

# Hypermedia – Klausur SoSe 2011

Nachname: [REDACTED]

Vorname: [REDACTED]

Matrikelnr.: [REDACTED]

Aufgabe	1	2	
Punkte	48	50	Summe = 98
erreicht	45	48	Summe = 93

Note 1,3  
17.7.11  
[Signature]

## Aufgabe 4) (48 Punkte)

Gegeben ist folgende HTML-Datei:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="de" xml:lang="de" >
  <head>
    <title>Hypermedien SoSe 2011</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  </head>
  <body>
    <h1>Selektoren</h1>
    <p lang="de">
      Selektoren sind Teil einer css-Regel und beschreiben welche Bestandteile
      des html-Dokumentes mit einem Style versehen werden soll.
    </p>
    <p lang="en-US">
      Selectors are part of any css-rule, which describe the specific components
      to be styled within any html-document.
    </p>
    <ul>
      <li>Universal-Selektor</li>
      <li>Typ-Selektoren (Elemente)</li>
      <li>Attribut-Selektoren</li>
      <li>Klassen-Selektor</li>
      <li>id-Selektor</li>
      <li>&Uuml;ber das Attribut <span style="font-style:italic;">lang</span></li>
      <li class="besonders">Pseudo-Klassen:
        <ul class="rechts">
          <li>:active</li>
          <li>:first-child</li>
          <li>:focus</li>
          <li>:hover</li>
          <li>:link</li>
          <li>:visited</li>
        </ul>
      </li>
    </ul>
  </body>
</html>
```

*Eindring s. Aufgabe 2*

```

<li class="besonders">Pseudo-Elemente:
  <ul class="rechts">
    <li>:before</li>
    <li>:after</li>
    <li>:first-letter</li>
    <li>:first-line</li>
  </ul>
</li>
<li>Nachfahren-Selektor</li>
<li>Kind-Selektor</li>
<li>Geschwister-Selektor</li>
</ul>
</body>
</html>

```

- 4 1. Stellen Sie skizzenhaft die Darstellung der Datei im Browser dar, bevor Sie die CSS-Datei erstellen. (4 Punkte) ✓
- 4 2. Binden Sie die norm.css ein, die alle Browser-Standards zurücksetzt und Ihre eigene CSS-Datei. (4 Punkte) ✓
3. Erstellen Sie eine externe CSS-Datei, die die Darstellung wie folgt verändert, wobei im HTML keine Veränderungen vorgenommen werden, außer die CSS-Datei einzubinden. (je 4 Punkte):

- ✓ • Der gesamte Inhalt soll die Breite von 600px bekommen und in der Mitte des Browsers platziert sein. *with margin auf*
- ✓ • Die Schriften im p-Element sollen etwas kleiner als Browser-spezifisch dargestellt sein. *p font size 35%*
- ✓ • Überschriften 1. Ordnung sollen doppelt so groß sein wie <p>, die Farbe Rot erhalten und grau hinterlegt sein. *h1 200% #f00 #ccc*
- ✓ • Die ersten Buchstaben der ersten beiden Absätze sollen extra groß dargestellt werden. *p: first letter 150%*
- ✓ • Der zweite Absatz <p lang="en-US"> soll kursiv sein. *anders als in Darstellung*  
Verwenden Sie einen Attribut-Selektor.
- ✓ • Die Listenelemente der äußeren Listen erhalten einen Punkt (disc) die der untergeordneten Listen einen Kreis (circle). *ul li disc*  
Verwenden Sie einen Attribut-Selektor.
- 2 • Die zwei mit „besonders“ ausgemerkten Listenelemente sind fett. *besonders bold*
- 3 • Die untergeordneten Listenelemente sind nach rechts gerückt und umrahmt. *x*
- ✓ • Die untergeordneten Listenelemente erhalten beim Überfahren mit der Maus einen grauen Hintergrund. *onmouseover bg #ccc*  
Verwenden Sie dafür einen Nachfahren-Selektor.
- ✓ • Geben Sie der CSS-Datei einen Namen und binden Sie sie korrekt ein. Markieren Sie die Stelle in der HTML-Datei.

*S. Aufgabe 1.2*

## Selektoren

Selektoren sind Teil einer css-Regel und beschreiben welche Bestandteile des html-Dokumentes mit einem Style versehen werden soll.

Selectors are part of any css-rule, which describe the specific components to be styled within any html-document.

- Universal-Selektor
- Typ-Selektoren (Elemente)
- Attribut-Selektoren
- Klassen-Selektor
- id-Selektor
- Über das Attribut lang
- **Pseudo-Klassen:**
  - :active
  - :first-child
  - :focus
  - :hover
  - :link
  - :visited
- **Pseudo-Elemente:**
  - :before
  - :after
  - :first-letter
  - :first-line
- Nachfahren-Selektor
- Kind-Selektor
- Geschwister-Selektor

↳ kursiv aber nicht genau verlangt

### Aufgabe 5) (50 Punkte)

Die folgenden HTML- und JavaScript-Dateien gehören zusammen.  
Beantworten Sie die folgenden Fragen.

- a) Was bewirkt die JavaScript-Datei? (40 Punkte)  
In der Funktion *erweitern()* finden Sie Anweisungen, die wir nicht explizit besprochen haben, die Sie aber vom Erlernten ableiten können.  
Erklären diese Funktion Zeile für Zeile genau.
- b) Verändern Sie die Funktion *zufuegen()* so, daß ein beliebiger Text, in einem beliebigen Element an einem beliebigen Ort innerhalb der HTML-Datei hinzugefügt werden kann.  
(10 Punkte)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="de" xml:lang="de">

  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
    1" />
    <link type="text/css" rel="stylesheet" href="auswahl.css" />
    <script type="text/javascript" src="funktionen.js" /></script>
    <title>JavaScript - Auswahl</title>
  </head>

  <body id="main" onLoad="begruessen()">
    <p id="p1" onClick="wertAusgeben('p1');wertAusgeben('d1')">
      Triff Deine Wahl:
    </p>
    <div id="d1" onClick="zufuegen();">
      <p>Hinzufügen ...</p>
    </div>

    <div id="d2" onClick="erweitern('d2', 'p', 'img', 'src', 'Ich bin Mimi!',
    'mimi.gif');">
      <p>
        Drück mich!
      </p>
    </div>

  </body>
</html>
```

