Hypermedia – Klausur SoSe 2010

Bitte füllen Sie die folgenden Felder aus:

Nachname: Vorname: Matrikelnr.:

Aufgabe	1	2	3	4	5	92
Punkte	6	10	8	44	24 ~~32~~	Summe = ~~100~~
erreicht	6	10	8	44	24	Summe = 92

Note 1,0
10.07.10

Aufgabe 1) (6 Punkte)

Stellen Sie die folgende CSS-Regel mit dem Farbton als Dezimalzahl dar. Geben Sie den Rechenweg vollständig an und schreiben Sie die Regel vollständig und korrekt auf.

.hintergrund {background-color: #456;}

Aufgabe 2) (10 Punkte)

Erklären Sie jeweils die Begriffe HTML, CSS, JavaScript und PHP, wozu sie dienen und wie sie zusammenspielen können.

Aufgabe 3) (8 Punkte)

Berechnen Sie die Spezifität der folgenden Selektoren (4 Punkte) und erklären Sie was Spezifität bedeutet und wann sie eingesetzt wird (4 Punkte).

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

Aufgabe 4) (44 Punkte)

Gegeben ist folgende HTML-Datei:

```
<!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 rel = "stylesheet" media = "screen" href = "style-screen.css" type = "text/css" />

        <title>XHTML und CSS</title>

    </head>


    <body>
        <div id = "wrapper" >
            <!-- Überschrift, Logo, ... -->

            <div class = "headline" >
                <h1>XHTML und CSS</h1>
            </div>

            <!-- Hauptinhalt -->
            <div id = "content" >
                <p>
                    Einiges zu den Themen <span>XHTML</span> und
                    <span>CSS</span>...<br />
                    ... erfahren Sie hier.
                </p>
                <p>
                    z.B. über das Boxmodell ...
                </p>
                <p>
                    ... über Vererbung ...
                </p>
            </div>
            <! --Links zum Thema -->
            <div id = "content-right" >
                    <h2>Links zum Thema:</h2>
                    <ul>
                        <li><a href="">link</a></li>
                        <li><a href="">link</a></li>
                        <li ><a href="">link</a></li>
                        <li><a href="">link</a></li>
                    </ul>
            </div>
            <! --Fußzeile -->
            <div id = "footer" >Copyright und so weiter</div>
        </div>
    </body>
</html>
```

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

2. Erstellen Sie eine externe CSS-Datei, die die Darstellung wie folgt verändert (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.
   - Die gesamte Seite erhält die Hintergrundfarbe Rot und die einzelnen Boxen darin erhalten ein helleres Rot.
   - Die Links sollen zwar untereinander, aber nicht in Listenform dargestellt werden.
   - Beim Überfahren der Links werden diese fett dargestellt.
   - Die div „Hauptinhalt" und „Links zum Thema" sollen nebeneinander platziert werden.
   - Das div „Fußzeile" wird unter den beiden div „Hauptinhalt" und „Links zum Thema" über deren gesamte Breite angezeigt
   - Das erste p-Element im div „Hauptinthalt" soll fett dargestellt werden, die beiden weiteren p-Elemente nicht.
   - Färben Sie die Worte „XHTML" und „CSS" aus dem ersten p-Element im div „Hauptinhalt" blau ein.
   - Geben Sie der CSS-Datei einen Namen und binden Sie sie korrekt ein.

     Verwenden Sie die Attribute „class" und „id" so, daß es ihren Aufgaben entspricht.

     Alles was das HTML betrifft schreiben Sie direkt in den Quelltext hinein.

Aufgabe 5) (32 Punkte)

Gegeben sind die folgenden HTML- und CSS-Dateien:

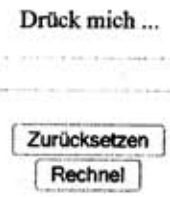
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<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="styles/begruessung.css" />
                <script type="text/javascript" src="skripten/begruessung.js" /></script>
        <title>JavaScript</title>
        </head>

        <body onload="begruessung();wachWerden();">
                <div id="hitMe" onclick="KnotenWert('hitMe');">
                        <p id="text" onclick="begruessung();"
                        onclick="KnotenWert('text');">Drück mich ...
                        </p>
                </div>
                        <form name="formular" id="formular" action="machWas.php"
                        method="post" />
                        <input type="text" id="feld1" name="feld1"
                        onclick="KnotenWert('feld1');" />
                        <input type="text" id="feld2" name="feld2"
                        onclick="KnotenWert('feld2');" onclick="aendern();" />
                        <input type="reset" id="knopt" onclick="KnotenWert('knopf');" />
                        <input type="button" value="Rechne!"onclick="berechnen();" />
                </form>
        </body>
</html>
```

Browseransicht:



