

# Digitale Medien

Übung

# JavaScript

Browser-Scriptsprache,  
ursprünglich von Netscape entwickelt.

- wird direkt im Browser ausgeführt
- ermöglicht Manipulation von Webseiten, Reaktion auf Nutzereingaben, etc.

Wird u.a. zur Realisierung von dynamischen Webseiten genutzt (DHTML, AJAX).

Beispiele:

<http://docs.google.com>

<http://www.elizium.nu/scripts/lemmings/>

Für die folgenden Aufgaben:

1. HTML-Grundgerüst im Texteditor erstellen  
(siehe hier: <http://de.selfhtml.org/html/allgemein/grundgeruest.htm>)
2. Als HTML-Datei abspeichern (z.B. "index.html")
3. Im Browser (Firefox) öffnen

Dann:

Jeweils eine Änderung in der HTML-Datei durchführen, abspeichern und im Browser anschauen (Refresh: 'F5')

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>

## Einbindung von Javascript in HTML:

### 1. Direkt in einem HTML-Tag:

```
<p onclick="alert('klick, klick')">Klick mich!</p>
```

```
<body onload="alert('hi!')">
```

### 2. Im Header:

```
<head>  
  <script type="text/javascript">  
    alert('klick klick');  
  </script>  
</head>
```

### 3. In einer eigenen Datei:

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

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>

## Arithmetische Operationen:

```
alert(4 + 4);
```

## Variablen:

- keine strikte Typisierung wie in Java
- keine Deklaration notwendig, Variablen können einfach benutzt werden

```
blah = 4;  
unblah = (blah * blah) + 4;  
alert(unblah);
```

## Bedingte Anweisungen:

```
zahl = 4;
if(zahl > 5){
    alert("größer.");
} else if(zahl < 5){
    alert("kleiner.");
} else {
    alert("gleich.");
}
```

## Schleifen:

```
zaehler = 1;
while(zaehler != 10){
    document.write("<p>" + zaehler + "</p>");
    zaehler++;
}
```

## Ausgabe per Pop-Up:

```
alert("nerv!");
```

## Eingabe per Pop-Up:

```
ergebnis = window.prompt("Was denke ich wohl gerade?");
```

## Funktionen:

```
function pluszehn(x){  
    return x + 10;  
}
```

...

```
<p onclick="alert(pluszehn(10))">  
    Zehn plus zehn ist ...?  
</p>
```

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>

## Aufgabe:

Browser soll eine Zufallszahl zwischen 1 und 10 generieren und der Benutzer soll solange Zahlen eingeben, bis er sie erraten hat.

Generierung einer ganzen Zufallszahl von 1 bis einschließlich x:

```
zufall = 1 + (x - 1) * (Math.random());  
zufall = Math.round(zufall);
```

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>



```
<script type="text/javascript">
    function zufall(){
        zahl = 1 + 9 * (Math.random());
        return Math.round(zahl);
    }

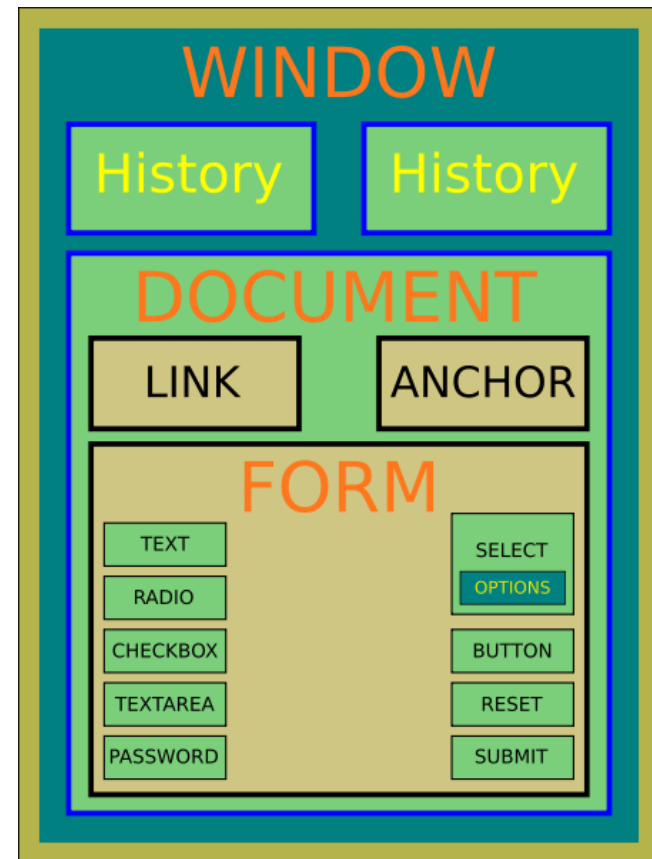
    function guess(){
        zahl = zufall();
        rate = -1;
        while(rate != zahl){
            rate = window.prompt();
        }
        alert("richtig! die zahl war " + zahl);
    }
</script>
</head>

<body onload="guess();">
```