5  
//5 Punkte

function begruessen(){

```
    var name = prompt("Hallo", "");  
  
    if(name){  
        alert("Hallo " + name);  
    }  
  
    else{  
        alert("Du musst schon einen Namen eingeben!");  
    }  
}
```

//10 Punkte

function wertAusgeben(el){

```
    var element = document.getElementById(el).nodeValue;  
    alert(element);  
  
    var text = document.getElementById(el).firstChild.nodeValue;  
    alert(text);  
}
```

//10 Punkte lesen, 10 Punkte verändern

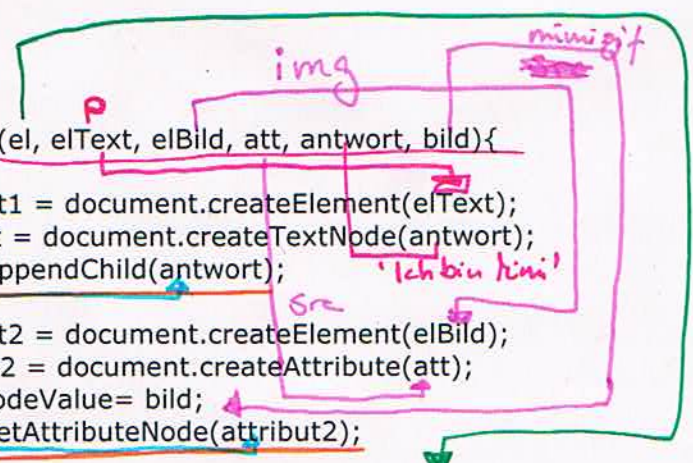
function zufuegen(){

```
    var neuesElement = document.createElement("p");  
  
    var text = document.createTextNode("Hurra, hier bin ICH!");  
  
    neuesElement.appendChild(text);  
  
    var ausgabe = document.getElementById("d1");  
    ausgabe.appendChild(neuesElement);  
}
```

