

Übungsblatt 3: HTML und Cascading Style Sheets

Vorstellung in den Tutorien am 6. November 2015

Hinweis: Verwenden Sie für alle Programmieraufgaben Mozilla Thimble¹.

3.1 HTML Musikdatei einbinden (1 Punkt) ✓

Binden Sie in eine HTML-Seite eine Audiodatei ein, die automatisch startet, wenn die Webseite geladen ist. Wenn die Audiodatei abgespielt wurde, soll sie automatisch wiederholt werden. Der Besucher sollte außerdem die Möglichkeit haben, das Abspielen zu unterbrechen.

Hinweis: Da in Thimble keine Audio-Datei hochgeladen werden kann, nutzen Sie eine externe Quelle. Zum Beispiel können Sie in ihrer Owncloud² (automatisch von jedem Studenten der TUB nutzbar) eine Audiodatei als Link freigeben und die entsprechende URL dann einbinden.

¹<https://thimble.mozilla.org>

²<https://tubcloud.tu-berlin.de/>

Cascading Style Sheets



Sie beginnen nun mit CSS. Lesen Sie sich für die grundlegende Verwendung die Vorlesungsunterlagen durch. In dieser Übung sind die verschiedenen Selektoren und das Box-Modell wichtig – die Positionierung von Elementen wird noch nicht abgefragt. In den Quellcode-Beispielen finden Sie außerdem häufig verwendete Eigenschaften, die ebenfalls interessant sind. Die gegebenen Folien sind jedoch nicht ausreichend, um alle Aufgaben vollständig zu lösen. Suchen Sie sich daher selbstständig weitere Quellen oder benutzen Sie die Standards.

3.2 CSS Eigenschaften (2 Punkte)

Finden Sie die CSS-Eigenschaften um die folgenden Aspekte im HTML-Dokument ändern zu können. Informieren Sie sich auch über die möglichen Wertbelegungen jeder Eigenschaft. Benutzen Sie zum Testen ein selbsterstelltes HTML-Dokument.

- Schriftart und -familie
- Schriftstil (Kursiv und Fett)
- Schriftgröße und Zeilenabstand
- Weitere Schriftveränderungen: Kapitälchen (kleine Großbuchstaben), Unterstreichungen, Durchgestrichen, Text komplett groß schreiben, Text komplett klein schreiben, nur den ersten Buchstaben eines jeden Wortes groß schreiben
- Textfarbe
- Textausrichtung
- Hintergrundfarbe und Hintergrundbilder
- Marker einer Liste verändern
- Übergänge zwischen zwei Zuständen eines Elements definieren (<transitions>)

3.3 CSS Prioritäten und Vererbung (3 Punkte)



Es gibt mehrere unterschiedliche Möglichkeiten CSS-Anweisungen einzubinden (externe Datei, im `<head>`-Bereich oder inline im `style`-Attribut). Mittels verschiedener Selektoren werden die gewünschten Elemente ausgewählt, für die dann die Eigenschaften angegeben werden. Einige dieser Eigenschaften werden automatisch an Kind-Elemente weitervererbt, andere nicht. Arbeiten Sie zum Verständnis dieser Aufgabe folgendes Dokument durch: <http://www.w3.org/TR/CSS21/cascade.html>.

In dieser Aufgabe geht es darum eine erweiterte Liste über die Priorisierung von CSS-Regeln zu erstellen. Finden Sie die Priorisierungen für folgende Sachverhalte heraus, indem Sie es selbst ausprobieren:

- externe Stylesheet-Datei mittels `<link>` eingebunden
- Style mittels `<style>` definiert
- Style im `style`-Attribut definiert
- Anweisungen mit `!important` definiert
- Vererbte Eigenschaften
- Positionierung innerhalb einer Style-Datei
- Unterscheidung der Priorisierung bei folgenden Selektoren:
 - Universalselektor
 - Elementselektor
 - ID-Selektor
 - Klassenselektor
 - Nachfahrselektor
 - Attributselektor
 - Pseudoklasse `:hover`
 - Pseudoklasse `:not`

