

## applet Objekt des Internet Explorer

APPLET-Tag ist ab HTML 4.01 deprecated: Es soll OBJECT verwendet werden.

Objekt zum Einbinden ausführbaren Codes auf Basis einer Javamaschine (Virtuelle Javamaschine).

Applet muss alle Eigenschaften und Methoden, die im HTML-Dokument benutzt werden sollen, als **public** deklarieren.

### Java-Applet (\*.class) in HTML einbinden:

Beispiel

```

<APPLET
    ID="ID_Applet"
    CODE="class_datei"
    CODEBASE="quellverzeichnis"
    HEIGHT=applet_fenster_hoehe_in_pixel
    WIDTH=applet_fenster_breite_in_pixel
    MAYSCRIPT=true oder false
    ALT="alternativer_text"
    ALIGN=ausrichtung
    HSPACE=fenster_abstand_links_und_rechts_von_umgebung
    VSPACE=fenster_abstand_loben_und_unten_von_umgebung
>
    <PARAM Name="parameter_name"
        VALUE=parameter_wert
    >
    .....
</APPLET>

```

### Zugriff:

**ID\_Applet**.eigenschaft  
**ID\_Applet**.methode()

### Eigenschaften:

|                           |  |
|---------------------------|--|
| <b>.accessKey</b>         | Tastaturzugriff auf ein Objekt per Alt + Taste<br>bei Ausführung der Tastenkombination wird<br>das Sprungziel wird focussiert<br>die Sprungquelle defocussiert<br>das Focus-Ereignis ausgelöst   |
| <b>.align</b>             | vor IE 5.x funktioniert Tastenkombination nicht mit jedem Objekt<br>Ausrichtung  |
| <b>.alt</b>               | Alternativer Text als Tooltip  |
| <b>.altHTML</b>           | HTML-Script, der ausgeführt wird, wenn Objekt nicht geladen werden kann  |
| <b>.archive</b>           | Zeichenkette für Archive-Funktionalität eines Objektes   |
| <b>.canHaveHTML</b>       | prüfen ob Objekt HTML-Tags enthalten darf  |
| <b>.className</b>         | Klassenreferenz, Klassenname   |
| <b>.clientHeight</b>      | Objekthöhe in Pixel ohne Abstand zum Umgebungsobjekt<br>ohne Rahmen<br>ohne Scrollbalken   |
| <b>.clientLeft</b>        | Abstand in Pixel zum linken Rand des Fensters  |
| <b>.clientTop</b>         | Abstand in Pixel zum oberen Rand des Fensters  |
| <b>.clientWidth</b>       | Objekt-Breite in Pixel ohne Abstand zum Umgebungsobjekt<br>ohne Rahmen<br>ohne Scrollbalken  |
| <b>.code</b>              | Url der *.class-Datei (kompilierter Javacode)  |
| <b>.codeBase</b>          | Url der Komponente für Dowload der Komponente vom Server auf den Client<br>optionale Versionsprüfung möglich (nicht bei Applet-Komponente) um unnötigen<br>Download zu ersparen: vorheriger Ableich der Version der Komponente auf Server und Client<br>auch wenn der Versionsabgleich keinen Download ergab, wird immer HTTP Header-Transaktion<br>ausgelöst                          |
| <b>.dataFld</b>           | Datenquelle-Name vergeben (ID)   |
| <b>.dataSrc</b>           | Datenquelle als Anker festlegen  |
| <b>.firstChild</b>        | Zeiger auf das ERSTE Kind laut childNodes-Collection eines Objektes  |
| <b>.hideFocus</b>         | Focussierbarkeit   |
| <b>.hspace</b>            | horizontaler Abstand in Pixel zum Elternobjekt   |
| <b>.id</b>                | Bezeichner des Objektes für Referenzierung des Objektes (Zeiger, ID, logischer Objektname)<br>Hinweis: Browser erzeugt pro Objekt ein internes ID, das per Eigenschaft .uniqueID ermittelt<br>und anstelle der Eigenschaft .id verwendet werden kann (falls Browser und<br>betroffenes Objekt die Eigenschaft .uniqueID kennen).<br>Zeiger aus ID bilden var Zeiger = eval(object.id); |
| <b>.isContentEditable</b> | Editierbarkeit des Objekt-Content (auch wenn kein Layout hat)<br>Content = Beziehung des Objektes zum Umfeld z.B. bezüglich Layout etc.  |
| <b>.isDisabled</b>        | Interaktionsfähigkeit<br>nur wenn sichtbar so User-Interaktion möglich   |
| <b>.isMultiLine</b>       | Mehrzeiligkeit des Objektinhaltes  |
| <b>.isTextEdit</b>        | Erzeugbarkeit eines Textbereiches  |
| <b>.lang</b>              | Sprache für Anzeige von Sonderzeichen etc.   |
| <b>.language</b>          | Sprache für Script festlegen   |