15  
//15 Punkte

function erweitern(el, elText, elBild, att, antwort, bild){

```
1 var element1 = document.createElement(elText);  
2 var antwort = document.createTextNode(antwort);  
3 element1.appendChild(antwort);  
4 var element2 = document.createElement(elBild);  
5 var attribut2 = document.createAttribute(att);  
6 attribut2.nodeValue = bild;  
7 element2.setAttributeNode(attribut2);  
8 var ausgabe = document.getElementById(el);  
9 ausgabe.appendChild(element1);  
10 ausgabe.appendChild(element2);  
}
```



01. Browserdarstellung

Titelleiste Browserfenster

Hypermedien SoSe 2011

h1

# Selektoren

h1

p lang = "de"

Selektoren sind Teil einer CSS-Regel und beschreiben welche Bestandteile des HTML-Dokumentes mit einem Style versehen werden soll

p lang = "de"

p lang = "en"

Selectors are part of any CSS-rule, which describe the specific components to be styled within any HTML-document.

p lang = "en"

ul

- li
- li
- li
- li
- li

- Universal-Selektor
- Typ-Selektoren (Elemente)
- Attribut-Selektoren
- Klassen-Selektor
- id-Selektor
- über das Attribut lang
- Pseudo-Klassen:

ul

li class besonders  
li class besonders  
ul class selbst:

- : active
- : first-child
- : focus
- : hover
- : link
- : visited

li class besonders  
ul class selbst:

Pseudo-Elemente

- : before
- : after
- : first-letter
- : first-line

- Nachfahren-Selektor
- Kind-Selektor
- Geschwister-Selektor

## 2. Einbindung CSS [Dateien liegen im Ordner CSS]

```
<link type = "text/css" rel = "stylesheet" href = "css/norm.css" media = "screen" />
```

```
<link type = "text/css" rel = "stylesheet" href = "css/meinStyle.css" media = "screen" />
```

↳ Einbindung meiner CSS-Datei

## ① 3 Externe CSS : mainStyle.css

body {

width : 600px ;  
margin : auto ;  
font : normal normal normal 35px ; sans-serif ;  
}

h1 {

font-size : 200px ;  
color : #F00 ;  
background-color : #C0C0C0 ;  
}

p: first-letters {

font-size : 150% ;  
}

p [lang="en-US"] {

font-style : italic ;  
}

ul {

list-style : disc ;  
}

ul [class="rechts"] {

list-style : circle ;  
border-style : dashed ;  
padding : 0px 0px 0px 20px ;  
}

20px  
kommt nicht hin

li. besonders {

font-style : bold ;  
}

so sind auch die  
unterpunkte fett

① 3

```
ul[class="rechts"] li:hover {
  background-color: #c0c0c0;
}
```

11.07.2011

Einbindung:

```
<link type="text/css" rel="stylesheet"
href="css/meinStyle.css" media="screen" />
```

②a) Wird die <sup>HTML</sup> Seite geladen, so wird die Funktion begmassen() aufgerufen.

Mit dem Prompt Befehl der Funktion taucht das Wort Hallo ~~und~~ ein Eingabefeld auf und ein Ok-Button auf.

Gibt der User ~~etwas~~ ein wird die Eingabe der Variablen name

Wenn der User den prompt Befehl mit Ok beantwortet, wird der Inhalt der Variablen name zugegeben.

Im nächsten Schritt wird überprüft, ob die Variable name einen Inhalt hat, dann wird ausgegeben

Hallo + Benutzername ← Eingabe des Users

Wenn ~~nicht~~ die Variable keinen Inhalt hat, kommt die Mitteilung:

Du musst schon einen Namen eingeben!

Weiterhin gibt es in der HTML 3 Stellen an denen der User, wenn er klickt Funktionen im Javascript aufruft.  
Eventhandler onclick an den Stellen

p id="p1"

wert ausgeben('p1'); wert ausgeben('d1');

div id="d1"

zufuegen();

div id="d2"

erweiteren('d2', 'p', 'img', 'src', 'ich bin kumi!', 'mini.gif');

## wert ausgehen

p1 : Sucht im Dokument die id = "p1" und weist der variable Element den Knotenwert zu!