```
function begruessung(){

        var name = prompt("Hallo! Wie heißt Du?");
        name="";

        if(name){
              alert("Schön Dich zu sehen, " + name);
        }

        else if(!name){
              alert("Du mußt schon einen Namen eingeben!");
        }
}

function aendern(){

        if(document.getElementById("feld1").value==""){
              alert("Gib einen Wert ein Feld 1 ein!");
        }
        else{
        document.getElementById("feld2").value=document.getElementById("feld1").valu
e;
        }
}

function berechnen(){

        var summe=0;
        /*Daten aus Feldern werden als Strings entgegen genommen.
        Strings werden konkateniert, deshalb wird hier konvertiert mit parseInt()
        Die Funktion isNAN steht für prüft, ob der übergebene Parameter eine Zahl ist.*/

        if(isNaN(document.getElementById("feld1").value)||isNaN(document.getElementB
        yId("feld2").value)){
              alert("Das ist keine Zahl!");
        }

        else{
              summe = parseInt(document.getElementById("feld1").value) +
              parseInt(document.getElementById("feld2").value);
              alert(summe);
        }

}

function wachWerden(){

        setInterval("alert('Aufwachen!!!');", 50000);
}
```

Erklären Sie was mit Hilfe des JavaSkript abläuft,

- Wenn die Datei im Browser geladen wird (4 Punkte)
- Wenn eine Zahl in das erste Formularfeld eingegeben wird und auf das div „hitMe"" gedrückt wird (10 Punkte)
- Wenn ein Buchstabe in das erste Formularfeld eingeben wird und auf das div „hitMe"" gedrückt wird (6 Punkte)
- Wenn die Seite längere Zeit geöffnet bleibt (4 Punkte)

1) Umrechnung Hex_{16} in Dezimal

Gegeben: #456; => #445566 ✓

Lösung: $44_{16} = 4 \cdot 16 + 4 = 68_{10}$ ✓

$55_{16} = 5 \cdot 16 + 5 = 85_{10}$ ✓

$66_{16} = 6 \cdot 16 + 6 = 102_{10}$ ✓

=> .hintergrund {background-color: rgb(68,85,102);} ✓

2)

HTML: Hypertext Markup Language

Dient zum strukturierten Aufbau eines Hypertextes, welches durch Elemente realisiert wird. Jedes Element ist so bezeichnet, dass es den Inhalt eindeutig typisiert (Absätze – ; Tabelle – <table> ... </table>; usw. Wird zB verwendet zum Strukturieren einer Webseite.

CSS: Cascading Style Sheet

Dient zur logischen Formatierung eines Hypertextes. Ermöglicht farbliche Gestaltung und Positionierung von HTML-Elementen. Wird für die Gestaltung von HTML-Dokumenten verwendet, wie zB Webseiten.

JavaScript: JS ist eine Scriptsprache. Sie ermöglicht die dynamische Manipulation des DOMs direkt auf dem Client (Browser). Wird benutzt um Plausibilitätsprüfungen (z.B von HTML-Formularen) durchzuführen, HTML-Inhalte ohne Seiten-Refresh auszutauschen oder die Interaktion mit PHP-Scripten ebenfalls ohne die Seite neuzuladen, zu ermöglichen (Ajax)

PHP: Hypertext Pre-Processor

Logische Scriptsprache, welche von einem Server interpretiert wird und an den Browser HTML/CSS und/oder JavaScript-Code zurückgibt.

Wird zB verwendet um dynamischen HTML-Code zu generieren (z.B Tabellen mit Inhalt aus einer Datenbank), Berechnungen durchzuführen und als HTML-Ausgabe zu liefern. Ebenfalls kann PHP dynamische Stylesheets generieren oder aber auch JavaScript-Code dynamisch generieren.

3.

	A	B	C	D	
a) h1, h2, h3	0	0	0	3	✓
b) #content p	0	1	0	1	✓
c) .fettkursiv	0	0	1	0	✓
d) h1.blau	0	0	1	1	✓
e) body ul li	0	0	0	3	✓
f) #content > p	0	1	0	1	✓

Die Spezifität entscheidet über die Priorität und Priorisierung von im CSS-Code verwendeten Stildefinitionen. Dabei entscheidet primär der „Typ" des Stiles die Priorität. Ein Element, welches über das style-Attribut eine ~~ID oder Klasse~~ Formatierung definiert, hat eine höhere Wertigkeit als die Definition einer ~~ID oder Klasse~~ Formatierung über eine in einem externen Stylesheet. Folgendes gilt: ✓

A: style-Attribut, z.B `<span style="color: red;">...</span>`

B: #id-Definition, z.B
```
#fehler {
    color: red;
}
```

C: Klassendefinition, z.B
```
~~.class fehler~~ .fehler {
    color: red;
}
```

D: Elementselektor, z.B
```
p {
    margin: 0;
}
```

Priorität ↑

↳ Die Spezifizierung entscheidet also, welche Stildefinition letztendlich verwendet wird, wenn bsp. Redundanzen aufgetreten sind. ✓

XHTML und CSS

Einiges zu den Themen XHTML und CSS ...
... erfahren Sie hier.

z.B über das Boxmodell ...

... über Vererbung ...

Links zum Thema:

- link
- link
- link
- link

Copyright und so weiter

4.2)