Machen Sie sich darüber Gedanken, wann Sie welche Form verwenden würden. Für diese Aufgabe können Sie nachstehenden Quelltext nutzen und erweitern:

```
1 ...
2 <body>
3   <main>
4     <section id="firstSection" class="sections">
5       <h1>Mozilla Thimble</h1>
6
7       <p id="firstTextPart" class="text">Make something <strong>amazing</strong> with
        the web!</p>
8     </section>
9   </main>
10 </body>
```

3.4 Box-Modell (2 Punkte)



Wenn der Browser ein HTML-Element darstellt, wird Platz benötigt. Das Box-Modell besagt, dass es für jedes Element vier Bereiche gibt, die Raum einnehmen: *Margin*, *Border*, *Padding* und das *Element* selbst. Arbeiten Sie sich in dieses *Box-Modell* ein, indem Sie folgende Aufgaben durchführen:

- Versuchen Sie alle Bereiche des Box-Modells einzufärben. Was ist möglich und was fällt Ihnen auf?
- Erstellen Sie eine Liste mit einem einleitenden Text und mindestens 6 verschiedenen Listenelementen. Entfernen Sie den Abstand zwischen einleitendem Text und erstem Listenelement. Bringen Sie jedes Listenelement auf gleiche Höhe, wie den einleitenden Text. Nun soll jedes zweite Listenelement rechtsbündig ausgerichtet werden, von einem gestrichelten Rand umgeben sein und einen Abstand von 20px zum nächsten Listenelement aufweisen.
- Positionieren Sie zwei Bilder in einem `<figure>`-Element nebeneinander und vergrößern Sie die Abstände zwischen den Bildern. Fügen Sie einen ordentlichen Rahmen hinzu. Setzen Sie außerdem eine Bildunterschrift, die für beide Bilder gilt. Versuchen Sie diesen Text lediglich durch die möglichen Bereiche des Box-Modells mittig zwischen den beiden Bildern zu positionieren.

3.5 Button stylen (1 Punkt)



Stylen Sie den folgenden Button, damit die gegebenen Anforderungen erfüllt sind.

```
1 <button name="myData" type="submit">
2   Klick mich
3 </button>
```

Anforderungen – der Button muss...

- eine weiße, fette Schrift mit etwas Schatten, ohne Serifen, in der Größe 16px und einen Zeilenabstand von 30px haben
- einen Abstand zwischen Text und Rand von Links/Rechts 20px und Oben/Unten 10px haben
- abgerundete Ecken haben
- einen verschwommenen Schatten haben
- einen Hintergrund mit blauen Farbverlauf haben (`linear-gradient()`). Dabei soll es Unten dunkler sein als Oben.
- ein visuelles Feedback geben, wenn er geklickt wird. Dabei soll sich der Farbverlauf umdrehen.

3.6 Listen und Überschriften nummerieren (1 Punkt)



In dieser Aufgabe sollen Sie jede Überschrift oder jedes Listenelement (derselben Liste) nummerieren. Nutzen Sie verschiedene Ebenen (z.B. `<h1>`–`<h6>`). Pro Ebene soll die Nummerierung neu starten und – durch einen Punkt separiert – an die Nummern der darüber liegenden Ebene(n) angehängt werden. Wenn Sie zum Beispiel drei Ebenen haben, könnte die Nummerierung wie folgt aussehen: 1 – 1.1 – 2 – 3 – 3.1 – 3.2 – 3.2.1 – 3.2.2 – 3.2.3 usw. Jede Ebene soll im Vergleich zur übergeordneten Ebene etwas eingerückt sein.

Nutzen Sie zum Lösen dieser Aufgabe CSS-Counter und das Pseudoelement `::before`. Auf welche Schwierigkeiten stoßen Sie?

Hinweis: Sie dürfen sich selbst aussuchen, ob sie diese Aufgabe anhand von Listen oder Überschriften lösen.