Interactive Media – Klausur SoSe 2012

Nachname: [redacted] Vorname: [redacted] Matrikelnr.: [redacted]

Note 1,3

23.7.12

Aufgabe	1	2	3	4	
Punkte	10	20	48	50	Summe = 128
erreicht	6	18	43	50	Summe = ~~115~~ 117

Aufgabe 1) (10 Punkte)

Was ist eine Netzwerk-Topologie, was beschreibt sie und wozu dient sie?
Beschreiben Sie eine beliebige Form im Detail.

Aufgabe 2) (20 Punkte)

Erklären Sie was Spezifität bedeutet und wann sie eingesetzt wird. (8 Punkte) auf Blatt
Für was stellen die folgenden einzelnen Selektoren jeweils eine Regel auf? (12 Punkte)

	A	B	C	D	
h1, h2, h3	0	0	0	1	alle drei h's sind gleichwertig
#content p	0	1	0	1	betrifft alle p, die auf das Element mit Attribut id="content" folgen
.fettKursiv	0	0	1	0	für alle Element mit Klasse fettKursiv
h1.blau	0	0	1	1	für h1 Elemente mit Klasse blau
body ul li	0	0	0	3	für alle li, die auf ul und body folgen, Zeitpunkt egal
#content > p	0	1	0	1	für alle p, die direkt auf das Element mit id=content folgt

das erste das

Punkte	%	Note	Punkte	%	Note	Punkte	%	Note	Punkte	%	Note	Punkte	%	Note
124	97	1,0	119	93	1,3	115	90	1,7	111	87	2,0	106	83	2,3
102	80	2,7	99	77	3,0	93	73	3,3	86	67	3,7	64	50	4,0

Aufgabe 3) (48 Punkte)

Gegeben ist die folgende HTML-Datei:
Bitte lesen Sie erst den gesamten Aufgabentext zu Ende, bevor Sie mit der Aufgabe beginnen!

```
<!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="monitor.css" media="screen" />


      <title>XHTML und CSS</title>

    </head>


    <body>

      <h1>XHTML und CSS</h1>

      <h2>Einiges zu den Themen <em>XHTML</em> und <em>CSS</em>...
          erfahren Sie hier</h2>

      <p>z.B. über das Boxmodell ...</p>

      <p>... über Vererbung ...</p>

      <h2>Links zum Thema:</h2>

      <ul>

        <li><a href="">css</a></li>
        <li><a href="">javascript</a></li>
        <li ><a href="">html</a></li>
        <li><a href="">php</a></li>

      </ul>

      <!-- Fußzeile -->
      <p>Copyright und so weiter</p>
        id="fuss"
    </body>
  </html>
```

1. Stellen Sie skizzenhaft die Darstellung der Datei im Browser dar, bevor Sie eine CSS-Datei erstellen. (4 Punkte)

2. Was verändert sich, wenn ich die Datei norm.css einbinde? (4 Punkte)

3. Erstellen Sie eine externe CSS-Datei, die die Darstellung wie folgt verändert, bedenken Sie, daß Sie bereits die norm.css eingebunden haben.
(je 4 Punkte):

- Alle Schriften sollen Verdana oder zumindest serifenlos sein.
- Die Schriften im p-Element sollen eine Größe von 12px erhalten, die Überschriften 1. Ordnung sollen doppelt so groß, die Überschriften 2. Ordnung 1,5 mal so groß und die Verlinkungen 80% bemessen am p-Element sein.
Vereinbaren Sie die Größen relativ aufeinander bezogen.
- Die gesamte Seite erhält die Hintergrundfarbe Grau und die einzelnen Elemente darin erhalten ein helleres Grau.
- Die Links sollen nebeneinander und nicht in Listenform dargestellt werden, also auch ohne Listenpunkte.
- Beim Überfahren der Links werden diese fett dargestellt.
- Das p „Fußzeile" soll kleiner als der Rest der Seite dargestellt werden.
- Der erste Buchstabe der Überschriften soll fett dargestellt werden.
- Färben Sie die Worte „XHTML" und „CSS" blau ein und lassen Sie beim Überfahren mit der Maus die Schrift größer und fetter werden.
- Geben Sie der CSS-Datei einen Namen und binden Sie sie korrekt ein.