|                  |  |
|------------------|--|
| .lastChild       | Zeiger auf das LETZTE Kind laut childNodes collection eines Objektes   |
| .name            | Name des Objektes (nicht ID !!!)<br>muss beim Formular für alle zu sendenden Felder kodiert sein !!<br>Element darf nicht per Methode .createElement() erzeugt worden sein   |
| .nextSibling     | Zeiger auf das NACHFOLGENDE Kind laut childNodes collection eines Objektes   |
| .nodeName        | String als Name des Kindes (Knoten, Node, Element)<br>also TAG-Bezeichner, Attribut-Name; #text für Anker  |
| .nodeType        | Knotentyp laut attributes Collection   |
| .nodeValue       | Knotenwert (Wert des Kindes, Node, Elementes)<br>nur für Text- und Attribut-Elemente<br>nicht für Element-Knoten (Knotentyp 1)   |
| .object          | Zeiger auf das Objekt laut Applet bzw. laut Objekt object  |
| .offsetHeight    | Y-Koordinate der rechten unteren Ecke des Objektes bezüglich Koordinatensystem des Elternobjektes (.offsetParent)  |
| .offsetLeft      | X-Koordinate der linken oberen Ecke Objektes bezüglich Koordinatensystem des Elternobjektes (.offsetParent)  |
| .offsetParent    | Referenz der Eltern<br>für Nutzung von .offsetHeight, .offsetLeft, .offsetTop und .offsetWidth   |
| .offsetTop       | Y-Koordinate der linken oberen Ecke des Objektes bezüglich Koordinatensystem des Elternobjektes (.offsetParent)  |
| .offsetWidth     | X-Koordinate der rechten unteren Ecke des Objektes bezüglich Koordinatensystem des Elternobjektes (.offsetParent)  |
| .outerHTML       | Referenz auf Bereich ab inklusive HTML-Start bis hinter -Ende-Tag<br>wirksam mit parsen des Ende-Tag<br>nur nach kompletten Einlesen des Dokumentes nutzbar  |
| .outerText       | Referenz auf den gesamten Plain-Text im Objekt<br>nur nach kompletten einlesen des Dokumentes nutzbar  |
| .ownerDocument   | Referenz auf das document Objekt zu dem der Knoten gehört, also in dem der Knoten erzeugt wurde  |
| .parentElement   | Referenz auf das Elternobjekt, also nicht Elternknoten innerhalb DOM   |
| .parentNode      | Referenz auf Elternknoten innerhalb der DOM-Hierarchie   |
| .parentTextEdit  | Textbereich des Elternobjektes referenzieren   |
| .previousSibling | Referenz auf das Vorgängerkind   |
| .readyState      | aktueller Status des Objektes beim Füllen des Objektes mit Daten   |
| .scopeName       | Namensraum laut XMLNS-Attribut   |
| .scrollHeight    | Höhe des vertikalen Scrollbereiches, also Abstand von oberen und unteren sichtbaren Rand des Umgebungsobjektes   |
| .scrollLeft      | Abstand von linken sichtbaren Randes des Umgebungsobjektes zum linken Rand des Objektes<br>nur nutzbar nach dem kompletten Laden des Dokumentes  |
| .scrollTop       | Abstand von oberen sichtbaren Randes des Umgebungsobjektes zum oberen Rand des Objektes<br>nur nutzbar nach dem kompletten Laden des Dokumentes  |
| .scrollWidth     | Breite des horizontalen Scrollbereiches, also Abstand von linken und rechten   |
| .sourceIndex     | Index des Objektes in der Collection document.all  |
| .src             | Url der Daten z.B. vom Image in normaler Auflösung   |
| STYLE            | direkt im HTML-Element kodierter Style (Inline-Style)<br>Hinweis: für Scripting ist das Style-Objekt zu nutzen   |
| .tabIndex        | Index des Elementes in der Tab-Tasten-Folge<br>für Anspringen des Dokumentes<br>Anspringen verbunden mit Focus erhalten<br>--> Ereignisse werden ausgelöst !!<br>unter IE 5.x<br>onblur, onfocus<br>ab IE 5.x<br>onblur, onfocus, onkeydown, onkeypress, onkeyup<br>Anspringen default per TAB-Taste<br>für A, BODY, BUTTON, FRAME, IFRAME, IMG, INPUT, SELECT, TEXTAREA<br>Anspringen default nicht per TAB-Taste<br>für APPLET, DIV, FRAMESET, SPAN, TABLE, TD |
| .tagName         | Tag-Bezeichner des Objektes  |
| .tagUrn          | Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut  |
| .title           | Tooltip-Text bei Mouse over über Objekt  |
| .uniqueID        | durch den Browser automatisch-generiertes ID des Objektes<br>Browser generiert zu verschiedenen Zeitpunkten auch verschiedene ID, wenn Objekt mehrmals geladen wurde<br>kann anstelle eines privat vergebenen ID als ID-Attributwert weiterverwendet werden  |
| UNSELECTABLE     | Selektionsfähigkeit eines Objektes   |
| .vspace          | vertikaler Abstand in Pixel zum Elternobjekt   |
| <b>Methoden:</b> |  |
| .addBehavior()   | DHTML-Verhaltenseigenschaft einem Element hinzufügen<br>Empfehlung: Standard-IE-Eigenschaften nutzen, da diese mit "#default#behaviorName" komplett erfasst werden und bereits im Browser implementiert sind (keine HTC-Datei nötig).<br>ab IE 5.x bis unter IE 5.5  |
| .applyElement()  | Elementeigenschaft "Kind sein" oder "Eltern sein" festlegen, also die Lage im DOM, und danach Referenz laut DOM liefern  |



