

attributes Collection des Internet Explorer

Diese Collection sammelt die Zeiger aller Attribute eines HTML-Elementes

Feld der Objekt-Attribute-Referenzen

aber nicht Style-Attribute-Werte des Objektes, da diese in der Eigenschaft `.cssText` abgeleitet sind (ab IE 5.x)

Feld muss mit den Methoden extra gepflegt werden, da dies nicht automatisch mit der Modifikation von Attributen vollzogen wird.

Feldelement: ist attribute Objekt (siehe dort)

Syntax:

```
[ var FeldZeiger = ] object.attributes
[ var FeldElementZeiger = ] object.attributes[Index]
```

Index: Integer und ab 0
muss in [] kodiert sein

Beispiel 1:

```
<SCRIPT>
function ShowAttribs(ZeigerAufObjekt)
{
    var ZeigerAufFeld = ZeigerAufObjekt.attributes;

    for (var Index = 0; Index < ZeigerAufFeld.length; Index ++)
    {
        var ZeigerAufFeldElement = ZeigerAufFeld [Index];

        alert(
            ZeigerAufFeldElement.nodeName
            + '='
            + ZeigerAufFeldElement.nodeValue
            + '('
            + ZeigerAufFeldElement.specified
            + ')'
        );
    }
}
</SCRIPT>
```

Beispiel 2:

```
<SCRIPT>
function Anzeigen()
{
    for(var Index=0; Index < ID_Liste.attributes.length; Index ++)
    {
        if(ID_Liste.attributes[Index].specified)
        {
            alert(
                ID_Liste.attributes[Index].nodeName
                + " = "
                + ID_Liste.attributes[Index].nodeValue
            );
        }
    }
}
</SCRIPT>
<UL onclick="Anzeigen()">
    <LI ID = "ID_Liste" ACCESSKEY = "L">Listen-Element
</UL>
```

Beispiel 3:

```
<HTML>
<HEAD>
<SCRIPT>
    document.expando = false; // für gesamtes Dokument abschalten
</SCRIPT>
</HEAD>
<BODY>
<DIV>
    <SPAN ID="ID_Span" UNSELECTABLE="on">
        Dieser Text ist <B>selektierbar !!<B>
    </SPAN>
</DIV>
</BODY>
</HTML>
```

Beispiel für option objekt eines select objektes:

```
<SCRIPT>
```



```

function Anzeigen()
{
    var Kette = "Auswahl = " + ID_select.options(ID_select.selectedIndex).value;
    alert(Kette);
}
</SCRIPT>
<SELECT ID="ID_select" onchange = "Anzeigen()">
  <OPTION VALUE="1">Auswahl 1 </OPTION>
  <OPTION VALUE="2">Auswahl 2 </OPTION>
  <OPTION VALUE="3">Auswahl 3 </OPTION>
</SELECT>

```

Eigenschaften:

.expando Wirksamkeit von Attributen ein/aus
 true ein, Default
 false aus

Beispiel:

```

<HTML>
<HEAD>
<SCRIPT>
  document.expando = false; // für gesamtes Dokument abschalten
</SCRIPT>
</HEAD>
<BODY>
<DIV>
  <SPAN ID="ID_Span" UNSELECTABLE="on">
    Dieser Text ist <B>selektierbar !!<B>
  </SPAN>
</DIV>
</BODY>
</HTML>

```

.length Anzahl der Feldelemente also Feldlänge
 .specified prüfen ob Objekt Attribute hat
 true, so Attribute ist vorhanden
 false, so keine Attribute vorhanden

Beispiel:

```

<SCRIPT>
function Anzeigen()
{
    var FeldAllerAttribute = ID_List.attributes;
    alert(FeldAllerAttribute(0).nodeName); // Knotenname

    for( var i=0; i< FeldAllerAttribute.length; i++)
    {
        // neues LI-Element als Knoten der Liste anhängen
        var NeuesListenElement =document.createElement("LI");
        ID_List.appendChild(NeuesListenElement);

        // Wert des neuen LI-Elementes ist Text, also Textknoten
        var WertNeuesListenElement = document.createTextNode(
            i
            + " "
            + FeldAllerAttribute (i).nodeName
            + " = "
            + FeldAllerAttribute (i).nodeValue
        );
        NeuesListenElement.appendChild(WertNeuesListenElement);

        // auf specified prüfen
        if(FeldAllerAttribute (i).nodeValue != null )
        {
            alert(
                FeldAllerAttribute(i).nodeName
                + " specified: "
                + FeldAllerAttribute(i).specified // boolean
            );
        }
    }
}
</SCRIPT>
<UL ID="ID_List" onclick = " Anzeigen()">
  <LI>Klick mich
</UL>

