

childNodes Collection des Internet Explorer

Feld der Zeiger aller Kinder-Knoten eines Objektes, also Feld der Zeiger aller HTML-Knoten-Kinder **und** Textknoten-Kinder des Objektes Collection dient zur Ermittlung **nur von Knoteneigenschaften laut DOM**, falls diese im Element implementiert sind:

Elementefolge NICHT laut HTML-Coding sondern laut DOM.

Element (Knoten) kann per HTML oder Methode .createElement() erzeugt worden sein (falls Methode erlaubt ist). Diese Collection sammelt die Zeiger aller Kinder**knoten** eines HTML-Elementes.

Syntax:

```
[ var ZeigerAufFeld = ] object.childNodes
[ var ZeigerAufFeldElement = ] object.childNodes[Index]

object                Zeiger auf Elternobjekt

Index                Integer und ab 0
                    muss in [ ] kodiert sein

ZeigerAufFeldElement ist null, wenn Feldelement nicht vorhanden
```

Zugriff auf Element:

Je nach Art des Kind-Elementes stehen dem Kind Eigenschaften und Methoden zur Verfügung (siehe Objektbeschreibungen):

```
object.childNodes[Index].eigenschaft_des_kind
object.childNodes[Index].methode_des_kind
```

```
object    Zeiger auf Elternobjekt

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

Beispiel 1:

```
<SCRIPT>
    var ZeigerAufFeld = ID_Body.childNodes;
</SCRIPT>
<BODY ID="ID_Body">
    <SPAN >Test </SPAN>
</BODY>
```

Beispiel 2:

```
// DIV erzeugen
var ZeigerAufDivKnoten = document.createElement("DIV");

// B-Tag im DIV erzeugen
var ZeigerAufBKnoten = document.createElement("B");
ZeigerAufDivKnoten.insertBefore(ZeigerAufBKnoten);

// erst jetzt das DIV in den BODY einfügen, also DIV sichtbar machen
document.body.insertBefore(ZeigerAufDivKnoten);

// Collection referenzieren
var ZeigerAufFeld = ZeigerAufDivKnoten.childNodes;
```

Beispiel 3:

```
<SCRIPT LANGUAGE="JScript">
    var ZeigerAufFeldAllerPtag = document.all.tags("P");
    if (ZeigerAufFeldAllerPtag!=null)
    {
        for (i=0; i< ZeigerAufFeldAllerPtag.length; i++)
        { ZeigerAufFeldAllerPtag [i].style.textDecoration="underline"; } // auch .item(i) kodierbar
    }
</SCRIPT>
```

Beispiel 4:

```
<SCRIPT>
    var ErstesKind_Index = 0; // Index ab 0
    var ErstesKind_Name = Liste.childNodes(ErstesKind_Index).nodeName;
    alert(ErstesKind_Name); // liefert den Tagnamen 'LI' von Listenelement 1
    // Hinweis: Listenelement 1 ist der Wert des Kindes
</SCRIPT>
<BODY>
    <UL ID = "Liste">
        <LI> Listenelement 1
        <LI> Listenelement 2
        <LI> Listenelement 3
    </UL>
```



<BODY>

Beispiel 5:

```

<SCRIPT>
function KnotenWertAendern( ZeigerAufListe,
                           IndexVonListenelement, // immer ab 0
                           Zeichenkette           // Listenelement muss Text sein
                           )
{
    var ReturnWert=false; // Annahme: Änderung schlägt fehl

    // prüfen auf UL-Tag
    if (ZeigerAufListe.nodeName == "UL")
    {
        // Anzahl der Listenelemente holen
        var AnzahlListenelemente= ZeigerAufListe.childNodes.length;
                                                // immer ab 1

        // und Anzahl und Index prüfen
        if ( (AnzahlListenelemente > 0) // immer ab 1
            && (IndexVonListenelement >= 0) // immer ab 0
            && (IndexVonListenelement < AnzahlListenelemente)
                                                // Index ist zulässig zur Anzahl
            )
        {
            // Zeiger auf das Listenelement laut Index holen
            var ZeigerAufListenelement =
                ZeigerAufListe.childNodes[IndexVonListenelement];

            // existiert das Listenelement ?
            if (ZeigerAufListenelement)
            {
                // ZeigerAufListenelement ist nicht null

                // Listenelement ist Textelement ?
                if (ZeigerAufListenelement.nodeType == 3)
                {
                    ZeigerAufListenelement.nodeValue =
                        Zeichenkette;
                    ReturnWert =true;
                }
            }
        }
    }

    return ReturnWert;
}
</SCRIPT>
<UL ID="Liste" onclick=" KnotenWertAendern(this, 0, 'Listenelement Neu')">
    <LI>Listenelement alt
</UL>

```

Beispiel 6:

```

<SCRIPT>
function TextElementAendern()
{
    var ZeigerAufTextElement = document.createTextNode("Neuer Text");
    var ZeigerAufSpanInhalt = ID_Span.childNodes(0);
    ZeigerAufSpanInhalt.replaceNode(ZeigerAufTextElement);
}
</SCRIPT>
<SPAN ID = "ID_Span" onclick=" TextElementAendern()">
    Original Text
</SPAN>

```

Beispiel 7:

```

<SCRIPT>
function Anzeige()
{
    var ZeigerAufOnClickEventQuelle=event.srcElement;
    var Feld =
        ZeigerAufOnClickEventQuelle.parentElement.getElementsByTagName("LI");
    alert(
        "Anzahl LI : "
        + Feld.length
        + "\nErster Eintrag: "
    );
}

```



```

        + Feld [0].childNodes[0].nodeValue
    );
}
</SCRIPT>
<UL onclick="Anzeige()">
  <LI>Menuepunkt 1
  <UL>
    <LI> Menuepunkt 1.1
    <OL>
      <LI> Menuepunkt 1 1.1
      <LI> Menuepunkt 1 1.2
    </OL>
    <LI> Menuepunkt 1.2
    <LI> Menuepunkt 1.3
  </UL>
<LI> Menuepunkt 2
<UL>
  <LI> Menuepunkt 2.1
  <LI> Menuepunkt 2.3
</UL>
<LI> Menuepunkt 3
</UL>

```

Eigenschaften:

.length

Methoden:

.item()

Anzahl der Feldelemente also Feldlänge

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>

```

.urns()

Referenz auf Feld aller Elemente mit gemeinsamer URN liefern

Beispiel:

```

<SCRIPT LANGUAGE="JScript">

```



```
var ZeigerAufFeldAllerURN1 coll = document.all.urns("URN1");
var Text = "";

if (ZeigerAufFeldAllerURN1 != null)
{
    for (i=0; i< ZeigerAufFeldAllerURN1.length; i++)
    {Text += ZeigerAufFeldAllerURN1.item(i).id + ' ';}
    alert (Text);
}
</SCRIPT>
```

