	1
Umfflussrichtung	2, 7, 12, 19, 24, 28, 33, 37, 41
Umgebungsobjekt	3, 8, 12, 20, 24, 29, 33, 38, 42
Uniform Resource Name	4, 8, 13, 16, 20, 25, 30, 34, 38, 43
UNSELECTABLE	4, 8, 13, 17, 21, 25, 30, 34, 38, 43
Url der Daten	20
Url des Image in geringerer Auflösung	19
Url von Videoclip	2, 19
URN	4, 8, 13, 16, 20, 25, 30, 34, 38, 43
Usereingabe im Formular	41
VALUE	2, 4, 8, 13, 17, 21, 25, 30, 32, 34, 38, 43
Varinate input Objekt	4, 8, 13, 16, 20, 25, 30, 34, 38, 43
vertikaler Scrollbereich	3, 8, 12, 20, 24, 29, 33, 38, 42
Video Tooltip	2, 19
Videoclip	2, 19
Videoclip starten	2, 20
Videoclip Url	2, 19
Viewbereich Objekt	6, 10, 15, 22, 27, 32, 36, 40, 45
Vorgängerkind	3, 8, 12, 16, 20, 24, 29, 33, 38, 42
VRML	19
VRML-Datei starten	2, 20
Wert einer Style-Eigenschaft	5, 6, 9, 10, 11, 14, 15, 17, 18, 22, 23, 26, 27, 31, 32, 35, 36, 39, 40, 44, 45
Wert eines Attributes belegen	6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Wert eines Objekt-Attributes	2, 4, 13, 17, 21, 25, 30, 34, 38, 43
Wert eines per HTML-Attributes	5, 9, 14, 17, 21, 26, 31, 35, 39, 44
Wert vom Attribut	6, 10, 15, 18, 23, 27, 32, 36, 40, 45
Wiederholungen Hintergrundmusik	19
Wiederholungen Sound	19
Windows Bitmap	18
Windows Enhanced Metafile	18
Windows Metafile	19
WMF	19
X Bitmap	19
XBM	19
X-Koordinate der rechten unteren Ecke des Objektes	3, 8, 12, 20, 24, 29, 33, 37, 42
XMLNS	3, 4, 8, 12, 13, 16, 20, 24, 25, 29, 30, 33, 34, 38, 42, 43
Y-Koordinate der linken oberen Ecke des Objektes	3, 8, 12, 20, 24, 29, 33, 37, 42
Y-Koordinate der rechten unteren Ecke des Objektes	3, 7, 12, 20, 24, 29, 33, 37, 42
Zeiger auf Element in einem Formular	2, 7, 12, 16, 19, 24, 28, 33, 37, 41
Zeiger auf das ERSTE Kind	2, 7, 12, 19, 24, 28, 33, 37, 41
Zeiger auf das LETZTE Kind	3, 7, 12, 19, 24, 29, 33, 37, 42
Zeiger auf das NACHFOLGENDE Kind	3, 7, 12, 16, 19, 24, 29, 33, 37, 42
Zeiger auf das Vorgängerkind	3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Zeiger auf Elternknoten	3, 8, 12, 16, 20, 24, 29, 33, 38, 42
Zeiger auf TextRectangle-Objekt	5, 9, 14, 21, 26, 31, 35, 39, 44
Zeigertausch	6, 11, 15, 18, 23, 27, 32, 36, 41, 45
Zugriff per Alt + Taste	2, 7, 11, 19, 23, 28, 32, 37, 41
Zustand des Ladens des Objektes	2, 19

