

Übung zur Vorlesung
Digitale Medien

Dr. Alexander De Luca
Ludwig-Maximilians-Universität München
Wintersemester 2013/2014

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://mrdoob.com/projects/chromeexperiments/ball-pool/>

JavaScript

Für die folgenden Aufgaben:

1. HTML-Grundgerüst im Texteditor erstellen.
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

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>
```

Javascript-Übersicht:

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

Einbindung

Einbindung von Javascript in HTML:

3. In einer eigenen Datei:

```
<head>
  <script src="klicker.js" type="text/javascript" />
</head>
<body onload="klick()">
</body>
```

klicker.js :

```
function klick() {
  alert("klick, klick");
}
```

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

Javascript

Arithmetische Operationen:

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

Variablen:

- keine strikte Typisierung wie in Java
- bei der Deklaration müssen keine Datentypen angegeben werden

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

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

Javascript

Bedingte Anweisungen:

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

Schleifen:

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

Javascript-Übersicht:

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

Javascript

Ausgabe per Pop-Up:

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

Eingabe per Pop-Up:

```
var 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>

Javascript

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:

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

Javascript-Übersicht:

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

Javascript

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

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

Javascript-Übersicht:

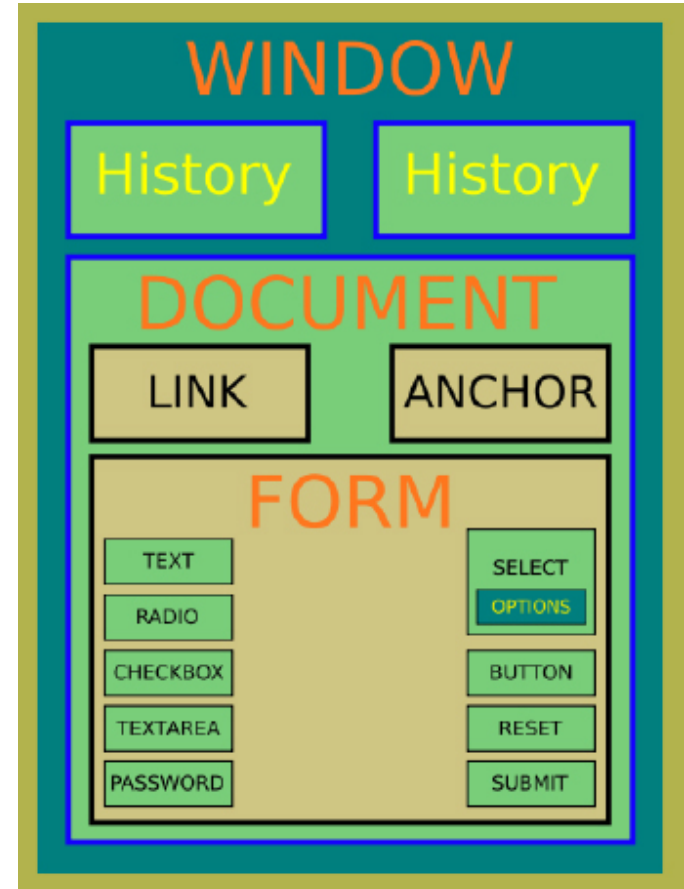
<http://de.selfhtml.org/javascript/index.htm>

DOM

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.

