

select.option Objekt des Internet Explorer

Objekt einer Option aus einer Selektion per document.select Objekt

Option ist anwählbar

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

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

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

Beispiel 1:

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

Beispiel 2:

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

Beispiel 3:

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

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

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

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

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

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

Beispiel 5:

```
<SCRIPT>
  function AusschnittSetzen()
  {
    ID_Image.style.clip= "rect(0,100, "
      + ID_select.options(ID_select.selectedIndex).value
      + ",100)";

    if (ID_Image.currentStyle.clipBottom == "60px")
    { alert("60 Pixel"); }
  }
</SCRIPT>
<IMG ID="ID_Image" SRC="test.jpg">
<SELECT ID="ID_Select" onchange="AusschnittSetzen()">
  <OPTION VALUE=100>100 Pixel </OPTION>
  <OPTION VALUE=40>40 Pixel </OPTION>
  <OPTION VALUE=50>50 Pixel </OPTION>
  <OPTION VALUE=60>60 Pixel </OPTION>
</SELECT>
```



Beispiel 6:

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

Beispiel 7:

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

Beispiel 8:

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

Beispiel 9:

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

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

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

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

Zugriff:

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

```
zeiger_auf_select_objekt z.B. laut ID-Attribut
zeiger_auf_option_objekt z.B. laut ID-Attribut
```

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

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



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

zeiger_auf_select_objekt z.B. laut ID-Attribut

Eigenschaften:

.begin Wartezeit einstellen und dann Objekt aktivieren laut Eigenschaft .timeAction
 siehe Objekt currTimeState und Behavior .style.time2

.canHaveChildren prüfen ob Kind möglich ist, also ob Objekt Parent sein kann

.canHaveHTML prüfen ob Objekt HTML-Tags enthalten darf

.className Klassenreferenz

.clientHeight Objekthöhe in Pixel ohne Abstand zum Umgebungsobjekt
 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

.defaultSelected Default-Selektionsstatus der Option, also Objektes document.select.option des Objektes document.select
 standardgemäß ist immer die oberste Option selektiert
 es kann aus einer Optionengruppe immer nur genau 1 Option selektiert sein

.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

.id Bezeichner des Objektes für Referenzierung des Objektes (Zeiger, ID)

.index laufende Nummer der Option, also Objektes document.select.option, in einer List-Box als
 document.select Objekt

.innerHTML Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag
 ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B.
 dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes,
 also nach dem Laden des Objektes
 aber wirksam erst mit parsen des HTML-Endetag

.innerText Referenz auf den Bereich zwischen HTML-Start und -Ende-Tag
 ein Tag ohne Ende-Tag kann kein innerHTML haben, z.B.
 dient zur Veränderung dieses Bereiches zur Laufzeit des Dokumentes,
 also nach dem Laden des Objektes
 aber wirksam erst mit parsen des HTML-Endetag

.isContentEditable Editierbarkeit des Objekt-Content (auch wenn kein Layout hat)
 Content = Beziehung des Objektes zum Umfeld z.B. bezüglich Layout etc.

.isDisabled Interaktionsfähigkeit
 nur wenn sichtbar so User-Interaktion möglich

.isMultiLine Mehrzeiligkeit des Objektinhaltes

.isTextEdit Erzeugbarkeit eines Textbereiches

.label Label einer Option, also Objektes document.select.option des Objektes document.select

.lang Sprache für Anzeige von Sonderzeichen etc.

.language Sprache für Script festlegen

.lastChild Zeiger auf das LETZTE Kind laut childNodes collection eines Objektes

.nextSibling Zeiger auf das NACHFOLGENDE Kind laut childNodes collection eines Objektes