DOM (Document Object Model):

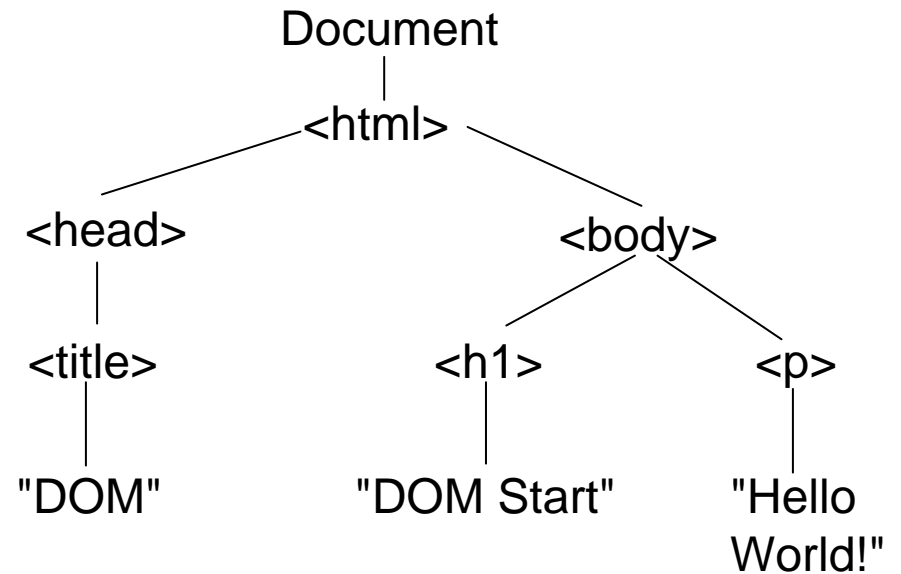
Vereinheitlichte, hierarchische Darstellung eines HTML- oder XML-Dokuments.

Jedes Dokument besteht aus verschachtelten Knoten (HTML-Elemente, Text), die miteinander verknüpft sind.



```
<html>
  <head>
    <title>DOM</title>
  </head>

  <body>
    <h1 id="head1">DOM Start</h1>
    <p id="par1">Hello world!</p>
  </body>
</html>
```



```
<html>
  <head>
    <title>DOM</title>
  </head>

  <body>
    <h1 id="head1">DOM Start</h1>
    <p id="par1">Hello world!</p>
  </body>
</html>
```

Zugriff auf ein Element mit `document.getElementById(id)`, Zugriff auf Unterknoten via `firstChild` bzw. `childNodes[x]`

```
alert(document.getElementById("head1"));
-> [object HTMLHeadingElement]
```

```
alert(document.getElementById("head1").firstChild.nodeValue);
alert(document.getElementById("head1").childNodes[0].nodeValue);
-> DOM Start
```

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>

## Aufgabe:

Veränderung der Schriftfarbe eines Textes von schwarz nach rot sobald der Benutzer den Text anklickt. Wird der rote Text nochmal angeklickt, wird die Farbe wieder auf schwarz gesetzt.

Zugriff auf ein Element mit `document.getElementById(id)`

Zugriff auf style-Informationen mit `[Knoten].style.[Attributname]`:

```
document.getElementById("headline1").style.fontWeight = "bold";
```

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>

## Aufgabe:

Veränderung der Schriftfarbe eines Textes von schwarz nach rot sobald der Benutzer den Text anklickt. Wird der rote Text nochmal angeklickt, wird die Farbe wieder auf schwarz gesetzt.

```
var red = false;
function changecol(){
  if(red == false){
    red = true;
    document.getElementById("par1").style.color = "#FF0000";
  } else {
    red = false;
    document.getElementById("par1").style.color = "#000000";
  }
}
```

...

```
<p id="par1" onclick="changecol()">Licht an!</p>
```

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>

## Aufgabe:

Erstellung eines Rollover-Buttons, d.h. ein Bild, das sich verändert, sobald der Benutzer die Maus darüber bewegt.

Zugriff auf ein Element mit `document.getElementById(id)`

Event-Handler für Mausbewegungen und Objekte:

`onmouseover`

`onmouseout`

Pfad eines Bildes = `[Bild-Knoten].src`

Javascript-Übersicht:

<http://de.selfhtml.org/javascript/index.htm>

## Aufgabe:

Erstellung eines Rollover-Buttons, d.h. ein Bild, das sich verändert, sobald der Benutzer die Maus darüber bewegt.

```
function changeme(){  
    document.getElementById("bild").src = "bild2.jpg";  
}
```

```
function changemeback(){  
    document.getElementById("bild").src = "bild1.jpg";  
}
```

```
...  

```