

Interactive Media – Klausur WS 2014/15

Nachname:

Vorname:

Matrikelnr.:

Aufgabe	1	2	3	
Punkte	20	50	30	Summe = 100
erreicht	15	37	27	Summe = 72 83

2,3
AS

Punkte	%	Note	Punkte	%	Note	Punkte	%	Note	Punkte	%	Note	Punkte	%	Note
100 - 97	97	1,0	96 - 93	93	1,3	92 - 90	90	1,7	89 - 87	87	2,0	86 - 83	83	2,3
82 - 80	80	2,7	79 - 77	77	3,0	76 - 73	73	3,3	72 - 67	67	3,7	66 - 50	50	4,0

Aufgabe 1) (20 Punkte)

Wie werden CSS-Regeln abgearbeitet?

(18)

1. Nach dem Stylesheet-Ursprung:

- 1) Browserstylesheet
- 2) Benutzerstylesheet
- 3) Autorenstylesheet
- 4) Autorenstylesheet mit !important
- 5) Benutzerstylesheet mit !important

Herzkunft ✓
Spezifität ✓
Sequenz ✓
Media

2. Nach ihrer Spezifität:

- 1) Elemente & Pseudoelemente
- 2) Klassen & Pseudoklassen
- 3) id-Attribut
- 4) style-Attribut

3. Wenn immer noch Konflikte bestehen, werden die Regeln entsprechend ihrer Reihenfolge (im CSS) Dokument abgearbeitet.

2a)

(lehr prof)

Einige js-Arbeitsanweisungen und Funktionen

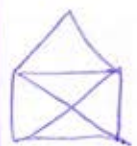
(prof)

Properties verwenden

(normal)

- className
- src
- style
- value

<code>?



(prof)

Methoden aufrufen und verwenden

(normal)

- create Element ()
- append Child ()
- create Text Node ()
- ~~set~~ Attribute ()

Textgröße

600 px

37

Aufgabe 2) (50 Punkte)

Studieren Sie die folgenden beiden Dateien (pa1.html und myStyles.css).

- 4) a) Skizzieren Sie die Darstellung im Browser ohne Einbindung eines Autoren-Stylesheets. Wichtig ist der Textfluß. (5 Punkte)
- 23) b) Skizzieren Sie die Darstellung im Browser mit Einbindung des Autoren-Stylesheets (s. nächste Seite). Beachten Sie die Größenangaben und beschriften Sie Elemente mit Größenangaben. (25 Punkte) *siehe letztes Blatt*
- 5) c) Binden Sie das Stylesheet „myStyles.css“ aus dem untergeordneten Verzeichnis „styles“ für alle Medien ein. (5 Punkte)
- 3) d) Wodurch unterscheiden sich die Werte **inline** und **inline-block**? (5 Punkte)
- 2) e) Was verändert sich in der Browseransicht, wenn im Stylesheet der Selektor **body::before** in **body ::before** verändert wird? (10 Punkte)

Für *haus.png* stellen Sie sich eine kleine Grafik vom Haus-des-Nikolaus vor, in der Größe von etwa 25px mal 25px (klein).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Nützliches in JS</title>
    <meta charset="utf-8" />
    <link href="styles/myStyles.css" type="text/css" rel="stylesheet" media="all" />
    <script src="interActPA2.js" type="text/javascript"></script>

  </head>
  <body>
    <header>
      <h1>Einige js-Arbeitsanweisungen und Funktionen</h1>
    </header>
    <section>
      <h2>Properties verwenden</h2>
      <ul>
        <li id="className"><code>className</code></li>
        <li id="src"><code>src</code></li>
        <li id="style"><code>style</code></li>
        <li id="value"><code>value</code></li>
      </ul>
      
      <h2>Methoden aufrufen und verwenden</h2>
      <ul>
        <li><code>createElement</code></li>
        <li><code>appendChild</code></li>
        <li><code>createTextNode</code></li>
        <li><code>setAttribute</code></li>
      </ul>
    </section>
  </body>
</html>
```

2d) Elemente mit der Eigenschaft `inline` werden nebeneinander dargestellt, solange sie in das Browserfenster passen.

▣ Inline-Block verhält sich ähnlich, nur dass das Element ein Blockelement bleibt, welches jedoch nur den benötigten Platz einnimmt. Auch Elemente mit dieser Eigenschaft können also nebeneinander stehen. Die Fläche um sie herum wird jedoch auch "klickbar".

ZB. Größenangaben sind erlaubt

e) ~~Das Wort "Aha!" wird nicht mehr dargestellt, da das Pseudoelement `::before` keinen Bezugspunkt mehr hat. Der Browser weiß also nicht mehr, vor welchem Kindknoten dieser Text platziert werden soll.~~

Das Leerzeichen dient im CSS als Nachkommenseparator.

genau: alle Elemente, die Nachfahren des `<body>` sind, bekommen "Aha!" vorangestellt.