.nodeName String als Name des Kindes (Knoten, Node, Element)
 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	deprectated ab IE 5.x Unterstützung von DirectAnimation z.B. für 2D, 3D, Sound
.ownerDocument	Referenz auf das document Objekt zu dem der Knoten gehört, also in dem der Knoten erzeugt wurde
.parentElement	Referenz auf das Elternobjekt, also nicht Elternknoten innerhalb DOM
.parentNode	Referenz auf Elternknoten innerhalb der DOM-Hierarchie
.parentTextEdit	Textbereich des Elternobjektes referenzieren
.previousSibling	Referenz auf das Vorgängerkind
.readyState	aktueller Status des Objektes beim Füllen des Objektes mit Daten
.scopeName	Namensraum laut XMLNS-Attribut
.scrollHeight	Höhe des vertikalen Scrollbereiches, also Abstand von oberen und unteren sichtbaren Rand des Umgebungsobjektes
.scrollLeft	Abstand von linken sichtbaren Randes des Umgebungsobjektes zum linken Rand des Objektes 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
.selected	Selektionsstatus der Option, also Objektes document.select.option des Objektes document.select standardgemäß ist immer die oberste Option selektiert es kann aus einer Optionengruppe immer nur genau 1 Option selektiert sein
STYLE	direkt im HTML-Element kodierter Style (Inline-Style) Hinweis: für Scripting ist das Style-Objekt zu nutzen Achtung: Bezüglich der Style-Eigenschaften .background-color und .color kann nur genau 1 Style deklariert werden für alle Objekte document.select.option und das document.select Objektes insgesamt. Es wird der Style des ersten in der Kodierungsfolge gefundenen Objektes document.select.option für alle anderen und das Objekt document.select selbst verwendet. Es wird angeraten, nur im document.selelct das STYLE-Attribut zu verwenden. Das Objekt document.select.option kann nur genau folgende Style-Eigenschaften erhalten: .background-color und .color
.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 deprectated 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
.systemLanguage	Standard-Sprache des Betriebssystems (Sprache der Installation) per navigator Objekt
.systemOverdubOrSubtitle	wird hier nicht erklärt
.tagName	Tag-Bezeichner des Objektes
.tagUrn	Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut
.text	interner Kettenwert einer Option, , also Objektes document.select.option des Objektes document.select ändert nicht den angezeigten Text hinter <OPTION ...> ist nur ein interner Wert und beim Formular der gesendete Wert der Option
.timeContainer	Typ der Timeline des Objektes siehe Objekt currTimeState und Behavior .style.time2
.uniqueID	durch den Browser automatisch-genriertes 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
.value	Wert eines Objekt-Attributes Hinweis: Wert eines Elementes ist z.T. über das VALUE-Attribut definierbar im Formular: wenn Eigenschaft .text kodiert, so wird deren Wert gesendet und der Wert von .value bleibt nur angezeigt wenn Eigenschaft .text nicht kodiert, so wird der Wert von .value gesendet
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
.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 Endtag (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)
.clearAttributes()	alle HTML-Attribute eines Objektes entfernen außer ID, STYLE und per Script definierte Attribute Script-erzeugte Attribute nicht entfernbar
.click()	DOM wird geändert 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 Überbereich 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
.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 !!
.normalize()	DOM wird geändert 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 DOM wird geändert
.removeAttributeNode()	entfernen von Attribut, egal ob es mit oder ohne HTML-Anweisung erzeugt wurde, und Referenz auf das entfernte Attribut liefern
.removeBehavior()	DOM wird geändert per Methode .addBehavior() einem Element hinzugefügte Verhaltenseigenschaft entfernen (stets VOR dem Entfernen des Elementes mit der zugeordneten Eigenschaft aus der Dokument-Hierarchie)
.removeChild()	DOM wird geändert Kind-Objekt aus einem Objekt entfernen aus DOM und Referenz auf das entfernte Kind liefern Sichtbarkeit erst, wenn Ende-Tag geparkt wurde, also das Dokument neu geladen wurde
.removeExpression()	DOM wird geändert Ausdruck entfernen, der für die Berechnung des Wertes einer Style-Eigenschaft als Objektreferenz der Form objekt.style.eigenschaft. dient. Ausdruck muss mit der Methode .setExpression() gesetzt worden sein
.removeNode()	DOM wird nicht geändert Knoten entfernen aus DOM und Referenz auf den entfernten Knoten liefern Sichtbarkeit erst wenn Ende-Tag geparkt wurde
.replaceAdjacentText()	DOM wird geändert Plain-Text (ohne HTML und Script) eines Elementes durch anderen Text ersetzen und Referenz auf den zu ersetzenden Text liefern
.replaceChild()	DOM wird nicht geändert Kind-Objekt ersetzen durch ein Objekt ersetzende Objekt muss per Methode .createElement() erzeugt worden sein Sichtbarkeit erst wenn Ende-Tag geparkt wurde
.replaceNode()	DOM wird geändert Objekt durch anderes Objekt komplett ersetzen und Referenz auf das komplett ersetzte Objekt liefern sichtbar erst mit parsen des Endetags
.setAttribute()	DOM wird geändert Wert von vorhandenem Attribut setzen wenn Attribut nicht vorhanden, so wird es automatisch erzeugt und mit dem Wert gefüllt
.setAttributeNode()	DOM wird nur bei Erzeugung geändert Attribut einem Knoten zuweisen und Referenz liefern
.setCapture()	DOM wird geändert Maus-Überwachung einschalten für ein Objekt Maus-Events sind : onmousedown, onmouseup, onmousemove, onclick, ondblclick, onmouseover und onmouseout. ab IE 5.5
.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

