

## Praktikum 5 (Mehr Styling)

Im Rahmen des Praktikums entwickeln wir eine Web-Anwendung, welche wir Schritt für Schritt mit weiteren Anforderungen, Funktionen und Technologien erweitern.

**Hinweis zu Bonuspunkten:** Zur Vergabe der Bonuspunkte werden wir den von Ihnen produzierten Quellcode begutachten, sobald Sie bestimmte *Meilensteine* erreicht haben. Sobald ein Praktikum einen solchen Meilenstein markiert, finden Sie einen entsprechenden Hinweis auf dem jeweiligen Praktikumsblatt. Dieser Hinweis beinhaltet auch, wieviele Bonuspunkte für den jeweiligen Meilenstein erzielbar sind. Da die Praktika aufeinander aufbauen, enthält ein Meilenstein jeweils alle vorhergehenden Praktikumsaufgaben - zur Erlangung aller Bonuspunkte sind also alle Aufgaben zu bearbeiten. Um die Bonuspunkte zu erhalten, sprechen Sie uns einfach im Praktikum an, sobald Sie einen Meilenstein erreicht haben. Wir schauen uns dann gemeinsam Ihren Stand in einer kleinen Abnahme an. Dabei muss Ihre Gruppe, in welcher Sie die Aufgabe gelöst haben, vollständig anwesend sein. Insgesamt gibt es drei Meilensteine, und es sind maximal 10 Bonuspunkte über das Praktikum erreichbar. *Wichtig:* Abnahmen erfolgen nur bis maximal 20.01.23 (Ende der Vorlesungszeit)!

### Hinweis zum entstehenden Code:

- Zur Verwaltung und zum kollaborativen Bearbeiten Ihres Quellcodes empfehlen wir Ihnen die Nutzung von Git (z.B. in Verbindung mit dem [GitLab-Server des FB4](#), Login per FH-Account). Eine kurze Einführung in Git finden Sie [hier](#).
- Falls Sie Git nicht verwenden, so legen Sie Ihren Quellcode pro Praktikumsaufgabe in separaten Verzeichnissen (z.B. „praktikum2“, „praktikum3“) ab, damit die einzelnen Entwicklungsschritte bei den Abnahmen ersichtlich sind.

## Aufgabe 1: Styling der Formulare

---

1. Modularisieren Sie alle CSS-Regeln für Formulare, indem Sie diese in einer neuen Datei `forms.css` im Verzeichnis `assets/css` ablegen. Binden Sie das neue Stylesheet an den geeigneten Stellen (und nur an diesen!) in der Web-Anwendung ein.
2. Definieren Sie eigene CSS-Regeln für die Darstellung ungültiger Eingaben in ihren Formularen. Dabei sollen nicht valide Zustände (z.B. nicht erlaubtes Datum) in den

Eingabeelementen visualisiert werden (z.B. durch eine entsprechende Hintergrund- oder Rahmenfarbe - dies dürfen Sie frei wählen).

3. Sorgen Sie dafür, dass Eingabeelemente (mit Ausnahme von `select` und `textarea`), sobald sie den Fokus bekommen (z.B. durch Anklicken), durch Änderung der Hintergrundfarbe hervorgehoben werden.
4. Definieren Sie eine *CSS-Regel*, die hinter Eingabeelemente für erforderliche Eingaben ( `required` ) ein rotes Sternchen ( " `*` " ) schreibt. Die Zeichenkette " `*` " soll dabei mit CSS-Mitteln eingefügt werden, d.h. sie darf *nicht* im HTML-Code stehen!
5. Stylen Sie die Schaltflächen zum Absenden von Formularen, mindestens durch (weitere Deklarationen sind erlaubt):
  1. Änderung der Hintergrundfarbe
  2. Setzen runder Ecken
  3. Hinzufügen eines Hover-Effektes
6. Versehen Sie die logischen Gruppen ( `fieldset` ), falls nicht schon geschehen, mit Beschriftungen. Nutzen Sie die Eigenschaften des Box-Modells, um die Gruppen, Beschriftungen sowie die enthaltenen Eingabeelemente (mit Ausnahme von `select` und `textarea` ) wie folgt zu stylen:

