

Wichtige Javascript Befehle

Message Box

- **alert**(„[Text der ausgegeben werden soll]“); //nur Ausgabe über Message-Box
 - alert(„Willkommen“);
- **confirm**(„[Text der ausgegeben werden soll]“); //gibt true/false zurück
- var [Variablenname] = **prompt**(„[Text der ausgegeben werden soll]“, „[Anfangswert]“);
 - z.B.: var name = prompt(„Bitte geben sie ihren Namen ein: “, „“);
 - Umwandlung von eingelesenen Wert in Integer
 - z.B.: **parseInt**(prompt(„Geben Sie eine Zahl ein: “, „“), **10**);

<body onload=“javascript:[Funktion, die beim Starten des Browsers geöffnet werden soll]()“>

<body onunload=“javascript:[Funktion, die beim Schließen des Browsers geöffnet werden soll]()“>

document. ... kann man auf alle Element im Hauptfenster zugreifen

document.write Ausgabe im Browser-Hauptfenster

window.document.backgroundColor = „yellow“ Hintergrundfarbe im Browser-Hauptfenster

window.status = „[Text der ausgegeben werden soll]“ Wird in der Statusleiste angezeigt

window.open() Wird neues Fenster geöffnet

window.close() Wird Fenster geschlossen

Aufruf: z.B.: onclick=“window.close()“

document.images. ... kann man auf die Eigenschaften von den eingefügten Bilder zugreifen

z.B.:
`<img src=„images/blau.jpg“ height=“300“ width=“300“
onclick=“openImage('bild1')“ name=“bild1“>
<script>
var h = document.images.bild1.height;`

document.images[index] eingefügten Bilder werden in einem Array gespeichert und können somit über den Index oder den Namen aufgerufen werden

document.images.[Name des Bildes] Aufruf über den Namen des Bildes

document.images[index].complete Status, ob Bild fertig geladen ist

.scr Gibt den Source-string des Bildes zurück

.length gibt die Anzahl der eingefügten Bilder zurück

.height gibt die Höhe des Bildes zurück oder kann geändert werden

.width gibt die Breite des Bildes zurück oder kann geändert werden

document.title kann der Titel geändert oder gelesen werden

Aufbau einer Funktion

function [Name der Funktion] ([übergebene Parameter])

```
{  
...  
}
```

Aufruf über den Namen der Funktion

Rückgabe mit **return**

Einfügen eines Bildes:

``

Muss nur der Source und der Name angegeben werden; die anderen Parameter sind optional einstellbar

Einfügen einer Tabelle (am Einfachsten mit der Palette (Funktion im Netbeans rechts oben))

```
<table border="[Breite des Rahmens]">
  <tbody>
    <tr> //Zeile der Tabelle
      <td>Kästchen der Tabelle
    </td>
  </tr>
</tbody>
</table>
```

Einfügen eines Form-Tags

```
<form name="[Name der Form]"
  onSubmit="return[=empfängt true/false] „[Funktion die Aufgerufen wird wenn der Submit-
  Button gedrückt wird]();"
  onReset="return[=empfängt true/false] „[Funktion die Aufgerufen wird wenn der Submit-
  Button gedrückt wird]();">

  //onSubmit, onReset ist optional
  //Aufruf der enthalten Element (Radiobuttons, Textboxes,...) mit dem [Namen des
  Form-Tags].[Name des Elementes]. ...
</form>
```

Einfügen eines Submit Buttons

```
<input type="submit" value="[Übergabewert]">
```

Einfügen eines Reset Buttons

```
<input type="reset" value="[Übergabewert]">
```

```
</form>
```

Einfügen einer Textbox

```
<input type="text" name="[Name der Textbox]" size="[Größe der Textbox]" value="[Anfangswert]"
readonly> //Muss „name“ enthalten, alles andere ist optional
```

Hineinsetzen des Cursors mit

```
[Name].focus()
```

Einfügen eines Radio-Buttons

```
<input type="radio" name="[Name des Radio-Buttons]" value="[Wert des Radio-Buttons]"> [Text  
der neben dem Radio Button stehen soll] />
```

- Bei eine **Gruppe** von Radio-Buttons muss der **Name bei allen gleich** sein
- kann dann über den value auf die einzelnen zugreifen (über Index, wie bei Arrays, oder über den Namen

Einfügen einer Checkbox

`<input type="checkbox" name="[Name der Checkbox]" value="[Wert der Checkbox]"> [Text der neben der Checkbox stehen soll] />`

- Bei eine **Gruppe** von Checkboxes muss der **Name bei allen gleich** sein
- kann dann über den value auf die einzelnen zugreifen (über Index, wie bei Arrays, oder über den Namen

Einfügen einer Textarea (=mehrzeilige Textbox)

`<textarea name="[Name]" cols="[Breite]" rows="[Zeilenanzahl]" maxlength="[Maximale Anzahl von Zeichen]" onkeyup="[Funktion die aufgerufen werden soll bei keyup]();" onblur="[Funktion die aufgerufen werden soll wenn der Cursor die Textarea verlässt]();"`

`</textarea>`

Hineinsetzen des Cursors mit

`[Name].focus()`

Einfügen eines Buttons

`<input type="[Name]" onclick="[Funktion die ausgeführt wird]()" value="[Wert]">[Text im Button]`
oder

`<button type="[Name]" onclick="[Funktion die ausgeführt wird]()" value="[Wert]">[Text im Button]</button>`

Einfügen einer Combobox

`<select name="[Name]" onchange="[Funktion, die ausgeführt wird wenn ein anderes Element ausgewählt wird]();">`

`<option value="[Wert]">^[Wert]</option>[Auswahlmöglichkeiten]`

`<option value="[Wert]">^[Wert]</option>[Auswahlmöglichkeiten]`

`<option value="[Wert]">^[Wert]</option>[Auswahlmöglichkeiten]`

...

`</select>`

sonstige Befehle

Array erstellen

var [Arrayname] = new Array([Länge des Arrays])
[Arrayname].charAt(index) Zeichen des aktuellen Indexes abfragen

Zufallszahl generieren

- Math.floor(Math.random()); //generiert eine Zahl zwischen 0 und 1
- um eine ganze Zufallszahl zu erhalten
 - Math.floor(Math.random()*10) // generiert eine ganze Zahl zwischen 0 und 10

Systemzeit:

var zeit = new Date();
zeit.getHours(); gibt die aktuelle Stunde zurück
zeit.getMinutes(); gibt die aktuelle Minute zurück
zeit.getSeconds(); gibt die aktuelle Sekunde zurück

Externe Datei in Script einbinden

```
<script type="text/javascript" src="cookie.js"></script>
```

Text verlinken

```
<a href="[Wo der Link hinzeigt]">[Text der verlinkt werden soll] </a>
```

Timer festlegen

setTimeout(„[Funktion die im Intervall ausgeführt werden soll]“, [Intervallzeit]);

Ausrichtung im Dokument ändern

```
<div align="[Ausrichtung z.B.:center]"> </div>
```