

# Interactive Media – Klausur SoSe 2012

Nachname:

Vorname:

Matrikelnr.:

Aufgabe	1	2	3	4	
Punkte	10	20	48	50	Summe = 128
erreicht					Summe =

## 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)  
Für was stellen die folgenden einzelnen Selektoren jeweils eine Regel auf? (12 Punkte)

h1, h2, h3  
#content p  
.fettKursiv  
h1.blau  
body ul li  
#content > p

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" />

    <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>

  </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);  
}
```