```
*{  
  box-sizing: border-box; px - padding - content  
  -webkit-box-sizing: border-box;  
  -moz-box-sizing: border-box;  
  
  font-size: 1rem;  
  font-family: Arial, sans-serif;  
}
```

```
body{  
  margin: 0 auto; wichtig!  
  width: 600px;  
}
```

```
h1{  
  font-size: 2rem; ✓  
}
```

```
h2{  
  font-size: 1.5rem; ✓  
  text-decoration: underline; ✓  
}
```

```
section{  
  position: relative; ✓  
  border: 1px dotted black; ✓  
}
```

```
[alt]{  
  position: absolute; }  
  top: 0; }  
  right: 0; }  
  padding: 10px 5px; }  
}
```




```
ul>li{ spezifisches  
  display: inline-block; relativ  
  border: 1px solid red; 200px  
  width: 200px;  
}
```

```
li{  
  border-left: 2px solid black; zwei nebeneinander  
}
```



```
code{  
  width: 400px;  
}
```

```
body::before{  
  content: "Aha!"; Aha  
  background-color: #c0c0c0; grau  
}
```



```
.list{  
  font-weight: bold; Klasse nicht verfehle!  
}
```

30 27

Aufgabe 3) (30 Punkte)

Gegeben sind die Dateien pa1.html, myStyles.css (s.o.) und die folgende unvollständige myscript.js
Bearbeiten Sie die jeweiligen Fragen.

```
window.onload=init;
```

```
function init(){
```

```
/*  
a) Lassen Sie per Mausklick auf jeden Punkt der zweiten Liste die beiden Funktionen  
changeClass() und addImage() ausführen (10 Punkte)  
*/
```

```
var liste = document.getElementById('ul')[1];  
for-Schleife für <li> ('li')[5,6,7,8]; (?) ?  
liste.addEventListener('click', change function(){ changeClass();  
addImage(); },  
false) ✓
```

```
}
```

```
/*  
b) Was bewirkt der folgende Anweisungsblock? Worauf bezieht sich hier „this“?  
(5 Punkte)  
*/
```

```
function changeClass(){  
var singleLi=this;  
singleLi.className='list';  
}
```

- ✓ „this“ bezieht sich auf das -Element, welches im Browser angeklicket wurde.
- ✓ Diesem wird die neue Klasse (class="list") zugeordnet.
- ✓ D.h. der Text wird von da an fett dargestellt.

/*

c) Formulieren Sie den Anweisungsblock für die Funktion addImage.
Mit der Funktion soll ein neues am Ende der HTML-Datei eingefügt werden.
(10 Punkte)

*/

function addImage(){

var bild = document.~~getElement~~ createElement('img');

var src = document.createElement('src');

src.value = 'Bild.jpg';

bild.setAttributeNode(src);

var platzieren = document.getElementsByTagName('body');

platzieren.appendChild(bild);

}

/*

d) Wie könnten Sie die Funktion erweitern, um ein beliebiges Bild einzufügen,
d.h. der Wert für das Attribut src soll variable sein.

Wenn Sie das schon in der vorherigen Frage getan haben, dann tun Sie hier nichts ☺
(5 Punkte)

*/

Indem ein Parameter, also die Bildsource, übergeben wird:

function addImage(x) {

var bild ... ;

var src ... ;

src.value = x;

bild.set ... ;

... ;

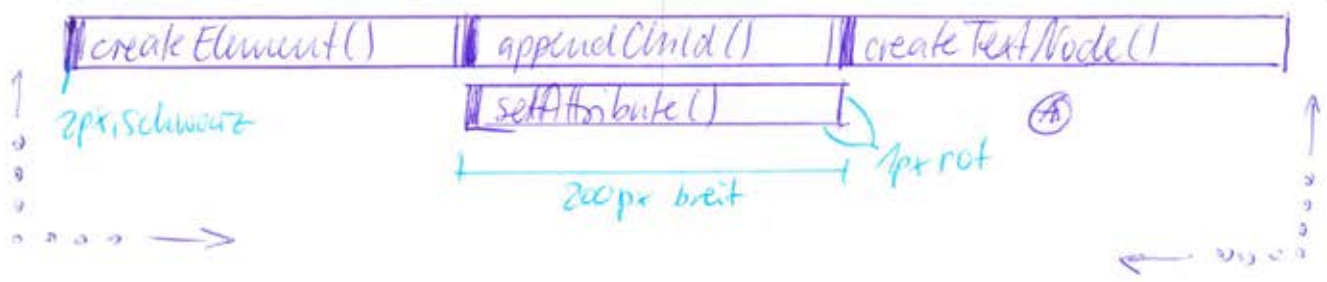
}

2b) Aha! Einige JS-Arbeitsanweisungen und Funktionen

1rem
2rem →



Methoden aufrufen und verwenden } 1,5rem



2rem = 2 mal so groß, wie normaler Text

1,5rem = 1,5 mal so groß, wie normaler Text

⊕ Hinweis: Border-box hat 600 px Begrenzung.
 Durch alle 1px breiten Rahmen passen nicht 3 Listenelemente nebeneinander, sondern nur 2, da rechts & links jeweils 1px fehlt. $600 - 2 = 598$ px

↳ Properties