4. Binden Sie Ihre CSS-Datei korrekt ein.

Sie dürfen in der HTML-Datei beliebig Attribute vergeben.
Sie können weitere Elemente hinzufügen, allerdings nur wenn unbedingt notwendig.
Sie können in die HTML-Datei hineinschreiben, wenn Ihnen der Platz fehlt, dann markieren Sie wo der entsprechenden Text stehen soll.

Aufgabe 4) (32 Punkte)

Gegeben sind die nachfolgende HTML- und CSS-Dateien,
beantworten Sie die Fragen im JavaScript.

```
<!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="func.css" />
    <script type="text/javascript" src="Klausfunc.js" /></script>
  <title>JavaScript</title>
  </head>

  <body id="main" onLoad="zufuegen('h1','Willkommen','main')">

    <p id="abs1" onClick="wertTyp('main');">Absatz 1</p>

    <div id="platzhalter">

      <p onClick="zufuegen('p','Hurra, hier bin ich!','platzhalter');">
      Anh&auml;ngen ...</p>

    </div>

    <p id="zahl" onClick="wiederholen();">Z&auml;hlen</>

    <p id="anzeige" onClick="erweitern();">
      Bild anzeigen ...
    </p>

  </body>
</html>
```

```
/*Was bewirkt die folgende Funktion, wenn sie in der HTML-Datei aufgerufen wird?
10 Punkte*/

function wertTyp(el){

    var element = document.getElementById(el).nodeType;
    alert("nodeType " + element + " ist Element");

    element = document.getElementById(el).nodeValue;
    alert(element);

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

```
/*Was bewirkt die folgende Funktion, wenn sie in der HTML-Datei aufgerufen wird?
10 Punkte*/

function zufuegen(x,y,z){

   var neuesElement = document.createElement(x);
   var text = document.createTextNode(y);

   neuesElement.appendChild(text);

   var ausgabe = document.getElementById(z);
   ausgabe.appendChild(neuesElement);
}

/*-------------------------------------------------------------*/

/*Was bewirkt die folgende Funktion, wenn sie in der HTML-Datei aufgerufen wird?
10 Punkte*/

function wiederholen(){

   var zahl = prompt("Gib eine Zahl ein ...", "5");

   if(zahl < 8){

      for (var x = zahl; x>=0; x--){
         zufuegen('p',zahl,'zahl');
      }
   }
   else
      zufuegen('p','Das ist mir zuviel!','zahl');
}

/*-------------------------------------------------------------*/

/*Wie kann diese Funktion verallgemeinert und vereinfacht werden?
Betrachten Sie dabei das gesamte Skript und nutzen Sie andere Funktionen, wenn
möglich(20 Punkte)*/


function erweitern(){

   var element1 = document.createElement("p");
   var text = document.createTextNode("Ich bin Mimi!");
   element1.appendChild(text);

   var element2 = document.createElement("img");
   var attribut2 = document.createAttribute("src");
   attribut2.nodeValue= "mimi14.gif";
   element2.setAttributeNode(attribut2);

   var ausgabe = document.getElementById("anzeige");
   ausgabe.appendChild(element1);
   ausgabe.appendChild(element2);
}
```

Aufgabe 1

```
<h1> XHTML und CSS
<h2> Einiges zu den Themen XHTML und CSS ... erfahren Sie hier
<p> z.B. über das Boxmodell
<p> ... über Vererbung
<h2> Links zum Thema:
<ul>
<li> • CSS
<li> • javascript
<li> • html
<li> • php
</ul>
<p> Copyright und so weiter
```

(-1) ✓

2. h1 und h2 haben die gleiche Schriftgröße/-breite/-style, wie Listenpunkte existieren dann nicht mehr. Die ganze Datei wird quasi normalisiert. ✓

3. @import „norm.css"; url(norm.css);

```
body{
    font-family: Verdana;
    font-size: 12px;
    background-color: #COCOCO;
}

p{
    font-size: 1em;
    background-color: #DODODO;
}

h1{
    font-size: 2em;
    background-color: #DODODO;
}

h2{
    font-size: 1,5em;
    background-color: #DODODO;}
```

✓ — Alternativen fehlen (-1)

zu 3)

