

# Praktikum 5 (Weiteres Styling, Positionierung)

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 18.01.24 (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 die folgenden 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 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 valider Name eines Semesterplanes) 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. 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
5. Definieren Sie eine *CSS-Regel*, die hinter Eingabeelemente für erforderliche Eingaben ( `required` ) die Zeichenkette "(erforderlich)" schreibt. Die Zeichenkette "(erforderlich)" soll dabei mit CSS-Mitteln eingefügt werden, d.h. sie darf *nicht* im HTML-Code stehen!
6. Versehen Sie die logischen Gruppen Ihrer Formulare ( `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:

### Neuen Semesterplan erstellen: Schritt 2

**BASISDATEN**  
**Studiengang: Wirtschaftsinf. BPO 2018**  
**Name:**  (erforderlich)

**SEMESTER**  

Sommersemester

**Jahr:**

2022

 (erforderlich)

**KURSE**  

☐ Digital Markets und Innovation [1. Semesterhälfte] V A-P (Mon 1. Stunde)

☐ Web-Technologien Ü I-L (Mon 3. Stunde)

☐ Mobile App Engineering [StgPO: PI/TI-19, MI-19, WI-18] P A-H (Tue 9. -10. Stunde)

☐ Lineare Algebra (WI) (Hessel-von Molo) T A-P (Tue 9. -10. Stunde)

☐ Web-Technologien V A-P (Wed 5. -6. Stunde)

**Erstellen**

"Basisdaten", "Semester" und "Kurse" sind dabei Beispiele für logische Gruppen mit Beschriftungen. "Name", "Jahr" usw. sind Eingabeelemente. Die roten Ränder sind ein Beispiel für eine Markierung ungültiger Eingaben gemäß 2.

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

1. Die Beschriftungen der logischen Gruppen besitzen runde Ecken und sind farblich hervorgehoben. Zudem ist der enthaltene Text in Großbuchstaben geschrieben (lösen Sie dies *ausschließlich* mit CSS!),
2. Die logischen Gruppen besitzen nur oben und links einen Rand. Zudem haben Sie runde Ecken.
3. Die Eingabelemente (mit Ausnahme von `select` und `textarea`) besitzen nur oben und links einen Rand.

Die konkret verwendeten Farben dürfen Sie frei wählen.

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

---

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

1. Heben Sie den *Bereich mit den Zusatzinformationen* ("Aktuelle Meldungen") 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:

Liste der Semesterpläne Details zum Semesterplan Details zum Kurs Neuen Semesterplan erstellen...

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

3. Formatieren Sie die Tabelle der gewählten Kurse auf der *Detailseite zum Semesterplan* (`plan.html`) so, dass *jede zweite Zeile sowie der Tabellenkopf* farblich hervorgehoben ist. Der Tabellenkopf soll dabei in einer anderen Farbe hervorgehoben sein als die Tabellenzeilen. Zudem sollen alle Ränder der Tabelle (Innen- und

Außenränder) sichtbar sein.

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.

## Aufgabe 3: Logo, Scrolling, Fußbereich (Positionierung)

---

1. Positionieren Sie das Logo und den Titel *auf allen Seiten* der Webseite *nebeneinander*. Den verwendeten Mechanismus für die Positionierung können Sie frei wählen.
2. Realisieren Sie (nur mit HTML und CSS!) folgendes Scrollverhalten für den Kopfbereich *auf allen Seiten*: Kopfbereich (mit Logo und Titel) sowie der Navigationsbereich sollen zunächst scrollbar sein. Sobald der Kopfbereich jedoch "weggescrolled" wurde, also nicht mehr sichtbar ist, soll der Navigationsbereich am oberen Rand fixiert bleiben und nicht mehr mitscrollen.
3. Sorgen Sie dafür, dass der Fußbereich mit dem Copyright-Hinweis *auf allen Seiten* der Webseite immer fest am unteren Rand angezeigt wird - er soll nicht "weggescrolled" werden können.

## Allgemeine Hinweise:

---

- Validieren Sie Ihren HTML-Code mit dem [W3C Markup Validator](#). Der entstehende HTML-Code soll beim Check keine Fehler und Warnungen produzieren.
- Validieren Sie Ihren CSS-Code mit dem [W3C CSS Validation Service](#). Der entstehende CSS-Code soll beim Check keine Fehler und Warnungen produzieren.