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



|                        |   |
|------------------------|---|
| .mergeAttributes()     | <p>alle Attribute eines Elementes in ein anderes Element kopieren und eventuell die Attribute im Ziel mischen</p> <p>Attribute sind: HTML<br/>Events<br/>Styles<br/>ab IE 5.01 auch ID, NAME</p> <p>Achtung: Diese Methode ist mir Vorsicht zu geniessen !!</p> <p>DOM wird geändert</p>              |
| .namedRecordset()      | <p>Zeiger desjenigen Datenquellen-Record-Objektes mit Namen, das den Standard-Record-Set enthält</p> <p>Hinweis: Namen zeigt die Mitgliedschaft in einem Datasource-Objekt (DSO) an</p> <p>Eigenschaft .recordset liefert den Zeiger für den Standard-Record-Set in einem Datasource-Objekt (DSO)</p> |
| .normalize()           | <p>Normalisierung des DOM zur Erreichung einer konsistenten Struktur</p> <p>Achtung: CDATA-Sections dürfen nicht enthalten sein, da diese immer Inkonsistenz erzeugen</p>   |
| .releaseCapture()      | <p>Maus-Überwachung ausschalten für ein Objekt</p> <p>Maus-Events sind : onmousedown, onmouseup, onmousemove, onclick, ondblclick,<br/>onmouseover und onmouseout.</p> <p>Hinweis: einschalten per Methode .setCapture()</p>  |
| .removeAttribute()     | <p>entfernen eines per HTML erzeugten Attributes</p> <p>Achtung: Der Browser unterscheidet zwischen HTML-erzeugte oder mit dieser Methode erzeugte Attribute!</p> <p>per Methode .createAttribute() erzeugte Attribute werden nicht erfasst</p> <p>DOM wird geändert</p>                              |
| .removeAttributeNode() | <p>entfernen von Attribut, egal ob es mit oder ohne HTML-Anweisung erzeugt wurde, und Referenz auf das entfernte Attribut liefern</p> <p>DOM wird geändert</p>  |
| .removeBehavior()      | <p>per Methode .addBehavior() einem Element hinzugefügte Verhaltenseigenschaft entfernen (stets VOR dem Entfernen des Elementes mit der zugeordneten Eigenschaft aus der Dokument-Hierarchie)</p> <p>DOM wird geändert</p>  |
| .removeExpression()    | <p>Ausdruck entfernen, der für die Berechnung des Wertes einer Style-Eigenschaft als Objektreferenz der Form objekt.style.eigenschaft. dient.</p> <p>Ausdruck muss mit der Methode .setExpression() gesetzt worden sein</p> <p>DOM wird nicht geändert</p>  |
| .replaceAdjacentText() | <p>Plain-Text (ohne HTML und Script) eines Elementes durch anderen Text ersetzen und Referenz auf den zu ersetzenden Text liefern</p> <p>DOM wird nicht geändert</p>  |
| .scrollIntoView()      | <p>Objekt derart scrollen, dass es im Fenster für User sichtbar wird</p> <p>Objekt muss an sich schon renderbar sein</p>  |
| .setActive()           | <p>Objekt für die Eventdurchreichung aktivieren</p> <p>aber ohne es zu fokussieren<br/>und ohne es scrollbar zu machen</p>  |
| .setAttribute()        | <p>Wert von vorhandenem Attribut setzen</p> <p>wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt</p> <p>DOM wird nur bei Erzeugung geändert</p>  |
| .setAttributeNode()    | <p>Attribut einem Knoten zuweisen und Referenz liefern</p> <p>DOM wird geändert</p>   |
| .setCapture()          | <p>Maus-Überwachung einschalten für ein Objekt</p> <p>Maus-Events sind : onmousedown, onmouseup, onmousemove, onclick, ondblclick,<br/>onmouseover und onmouseout.</p> <p>ab IE 5.5</p> <p>Hinweis: ausschalten per Methode .releaseCapture()</p>   |
| .setExpression()       | <p>Wert definieren, der als Ausdruck für die Methode .getExpression() zur Berechnung einer Style-Eigenschaft als Objektreferenz der Form objekt.style.eigenschaft. dient</p> <p>Ausdruck nur als Script kodierbar</p> <p>DOM wird nicht geändert</p>  |
| .swapNode()            | <p>Positionen von 2 Knoten im DOM tauschen (Zeigertausch)</p> <p>nur sichtbar wenn Endetag geparkt</p> <p>DOM wird geändert</p>   |