```

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

Beispiel



```

<SCRIPT>
function Anzeigen()
{
    var Kette = "Auswahl = " + ID_select.options(ID_select.selectedIndex).value;
    alert(Kette);
}
</SCRIPT>
<SELECT ID="ID_select" onchange = "Anzeigen()">
<OPTION VALUE="1">Auswahl 1 </OPTION>
<OPTION VALUE="2">Auswahl 2 </OPTION>
<OPTION VALUE="3">Auswahl 3 </OPTION>
</SELECT>

```

Methoden:

.getItem() Zeiger auf Attribut liefern anhand des Attributnamen (analog zu ID oder NAME-Attribut) ab IE 6.x

Beispiel:

```

<HTML>
<HEAD>
<SCRIPT>
function Init()
{
    var ZeigerAufFeld = ID_P.attributes;
    var ZeigerAufFeldElement = ZeigerAufFeld.getItem("align");
    alert("ALIGN Attribut Wert = " + ZeigerAufFeldElement.value);
}
</SCRIPT>
</HEAD>
<BODY ONLOAD="Init()">
<P ID="ID_P" ALIGN="center">Test</P>
</BODY>
</HTML>

```

.item() Referenz auf Feldelement anhand des Integer-Indexes oder des Attributnamen (analog zu ID oder NAME-Attribut) liefern außer bei Formular mit <INPUT TYPE=image ...> da dafür die children-Collection verwendet werden muss !!!

Beispiel 1:

```

<SCRIPT LANGUAGE="JScript">
var ZeigerAufCollectionDocumentAll = document.all;

if (ZeigerAufObjekt!=null)
{
    for (i=0; i< ZeigerAufCollectionDocumentAll.length; i++)
    {alert(ZeigerAufCollectionDocumentAll.item(i).tagName);}
}
</SCRIPT>

```

Beispiel 2:

```

<HTML>
<HEAD>
<SCRIPT>
function Init()
{
    var ZeigerAufFeld = ID_P.attributes;
    for (Index = 0; Index < ZeigerAufFeld.length; Index ++)
    {
        var ZeigerAufFeldElement = ZeigerAufFeld.item(Index);
        // hier numerischer Index
        var AttributWertSpezifiziert = ZeigerAufFeldElement.specified;
        // true oder false
        var KnotenName = ZeigerAufFeldElement.nodeName;
        // String
        var KnotenWert = ZeigerAufFeldElement.nodeValue;
        alert(
            "Knotenname = "
            + KnotenName
            + " mit spezifiziert = "
            + AttributWertSpezifiziert
            + " und Wert = "
            + KnotenWert
        );
    }
}
</SCRIPT>
</HEAD>
<BODY ONLOAD="Init()">

```



```

        <P ID="ID_P">Test</P>
    </BODY>
</HTML>

```

`.removeNamedItem()` Attribut entfernen anhand Attributname (analog zu ID oder NAME-Attribut), wobei danach der Standard-Attributwert automatisch weiterverwendet wird (falls Standard vorhanden ist), und Zeiger auf gelöscht Attribut liefern
ab IE 6

Beispiel:

```

<HTML>
<HEAD>
<SCRIPT>
    function Entfernen()
    {
        var ZeigerAufFeld = ID_Div.attributes;
        ZeigerAufFeld.removeNamedItem("TITLE");
    }
</SCRIPT>
</HEAD>
<BODY>
    <DIV onclick="Entfernen();" ID="ID_Div" TITLE="Tooltip-Text ">
        Klick um den Tooltip-Text zu entfernen
    </DIV>
</BODY>
</HTML>

```

`.setNamedItem()` Attribut hinzufügen anhand Zeiger auf Attribut wenn noch nicht im Feld vorhanden, so Anhängen an das Feldende wenn schon im Feld vorhanden, so überschreiben und Referenz auf das überschriebene Attribut (vor dem Überschreiben) liefern
Bsp: Attribut erzeugen `document.createAttribute("title");`
Hinweis: in HTML können Attributnamen groß oder klein geschrieben werden
ab IE 6

Beispiel:

```

<HTML>
<HEAD>
<SCRIPT>
    function Hinzufuegen()
    {
        // Attribut mit Wert erzeugen
        var ZeigerAufAttribut = document.createAttribute("title");
        ZeigerAufAttribut.value = "Tooltip-Text ";

        // Attribut als Feldelement anhängen
        var ZeigerAufFeld = ID_Div.attributes;
        ZeigerAufFeld.setNamedItem(ZeigerAufAttribut);
    }
</SCRIPT>
</HEAD>
<BODY onload="Hinzufuegen();">
    <DIV ID="ID_Div">Es wird ein Tooltip-Text hinzugefuegt</DIV>
</BODY>
</HTML>

```

