

noFrames Objekt des Internet Explorer

Container für denjenigen Code der Homepage, der nur durch Browser abgearbeitet wird, die FRAMESET-Tag nicht kennen.

Beispiel:

```
<FRAMESET>
  <NOFRAMES>
    Ihr Browser kennt das FRAMESET-Tag nicht
  </NOFRAMES>
</FRAMESET>
```

Zugriff auf noFrames-Eigenschaft:

Beispiel

```
var Wert = document.all.ZeigerAufnoFrames.eigenschaft;
```

mit ZeigerAufFrameset laut ID-Attribut

```
<NOFRAMES ID="ZeigerAufnoFrames" ..... >
```

Eigenschaften:

.canHaveHTML
.id

prüfen ob Objekt HTML-Tags enthalten darf
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

.parentElement

Referenz auf das Elternobjekt, also nicht Elternknoten innerhalb DOM

.readyState

aktueller Status des Objektes beim Füllen des Objektes mit Daten

.scopeName

Namensraum laut XMLNS-Attribut

.tagUrn

Uniform Resource Name (URN) laut Namensraum laut XMLNS-Attribut

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

.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.

.fireEvent()

ein Event auslösen

.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

.normalize()

Normalisierung des DOM zur Erreichung einer konsistenten Struktur
Achtung: CDATA-Sections dürfen nicht enthalten sein, da diese immer Inkonsistenz erzeugen
entfernen eines per HTML erzeugten Attributes

.removeAttribute()

Achtung: Der Browser unterscheidet zwischen HTML-erzeugte oder mit dieser Methode erzeugte Attribute!
per Methode .createAttribute() erzeugte Attribute werden nicht erfasst
DOM wird geändert

.removeBehavior()

per Methode .addBehavior() einem Element hinzugefügte Verhaltenseigenschaft entfernen (stets VOR dem Entfernen des Elementes mit der zugeordneten Eigenschaft aus der Dokument-Hierarchie)

.setAttributeNode()

DOM wird geändert
Attribut einem Knoten zuweisen und Referenz liefern
DOM wird geändert