Verantwortliche Personen

Autoren:

z.B. Yi Jie Jen,Anke Sch

Besitzer\*in - Name:

z.B. Zainab Nasira Attia \*

Besitzer\*in - E-Mail:

z.B. zainab@attia.com \*

Letztes Update

Datum:

TT . MM . JJJJ

Uhrzeit:

-- : --

"Verantwortliche Personen" und "Letztes Update" sind dabei Beispiele für logische Gruppen mit Beschriftungen. "Autoren", "BesitzerIn - Name", "BesitzerIn - E-Mail", "Datum" und "Uhrzeit" sind Eingabeelemente. Die roten Unterstreichungen sind ein Beispiel für eine Markierung ungültiger Eingaben gemäß 1.

Berücksichtigen Sie insbesondere folgende Stylings auf Basis des Box-Modells, die in der obigen Darstellung erkennbar sind:

1. Die Beschriftungen der logischen Gruppen besitzen nur links und rechts einen Rand sowie runde Ecken.

2. Die logischen Gruppen besitzen nur oben einen Rand.
3. Die Eingabelemente (mit Ausnahme von `select` und `textarea`) besitzen nur unten einen Rand.

Die konkret verwendeten Farben dürfen Sie jeweils frei wählen.

## Aufgabe 2: Styling weiterer Teile der Web-Anwendung

---

Fügen Sie die folgenden Stylings der bereits bestehenden Datei `style.css` hinzu.

1. Heben Sie den *Bereich mit den Zusatzinformationen* ("Neue Tutorials") auf allen Seiten hervor, und zwar mindestens durch (weitere Deklarationen sind erlaubt):
  1. Änderung der Hintergrundfarbe
  2. Setzen runder Ecken
  3. Setzen eines sinnvollen Innenabstandes
  4. Hervorheben des Randes
2. Stylen Sie die Hyperlinks im Navigationsbereich so, dass diese als einzelne Boxen in folgender Form dargestellt werden:

[Abonnierte Podcasts](#) [Details zum Podcast](#) [Details zur Episode](#) [Podcast hinzufügen](#)

Geben Sie den Boxen zusätzlich einen Hover-Effekt. Sobald sich der Mauszeiger über einer Box befindet, so soll sich die Hintergrundfarbe der Box anpassen. Die Farben dürfen Sie frei wählen.

3. Formatieren Sie die Tabelle mit den "Episoden zum Podcast" (auf der *Detailseite zu einem Podcast*, `podcast.html`) so, dass *jede ungerade Zeile im Tabellen-Body sowie der Tabellenkopf* farblich hervorgehoben sind. Der Tabellenkopf soll dabei in einer anderen Farbe hervorgehoben sein als die Zeilen im Tabellen-Body.

Die Formatierung soll auch dann noch funktionieren, wenn im HTML-Code weitere Tabellenzeilen hinzugefügt werden. Zudem soll die Formatierung *nur diese spezielle Tabelle* betreffen - falls wir weitere Tabellen in die Web-Anwendung einbauen, sollen diese nicht so formatiert werden.

4. Legen Sie als Schriftart für die gesamte Seite "Helvetica" fest. Falls diese nicht verfügbar ist, soll "Arial" verwendet werden, sonst eine andere Schriftart ohne Serifen.
5. Fügen Sie über eine CSS-Regel vor *jedem* E-Mail-Link (wie z.B. *Detailseite zu einem*

Podcast, `podcast.html` ) ein kleines Bild eines Briefumschlages ein. *Tipp:* Skalieren Sie das Bild vorher mit einem Grafikprogramm auf die gewünschte Größe. Als Alternative zu einem Bild können Sie auch das Unicode-Zeichen "✉" verwenden.

## Allgemeine Hinweise:

---

- Validieren Sie Ihren HTML-Code mit dem [W3C Markup Validator](#). Der entstehende HTML-Code soll beim Check keine Fehler und Warnungen produzieren.
- Ausnahme: Warnungen bzgl. `date` - und `time` -Eingabefeldern können Sie ignorieren.
- Validieren Sie Ihren CSS-Code mit dem [W3C CSS Validation Service](#). Der entstehende CSS-Code soll beim Check keine Fehler und Warnungen produzieren.