```
a {
    font-size: 80%;
    background-color: #DODODO;
}

ul li {
    background-color: #DODODO;
    display: inline;
    list-style-type: none; /* braucht man nicht wegen norm.css */
}

a:hover {
    font-weight: bold;
}

h1, h2 :first-letter {
    font-weight: bold;
}

em {
    color: #0000FF;
}

em:hover {
    font-size: 200%;
    font-weight: bold;
}

#fuss {
    font-size: 50%
}
```

Speichern: monitor.css

Aufg. 2) Spezifität beschreibt die Priorität der Selektoren. style-Attribute im html haben die höchste Priorität, danach folgen #id, dann Klassen u. Pseudo-Klassen und zuletzt Elemente und Pseudo-Elemente.

wann wird sie angewandt? (-2)

Aufg. 1) Eine Netzwerk-Topologie beschreibt den Aufbau eines Rechnernetz. Sie dient der Planung und Übersicht über das Netzwerk. (-4)

Als Beispiel wäre da die Stern Topologie. Ein Server steht in der Mitte und kommuniziert mit den Clients. Die Clients müssen über den Server um miteinander zu kommunizieren.

Fällt ein Kabel oder ein Client aus, ist es unproblematisch. Fällt der Server aus, ist alles funktioniert kein Netzwerk.

Interactive Media Klausur SoSe 2012

Aufg. 4) Die wertTyp Funktion gibt Knotentypen und -werte aus. Durch alert("nodeType "+element+" ist Element"); wird eine Meldung ausgegeben die den Text „nodeType 1 ist Element" ausgibt. Die 1 wird ausgegeben, da Elementknoten die 1. Stufe quasi angeben.
Beim zweiten Alert wird null ausgegeben, da Elementknoten bei nodeValue den Wert null haben.
Das dritte alert gibt den Wert vom ersten Kind des Elements aus. In diesem Fall wäre es „Absatz 1", da der Text das erste Kind eines p Elements ist. " "

Bei der Funktion zufuegen wird neuer Text angehängt. Zuerst wird durch createElement(x) ein neues p Element erstellt, weil p als erster Parameter übergeben wird. Als nächstes wird eine Variable mit dem Inhalt, dem zweiten übergebenen Parameter, erstellt. Danach wird der Inhalt bzw. Text dem Element als Kind angehängt. Danach wird noch eine ausgabe Variable definiert, damit das Script weiß, wo es dann das Element, welches der Ausgabe Variable als Kind angehängt wird, ausgegeben werden soll. *r

Wenn die Funktion wiederhden ausgeführt wird, dann wird zuerst ein Eingabefenster erscheinen. Dort muss man eine Zahl eingeben oder die 5 stehen lassen. Dann prüft das Script mit einer if, ob die Zahl kleiner als 8 ist. Falls dies nicht der Fall ist, wird die Funktion zufuegen aufgerufen. Wenn die Zahl aber kleiner als 8 ist, dann läuft eine for-Schleife durch, in der die Zahl auf 0 runter-

zu 4) gezählt wird. ~~Für~~ Jede Zahl wird dann mit der Funktion zufuegen ~~die Zahl~~ in der html-Datei ausgegeben. ✓

Die Funktion erweitern() kann man ~~wiefolgt~~ vereinfachen, wenn ~~function erweitern(){~~ der erste Teil durch die Funktion zufuegen ersetzt wird. Auf den zweiten Teil kann es nicht benutzt werden. *

```
function erweitern(){
    ~~var elementA~~ zufuegen('p','Ich bin Mimi!','anzeige');

    var element = document.createElement("img");
    var attribut = document.createAttribute("src");

    attribut.nodeValue = "mimi14.gif";
    element.setAttributeNode(attribut);

    var ausgabe = document.getElementById("anzeige");

    ausgabe.appendChild(element);
}
```

* Im zweiten Teil, also beim zweiten Element, wird statt einer TextNode, ein AttributeNode erstellt, demzufolge müsste die Funktion zufuegen umgeschrieben werden. ✓

(7.1)

*2 zufuegen wird auch schon beim Laden der Seite gestartet und dort wird kein p-Element erstellt, sondern ein h1. Die Funktion ✓ erstellt Elemente, füllt sie mit Inhalt und lässt sie ausgeben.