Danach wird die Variable Element ausgegeben - also der node.value davon  
wird "Tipp Deine Wahl!"

d1 : Sucht im Dokument die id = "d1" und weist der variable text den Knotenwert des Ersten Kindes zu.

Danach wird die Variable text ausgegeben - also der node.value des Ersten Kindes.

## Zufügen():

Der Variable neues Element wird ein im Dokument neu zu kreierendes p Element zugewiesen.  
[<p>-Element kreiert und zugewiesen]

Der Variable text wird ein Textknoten zugewiesen mit dem Inhalt "Hurra, hier bin ICH!"

Der Variable neues Element wird das Kind Variable text angehängt.

Der Variable ausgabe wird der Ort mit der id "d1" zugewiesen.

~~Der Variable~~ Der Variable ausgabe wird das Kind neues Element angehängt, d.h. an der Stelle mit der id "d1" wird ein p-Element mit dem Textknoten "Hurra, hier bin ich!" neu hinzugefügt.

# function erweitern (el, el Text, el Bild, att, antwort, bild)

11.07.2011

Die Funktion erweitern wird beim klick  
auf das div mit der id = "d2" aufgerufen  
und erwartet 6 Parameter (s.o.) bei  
der übergabe. Es werden übergeben

```
el = d2  
el Text = p  
el Bild = img  
att = src  
antwort = "Ich bin mimi"  
bild = mimi.gif
```

Zunächst wird die Variable element 1 deklariert  
und gleichzeitig ein neu zu erstellendes Element  
(aus der Variablen el Text) zugewiesen. In unserem  
Fall handelt es sich dabei um ein span-Element.

Danach wird die Variable antwort deklariert,  
das ein neuer Textknoten mit dem Inhalt  
antwort - in unserem Fall 'Ich bin mimi' -  
zugewiesen wird.

Im dritten Schritt wird dem Element 1 das  
Kind antwort angehängt.

Im nächsten Teil passiert es ähnlich:  
Der neuen Variablen element 2 wird ein  
Element aus der Variablen el Bild zugewiesen  
- in unserem Fall ein img-Element.

Dann wird der Variablen attribut 2 ein neues  
Attribut aus der Variablen att zugewiesen -  
bei uns das Attribut src = ""

Im nächsten Schritt wird ~~der~~ der Variablen attribut 2  
~~der~~ Wert (nodeValue) aus der Variablen  
bild ~~zu~~ (mimi.gif) zugewiesen, sodass es  
jetzt heißt: src = "mimi.gif"

Im nächsten Schritt wird diese Attribut samt  
Wert dem img-Element (element 2) zugewiesen.  
(Bestünde es schon, würde es überschrieben, besteht  
es nicht, wird es hinzugefügt)

Im letzten Abschnitt (Zeile 8-10) wird der Variablen  
ausgabe der Ort im Dokument zugewiesen, der  
über eine id gefunden wird - die id die  
beim aufruf im html an die Variable el  
übergeben wurde. Bei uns die id = "d2".

An dieser Stelle wird nun das Element 1  
und das element 2 angehängt.

Die Funktion erweitern, setzt dementsprechend in der HTML Datei in das div mit der id "d2" zuerst ein p Element mit dem Text "Ich bin Tim" und anschließend ein Bild (<img>) mit der Quellenangabe / Dateiname src="mimigif".

Die HTML Seite wurde um Text und Bild reicher.

② b) zufügen (tx, el, id)

```
var neuesElement = document.createElement(el);
```

```
var text = document.createTextNode(tx);
```

```
neuesElement.appendChild(text);
```

```
var ausgabe = document.getElementById(id);
```

```
ausgabe.appendChild(neuesElement);
```

}

Anfang im HTML (bspw.):

```
<div id="abs1" onclick="zufügen('daron', p, 'abs1')">
```

oder

```
<div id="abs2" onmouseover="zufügen('Dübergewicht und hergezerrt!', p, 'abs2')">
```