Markup und Tags

- Alles, was zwischen < und > steht, ist ein **Tag** (<html>, , …)
- Öffnende Tags (<html>) und schließende Tags (, </html>) werden mit einem / gekennzeichnet.
- Ein Tag markiert Text
- HTML: Hypertext Markup Language

<title> Hier steht der Seitentitel</title>

Öffnendes Tag Tag-Body Schließendes Tag

Struktur einer HTML-Seite

```
<!DOCTYPE html>
<html>
<!-- Meine erste Webseite -->
<head>
	<title>Meine erste

	Webseite</title>
</head>
<body>
	<p>Hallo Web!</p>
</body>
</html>
```

Doctype-Deklaration: Hinweis an den Browser
„Ab jetzt kommt HTML!“

Root-Tag (<html> ist ein MUSS und davor darf nur der Doctype stehen

Kommentare (<!- - Bla bla --> werden vom Browser ignoriert. Ein guter Ort für (Formular-) Passwörter?

Head-Bereich: alles, was mit der Seite zu tun hat, aber nicht zum Inhalt der Seite gehört.

Im Body steht der Inhalt der Seite.

<p> steht für Paragraph, also Textabsatz

Nach </html> darf nichts mehr kommen.

Styling mit CSS

- Neues HTML-Tag:
- Hat keine semantische Bedeutung
- Funktioniert in jedem Browser
- Markiert den enthaltenen Text (Text Text)
- Das style-Attribut enthält 1:n Style-Eigenschaften

Aufgabe:
Markieren Sie einen beliebigen Text mit und betrachten Sie das Ergebnis im Browser.

Span style

Aufgabe:
Fügen Sie diese Zeile in Ihr -Element ein.

Schöner Text

Style: enthält die Formatangaben
für das HTML-Element

Style-Eigenschaften bestehen aus
einem Namen, gefolgt von einem
Doppelpunkt und einem Wert.

Jede Style-Eigenschaften wird mit einem
Semikolon abgeschlossen

Inline-Styling lieber nicht verwenden!

Stylesheets

Aufgabe:
Übertragen Sie die Style-Angaben in eine .css-Datei und binden Sie diese ein.

- Können im <head>-Bereich oder in einer separaten Datei eingebunden werden
- Stylesheet-Dateien enden typischerweise auf .css
- Im <head>-Bereich mit <style>
- Um eine CSS-Datei einzubinden: <link>

<link href="styles.css" rel="stylesheet" />

Styles werden von außen nach innen wichtiger. Eigenschaften können somit Überschrieben werden.

Selektoren

Aufgabe (schwer):
Stylen Sie ein beliebiges Element über eine id.

- Tag-Selektor funktioniert in allen Browsern
- Die einfachsten Selektoren wählen **alle Tags einer Art** aus
- Selektor + 1:n Eigenschaften heißt Style- oder CSS-Regel
- Alle Eigenschaften, die zu einem Selektor gehören, werden in geschweiften Klammern gefasst
- Style-Eigenschaften werden wie in den Inline-Styles definiert
- Selektieren nach id funktioniert wie genauso wie Links zu einer #Sprungmarke

```
span {
        color: red;
        background-color: black;
}
```

Klassen-Selektor

- Genauer als Tag-Selektor, gröber als id-Selektor
- Klassen können beliebig oft in einem Dokument vorkommen
- Der Klassen-Selektor besteht aus einem Punkt
- Klassennamen (und ids) müssen mit einem Buchstaben anfangen (danach Ziffern, _ und – sind erlaubt)
- Mehrere Klassen sind erlaubt und werden durch ein Leerzeichen getrennt

```
.studium {
        color: blue;
}
```

```
<p class="studium">Rund ums Studium</p>
```

RGB

- Red, Green, Blue – Format
- Jede Komponente ist zweistellig als hexadezimale Zahl angegeben
- Werte von 0-9 sowie A-F
- 3B = 3*16 + 11 = 59; AF = 10*16+15 = 175 (FF = 255)
- Je höher der Wert, desto intensiver ist die Komponente in der Farbe vertreten

A = 10, B = 11, C = 12,
D = 13, E = 14, F = 15

Background

- background-image: url(bilder/bild1.jpg);
- background-position, background-repeat

Aufgabe
Definieren Sie ein Hintergrundbild für den gesamten Body Ihres HTML-Dokuments

Pseudoklassen

Aufgabe:
Ändern Sie die Link-Farben per CSS

- Links (<a>) verändern ihren Zustand (vor dem ersten Besuch in blau, danach lila)
- Pseudoklasse :visited, :hover
- Werden im Stylesheet mit einem Doppelpunkt angegeben

```
a, a:visited { color: blue }
```

Verschachtelungen

Legen Sie eine neue HTML-Seite an, mit einem Absatz und mehreren -Tags in diesem Absatz.
Legen Sie für den Absatz eine Schriftfarbe fest, zum Beispiel Grün.
Anschließend legen Sie für die einzelnen <span—Tags unterschiedliche Hintergrundfarben fest.

Diskutieren Sie das Ergebnis!

Lösung

```
<style>
        p {color: #00ff00;}
        #span1 {background-color: #000;}
        #span2 {background-color: #eee;}
</style>
```

```
<p>Irgendwas <span id=“span1“> mit </span> meinen
<span id=“span2“> Spans </span></p>
```

Font-size

- Absolute Größen: Pixel (px) oder Points (pt) – pt = 0,35mm
- xx-small, x-small, small, medium, large, x-large, xx-large (keine genauen Größen festgelegt!)
- Die Größe eines Pixels ist nicht konstant!
- Points passen nicht mit anderen Größenangaben in CSS zusammen

- Relative Größen: 120% bzw. 1.2em
- Macht die Schrift gegenüber dem umgebenden Element um 20% größer

Es gibt keine genaue Umrechnung zw. px und pt. Wie viele Pixel einem Point entsprechen, hängt von Größe und Auflösung des Bildschirms ab.