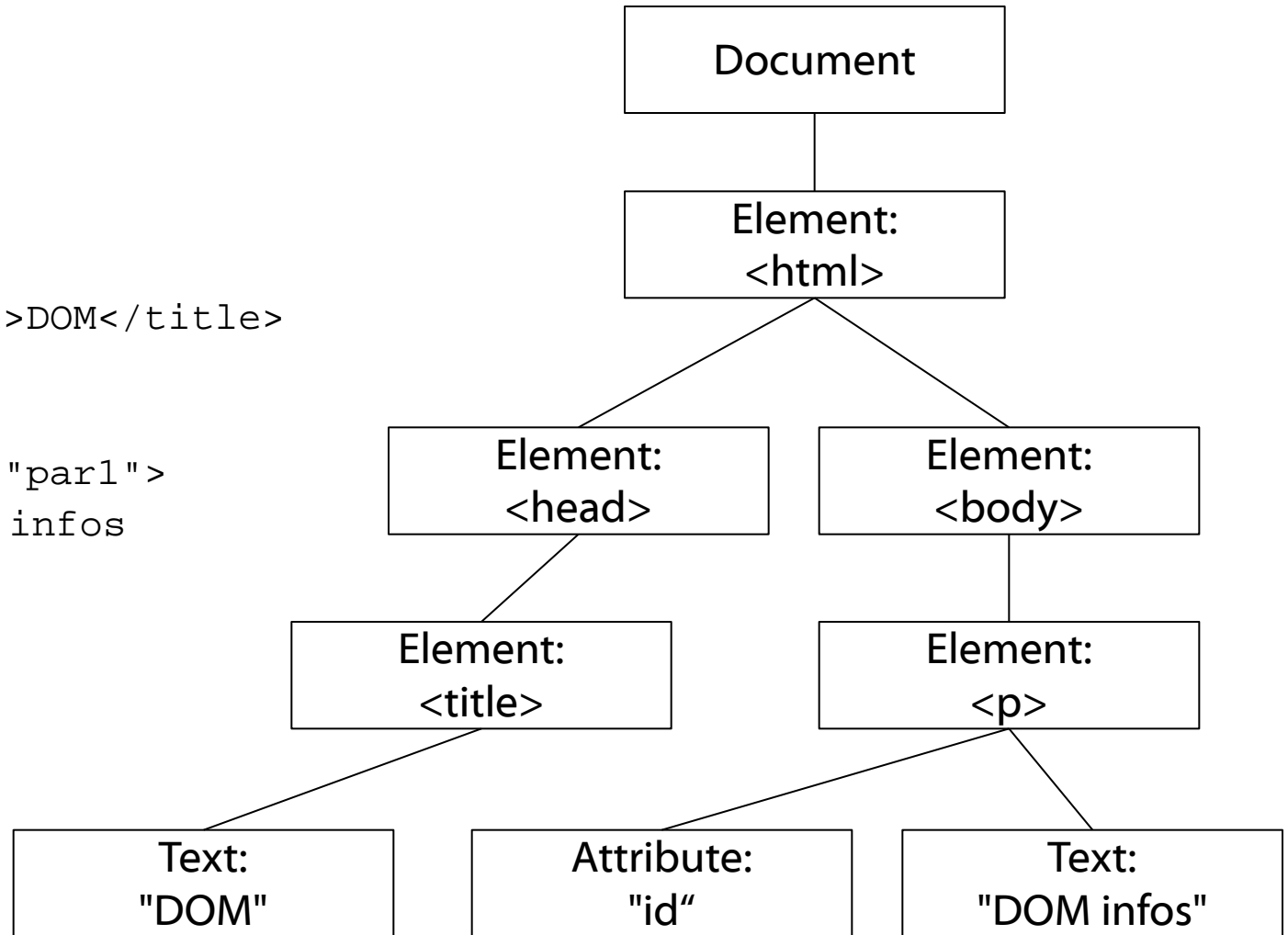


Javascript-Übersicht:

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

DOM

```
<html>  
  <head>  
    <title>DOM</title>  
  </head>  
  <body>  
    <p id="par1">  
      DOM infos  
    </p>  
  </body>  
</html>
```



Javascript-Übersicht:

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

DOM & Javascript

```
<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

DOM & Javascript

Hinzufügen von Event-Handlern:

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

veraltet!

entspricht:

```
<p id="par">  
    ...  
<script>  
document.getElementById("par")  
    .addEventListener("click",alertplus10);  
  
function alertplus10(){  
    alert(pluszehn(10));  
}  
</script>
```

Javascript-Übersicht:

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

DOM & Javascript

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("head1").style.fontWeight = "bold";
```

Javascript-Übersicht:

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

DOM & Javascript

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.

```
<p id="par1">Licht an!</p>
```

```
...
```

```
document.getElementById("par1")  
    .addEventListener("click",changeacol);
```

```
var red = false;
```

```
function changeacol(){
```

```
    if(red == false){
```

```
        red = true;
```

```
        document.getElementById("par1").style.color = "#FF0000";
```

```
    } else {
```

```
        red = false;
```

```
        document.getElementById("par1").style.color = "#000000";
```

```
    }
```

```
}
```


DOM & Javascript

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)`

Events für Mausbewegungen und Objekte:

`mouseover`

`mouseout`

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

Javascript-Übersicht:

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

DOM & Javascript

Aufgabe:

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

```

...
document.getElementById("bild")
    .addEventListener("mouseover", changeme);
document.getElementById("bild")
    .addEventListener("mouseout", changemeback);

function changeme(){
    document.getElementById("bild").src = "bild2.jpg";
}
function changemeback(){
    document.getElementById("bild").src = "bild1.jpg";
}
```

Javascript-Übersicht:

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

Formulare in HTML

Formulare definieren:

```
<form action="URI" method="Methode">  
...  
</form>
```

URI ist die Adresse der Datei die die Daten verarbeiten soll
method entweder get oder post

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

Formulare in HTML

Wichtige Formularelemente:

Einzeiliges Textfeld:

```
<input type="text">
```

Auswahlliste:

```
<select>  
  <option value="1" selected>Option 1</option>  
  <option value="2">Option 2</option>  
</select>
```

Button:

```
<input type="button" name="Name" value="Beschriftung">
```

Javascript-Übersicht:

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

Formulare

Aufgabe:

Erstellen Sie ein Formular mit einem Button mit der Aufschrift „Neues Element“ und darunter ein leeres Div-Element mit der id „container“.

```
<form action="http://www.example.org/cgi-bin/feedback.pl"
method="post">

    <input type="button" value="Neues Element">
</form>
<div id="container"></div>
```

Javascript-Übersicht:

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

Formulare

Aufgabe:

Fügen Sie bei jedem Klick auf den Button dem Div-Element einen Textparagraph hinzu, der den Text „Bla“ enthält:

Neues Element erstellen mit: `document.createElement(element-type)`

Neuen Textknoten erstellen mit: `document.createTextNode("text")`

Neues Element einfügen mit:

```
document[.getElementById(id)].appendChild(item);
```

Javascript-Übersicht:

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

Formulare

Aufgabe:

Fügen Sie bei jedem Klick auf den Button dem dem Div-Element einen Textparagraph hinzu, der den Text „Bla“ enthält:

...

```
<input type="button" value="Neues Element" id="butt">
```

...

```
document.getElementById("butt")  
    .addEventListener("click", createP);
```

```
function createP() {  
    var paragraph = document.createElement("p");  
    document.getElementById("container").appendChild(paragraph);  
    var pText = document.createTextNode("Bla");  
    paragraph.appendChild(pText);  
}
```

Javascript-Übersicht:

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