```
/* style-screen.css */
body {
    font-family: Verdana, sans-serif; background-color: red;
    font-size: 12px;
}

p {
    font-size: ~~12px;~~ 1em;
}

h1 {
    font-size: 2em;
}

h2 {
    font-size: 1.5em;
}

a {
    font-size: 80%;
}

dir {
    background-color: ~~#FF1000~~;
}

ul {
    list-style-type: none;
    margin: 0;
}

a:hover {
    font-weight: bold;
}
```

```
#content {
    float: left;
    width: 250px;
}

#content-right {
    width: 250px;
    margin-left: 10px;
}

#footer {
    clear: left;
}

#content > p {
    font-weight: bold;
}

.headline {
    color: blue;
}
```

(5)

1. Beim Laden des Dokuments werden die Funktionen begruessung() und wachWerden() aufgerufen.

begruessung: Textprompt / Eingabefenster öffnet sich und fordert den Nutzer auf seinen Namen einzugeben. Dieser wird in der Variable name gespeichert. ✓

Wenn ein Name eingegeben, dann Ausgabe im Hinweisfenster: | Schön, dich zu sehen, [name] | ✓

Wenn Feld leer gelassen, dann Hinweismeldung mit Fehlerbeschreibung ✓

wachWerden: Startet ein Intervall, welches alle 50.000ms die Hinweismeldung 'Aufwachen!' ausgibt. ✓

2. Funktion knotenWert wird ausgeführt, Diese existiert nicht. (?) bei Klick auf Formular-Input. ✓
→ Fehlermeldung in der Fehlerkonsole:

| knotenWert() is not a function on line xy. |

3. Nachdem Zahl eingegeben wurde und auf den Container geklickt wird, wird ebenfalls die Funktion knotenWert() ausgeführt. -> Dise existiert nicht. Fehlermeldung wie oben. ✓

Siehe 2. (?)

4. Alle 50 Sekunden erscheint das Alert-Hinweisfenster mit der Meldung 'Aufwachen!' ✓

Dies liegt daran, dass während des Seitenaufbaus die Funktion wachWerden aufgerufen wurde, welche ein ~~Timer~~ Intervall gestartet hat, dessen Anweisung (1. Parameter) alle 50.000 ms (2. Parameter) ausgeführt wird.

Farben: #000000 ...Schwarz | #FFFFFF ...Weiß

```
color : rgb (0,0,0);        color : black;
color : rgb (255,255,255);  color : white;
color : rgb(0%, 0%, 0%) ;   color : #fff;
color : rgb(100%, 100%, 100%);
```

HTML-Einwandgerüst:

```
<! Doctype ... >
<html xmls="http://www.w3.org/1999/xhtml" dir="ltr" lang="de">
```

CSS-Include:

CSS-Syntax:

```
<style type="text/css">[<style media="all" type="...
@import "allgemein.css";
@import url("allgemein.css");
@import url("druck.css") print, embossed;
                         handheld; screen;
</style>
```

HTML (extern):

```
<link rel="stylesheet" media="screen" href="style.css">
```

(attribut type="text/css" ggf.)

HTML-Attribut:

```
<p style="font-weight : bold; > Lorem</p>
```

BB:

width, padding, border-width, margin
height

[1px, 100%, 3em, ...]

```
<div>Blah</div>
<span>Blah</span>
```

Selektoren:

```
div > p { letter-spacing : -1px; }
div p + p + p + p { display : list-item; list-style : inside; }
```

Attr.-Selektoren

```
[alt] { border : none; }
```

PE: :after, :before, :first-letter, :first-line

```
h1.achtung : before { content : "Achtung!! "; color : red }
```

PK: :active, :first-child, :last-child, :focus, :hover, :lang, :link, :visited

```
li : first-child { color : #ff0000; }
```

PS:

```
#id {
  position : relative (fixed, absolute, static)
  left (/ top, bottom, right) : 100px;
}
```