Font-Size 2

- Fügen Sie Ihrer HTML oder CSS-Datei eine neue Regel für Überschriften hinzu.
- -Tags in Überschriften sollen 10% größer dargestellt werden.

Nachfahren-Selektor

- Der Selektor sucht nur in der entsprechenden Umgebung
- Funktioniert in ALLEN Browsern
- Elemente, die direkt in einem anderen Element enthalten sind, nennt man Kinder (children)
- Elemente, die in einem anderen Element enthalten sind, egal, wie tief verschaltet, nennt man Nachfahren (descendants)

```
h1 .studium {
        color: red;
        font-style: italic;
}
```

Kind-Selektoren

- Ähnlich zu Nachfahren-Selektoren
- Funktioniert in ALLEN Browsern
- Beispiel: h1 > span
- Selektiert alle „span“, die Kinder eines h1 sind

Übung

- Definieren Sie in einem Stylesheet, dass alle gegenüber ihrer Umgebung (z.B. <p>) doppelte Schriftgröße haben sollen
- Definieren Sie danach eine Klasse, die diese Regel wieder aufhebt und der Text wieder eine „normale“ Größe hat

Erklären Sie

- div span (1)
- div span.wichtig (2)
- div p span (3)
- #seitenkopf > span (4)
- .wichtig > p > span (5)
- .wichtig > .wichtig > .wichtig (6)

Lösung

- (1) alle spans innerhalb eines div. Egal, wie tief verschaltet ist
- (2) alle span mit der Klasse wichtig innerhalb eines divs
- (3) spans in einem p in einem div
- (4) spans innerhalb des Elements mit der id seitenkopf
- (5) spans, die direkte Nachfahren eines p sind, das wiederum direkter Nachbar eines Elementes mit der Klasse wichtig ist
- (6) Elemente mit der Klasse wichtig, die von Elementen mit der Klasse wichtig sind, die auch wieder Kinder von Elementen mit der Klasse wichtig sind ;-)

Listen

```
…
<body>
        <p>
                Theoretische Informatik <br/>
                Softwareentwicklung <br />
                Mathematik <br />
                Datenbanken <br />
        </p>
</body>
…
```

Eine gute Idee?

Listenelemente

- (unordered List)
- (ordered List) – aber NICHT sortiert!
- markiert jedes Element mit dem gleichen Zeichen
- zählt die Elemente fortlaufend
- Mit <li> (list item) werden die Elemente einer Liste verpackt

Übung: Erstellen Sie eine nummerierte Liste mit drei Elementen.

Tabellen

- <table> für zusammengehörige Daten, die eine Bedeutung haben
- <tbody> umschließt den eigentlichen Tabellen-Inhalt
- Tabellen sind zeilenorientiert!
- <tbody> hat als Kind-Elemente eine oder mehrere Tabellenzeilen (<tr>)
- Jede Zeile enthält eine oder mehrere Tabellenzellen (<td>)

Tabellen – Achtung!

- Kein Element für Spalten!
- Es gibt nur Zeilen und Zellen!
- Der Browser sorgt in der Darstellung dafür, dass die jeweils erste Zeile jeder Zeile zu einer Spalte wird, die jeweils zweite Zelle zu einer Spalte!
- Alle Zeilen MÜSSEN die gleiche Anzahl an Zellen haben!
- Auch leere Zellen müssen unbedingt angegeben werden!

Tabellen - Übung

Web-Technologien	CPWT	28 LV	60 Min.	5 ECTS
Datenbanken	DB	34 LV	120 Min.	10 ECTS
Theortische Informatik	TI	40 LV	120 Min.	10 ECTS
Mathematik	MA	120 LV	240 Min.	25 ECTS

Bauen Sie diese Tabelle mit HTML nach! Achtung! Die Creditpoints sind rechtsbündig ;-)

Tabellen - Lösung

```
…
<table>
 <tbody>
  <tr>
   <td>Web Techn..</td>
   <td>CPWT</td>
   <td>28 LV</td>
   <td>60 Min.</td>
   <td style=“text-align: right;“>5 ECTS</td>
  </tr>
  …
 </tbody>
</table>
```

Formulare

- Bisher: vom Server zum Client, oder?
- Jetzt: Informationen zum Server senden, z.B. für
  - Login-Masken
  - Kontaktformulare
  - Wikipedia-Seiten
  - …
- Wir benötigen: <form>-Tag!

Dieser Teil erfordert einen eigenen WebServer (z.B. XAMPP).

<form>-Tag

- Action gibt an, wohin die Daten gesendet werden
- Method gibt an, wie die Daten versendet werden sollen
- Accept-charset gibt den Character-Encoding an

```
<form action=“auswertung.php“ method=“get“ accept-charset=“utf-8“>
```

<input>-Tags

- Input deckt fast alle Arten von Eingaben ab.
- Der einfachste Typ: type=“text“ für ein einfaches Eingabefeld
- <input type=“text“ name=“vorname“>
- Der Name ist auch der Name, unter dem der Server diese Daten abrufen kann
- Funktioniert in ALLEN Browsern