

area Objekt des Internet Explorer

Area-Tag

Scripting ab 4.x

siehe auch Objekt map

Erzeugung:

Beispiel 1:

```

<IMG ID="ID_Img"
SRC="test.gif"
USEMAP="#MapName"
>
<MAP ID="ID_Map"
NAME="MapName">
<AREA SHAPE="rect" COORDS="0,0,33,33" HREF="test1.gif">
<AREA SHAPE="circle" COORDS="90,33,3" HREF="test2.gif">
</MAP>

```

Beispiel 2:

```

<MAP
NAME="logischer_map_name"
<AREA ID="ID_Area"
NAME="name_des_verweissensitiven_bereiches_der_grafik"
COORDS="koordinaten_liste_des_bereiches"
HREF="url"
SHAPE= "rect"
oder "poly"
oder "circle"
oder "default"
TARGET="logischer_window_name"
onClick="eventhandler_1"
onMouseOut="eventhandler_2"
onMouseOver="eventhandler_3"
> .....
</MAP>

```

SHAPE: rect Rechteck
 poly Vieleck
 circle Kreis

COORDS: bei rect und poly "x1,y1,....,xn,yn"
 circle "x,y,r"
 x Spalte in Pixel
 y Zeile in Pixel
 r Radius in Pixel

Hinweis: mehrere AREA sind möglich

Zugriff:

ID_Map.eigenschaft
ID_Map.methode()

Eigenschaften:

- .accessKey Tastaturzugriff auf ein Objekt per Alt + Taste
bei Ausführung der Tastenkombination wird
das Sprungziel wird focussiert
die Sprungquelle defocussiert
das Focus-Ereignis ausgelöst
- .alt vor IE 5.x funktioniert Tastenkombination nicht mit jedem Objekt
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
- .canHaveHTML prüfen ob Objekt HTML-Tags enthalten darf
- .className Klassenreferenz, Klassenname
- .coords Koordinaten eines Objektes in Abhängigkeit zur Eigenschaft .shape
- .dir Umflussrichtung
- .disabled Interaktionsfähigkeit
nur wenn sichtbar so User-Interaktion möglich
- .end Objektaktivitäten laut Eigenschaft .timeAction beenden
ab IE 6.x
alternativ: Eigenschaft .dur
siehe Objekt currTimeState und Behavior .style.time2
- .firstChild Zeiger auf das ERSTE Kind laut childNodes-Collection eines Objektes
- .hash Teil des Wertes der Eigenschaft .href also Teil der Url als Anker
Teil folgt direkt hinter dem Nummernkreuz # und ohne Nummernkreuz
- .hasMedia Objekt ist HTML-Media-Objekt
siehe Objekt currTimeState und Behavior .style.time2



.hideFocus	Focussierbarkeit						
.host	Hostname und Port einer Location oder Url in der Form " hostname:port "						
.hostname	Hostname einer Location oder Url in der Form " hostname:port " kann Domain oder IP sein						
.href	Ziel-Url oder Anker als Sprungziel (z.B. <A>-Tag) siehe auch Eigenschaften .rel und .rev						
.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						
.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)						
.noHref	HREF-Wirkung (Eigenschaft .href) ein/ aus bei Click auf den Link						
.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)						
.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						
.pathname	Datei und Pfad eines Objektes						
.port	Port einer Location oder Url in der Form " hostname:port " basierend auf dem aktuellen Protokoll laut Eigenschaft .protocol z.B. <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <th>Standard-Ports</th> <th>Protokoll</th> </tr> <tr> <td>21</td> <td>FTP</td> </tr> <tr> <td>80</td> <td>HTTP</td> </tr> </table>	Standard-Ports	Protokoll	21	FTP	80	HTTP
Standard-Ports	Protokoll						
21	FTP						
80	HTTP						
.previousSibling	Referenz auf das Vorgängerkind						
.protocol	Protokoll-Teil einer Url inklusive http und ftp liefern						
.readyState	aktueller Status des Objektes beim Füllen des Objektes mit Daten						
.scopeName	Namensraum laut XMLNS-Attribut						
.search	Teil des Wertes des Eigenschaft .href Teil folgt direkt dem Fragezeichen wird als Querystring oder Formdata bezeichnet hat nichts mit der Suche auf Webseiten zu tun Hinweis: Fragezeichen-Anhang an der HREF dient zur Übergabe von String-Werten einer Webseite an eine andere: Quellseite besitzt HREF mit " ...?....." ruft Zielseite mit diesem HREF auf Zielseite wurde von Quellseite aufgerufen liest den Teil von HREF hinter dem ? als Zeichenkettendaten						
.shape	Form/Gestalt eines Objektes siehe auch Eigenschaft .coords						
.sourceIndex	Index des Objektes in der Collection document.all						
STYLE	direkt im HTML-Element kodierter Style (Inline-Style) Hinweis: für Scripting ist das Style-Objekt zu nutzen						
.syncMaster	Synchronisierung der Animation des im Container liegenden Elementes auf Timeline						



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

.systemBitrate	wird hier nicht erklärt
.systemCaptions	wird hier nicht erklärt
.systemLanguage	Sprache festlegen für das Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.tabIndex	Index des Elementes in der Tab-Tasten-Folge für Anspringen des Dokumentes Anspringen verbunden mit Focus erhalten --> Ereignisse werden ausgelöst !! unter IE 5.x onblur, onfocus ab IE 5.x onblur, onfocus, onkeydown, onkeypress, onkeyup Anspringen default per TAB-Taste für A, BODY, BUTTON, FRAME, IFRAME, IMG, INPUT, SELECT, TEXTAREA Anspringen default nicht per TAB-Taste für APPLET, DIV, FRAMESET, SPAN, TABLE, TD
.tagName	Tag-Bezeichner des Objektes
.tagUrn	Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut
.target	Name des Ziel-Fenster bzw. Ziel-Frame
.timeContainer	Typ der Timeline des Objektes siehe Objekt currTimeState und Behavior .style.time2
.title	Tooltip-Text bei Mouse over über Objekt
.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
Methoden:	
.addBehavior()	DHTML-Verhaltenseigenschaft einem Element hinzufügen Empfehlung: Standard-IE-Eigenschaften nutzen, da diese mit "#default#behaviorName" komplett erfasst werden und bereits im Browser implementiert sind (keine HTC-Datei nötig). ab IE 5.x bis unter IE 5.5
.applyElement()	Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern DOM wird geändert 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_bezeichnet 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
.getElementsByName()	Referenz auf ein Feld (Collection) aller im Objekt befindlichen Kinder-Objekte mit gemeinsamen Tagnamen liefern, inklusive aller Kinder und Unterkinder etc. Hinweis: Natürlich kann auch das document-Objekt so verarbeitet werden (beachte dabei document.all Collection) Achtung: Kinder-Objekte, die keinen Tag-Name besitzen, werden nicht erfasst ! Für Verwaltung per ID (analog zum ID-Attribut): siehe Methode getElementById() Für Verwaltung per NAME (analog zum NAME-Attribut): siehe Methode .getElementsByName()
.getExpression()	DOM nicht geändert 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
.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



<code>.removeAttributeNode()</code>	DOM wird geändert entfernen von Attribut, egal ob es mit oder ohne HTML-Anweisung erzeugt wurde, und Referenz auf das entfernte Attribut liefern
<code>.removeBehavior()</code>	DOM wird geändert per Methode <code>.addBehavior()</code> einem Element hinzugefügte Verhaltenseigenschaft entfernen (stets VOR dem Entfernen des Elementes mit der zugeordneten Eigenschaft aus der Dokument-Hierarchie)
<code>.removeExpression()</code>	DOM wird geändert 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
<code>.replaceAdjacentText()</code>	DOM wird nicht geändert Plain-Text (ohne HTML und Script) eines Elementes durch anderen Text ersetzen und Referenz auf den zu ersetzenden Text liefern
<code>.scrollIntoView()</code>	DOM wird nicht geändert Objekt derart scrollen, dass es im Fenster für User sichtbar wird Objekt muss an sich schon renderbar sein
<code>.setActive()</code>	Objekt für die Eventdurchreichung aktivieren aber ohne es zu fokussieren und ohne es scrollbar zu machen
<code>.setAttribute()</code>	Wert von vorhandenem Attribut setzen wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt
<code>.setAttributeNode()</code>	DOM wird nur bei Erzeugung geändert Attribut einem Knoten zuweisen und Referenz liefern
<code>.setCapture()</code>	DOM wird geändert 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 Hinweis: ausschalten per Methode <code>.releaseCapture()</code>
<code>.setExpression()</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
<code>.swapNode()</code>	DOM wird nicht geändert Positionen von 2 Knoten im DOM tauschen (Zeigertausch) nur sichtbar wenn Endetag geparkt DOM wird geändert

