

Praktikum 7 (Responsive Layout, JavaScript, CTF)

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

Meilenstein 2: "CSS"

Sobald Sie Praktikum 4, 5, 6 und 7 vollständig bearbeitet haben, haben Sie Meilenstein 2 erreicht! Sprechen Sie uns (Sven Jörges, Andreas Harrer oder Julian Feder) im Praktikum an. Wir schauen uns dann gemeinsam Ihren Stand in einer kleinen Abnahme an. Dabei muss Ihre Gruppe, in welcher Sie die Praktika bearbeitet haben, vollständig anwesend sein. Für diesen Meilenstein sind insgesamt 3 Bonuspunkte erreichbar. Es folgt noch ein weiterer Meilenstein.

Hinweis zum entstehenden Code:

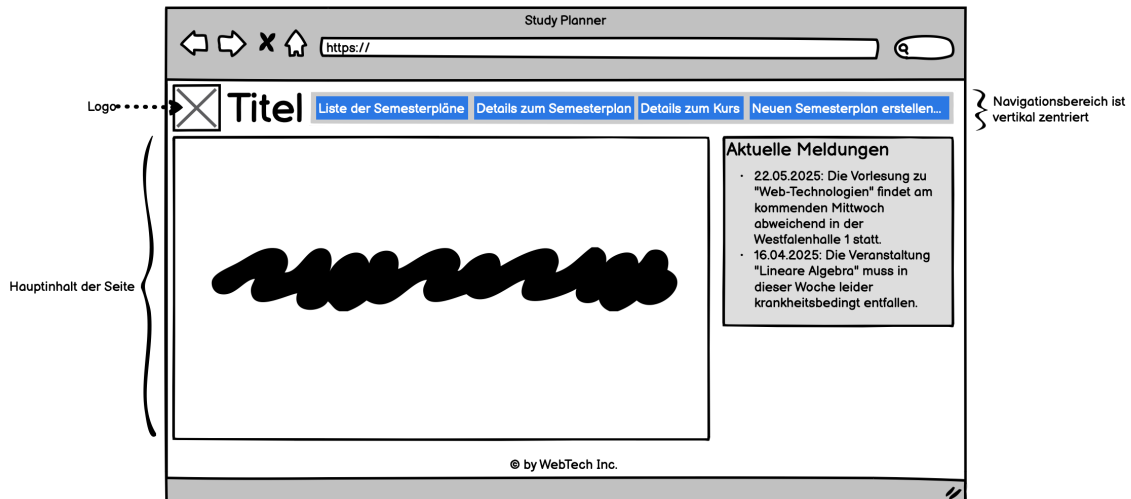
- Zur Verwaltung und zum kollaborativen Bearbeiten Ihres Quellcodes empfehlen wir Ihnen die Nutzung von Git (z.B. in Verbindung mit dem [GitLab-Server des FB4](#), Login per FH-Account). Eine kurze Einführung in Git finden Sie [hier](#).
- Falls Sie Git nicht verwenden, so legen Sie Ihren Quellcode pro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

Aufgabe 1: Responsive Seiten-Layouts

Erweitern Sie die Flexbox-Layouts für die Seiten *Liste der Semesterpläne* (`index.html`) und *Detailseite zum Kurs* (`kurs.html`) sowie die Grid-Layouts für die *Detailseite zum Semesterplan* (`plan.html`) und die Formularseiten (`plan-neu-schritt1.html` , `plan-neu-schritt2.html`) so, dass diese *responsive Layouts* sind. Folgendes soll für alle Seiten gelten:

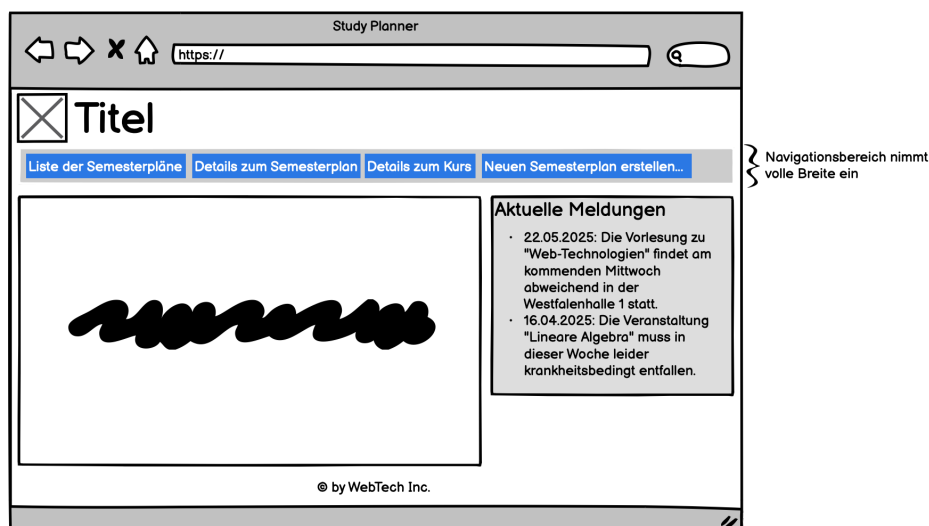
1. Die auf den Seiten verwendeten Layout-Mechanismen (Flexbox oder Grid) sollen jeweils beibehalten werden (also z.B. nicht alle Seiten auf Grid umbauen!), so wie in Praktikum 6 realisiert.

2. Vermeiden Sie doppelte/redundante CSS-Deklarationen. Nutzen Sie die vorhandene Modularisierung der Stylesheets.
3. Bei Änderung der Viewport-Breite soll sich das Layout fluide verhalten und den verfügbaren Platz ausfüllen. Ausnahme: Tabellen sind recht unflexibel und müssen daher nicht fluide sein.
4. Für *Viewport-Breiten* $> 1024 \text{ Pixel}$ soll folgendes Layout gelten:



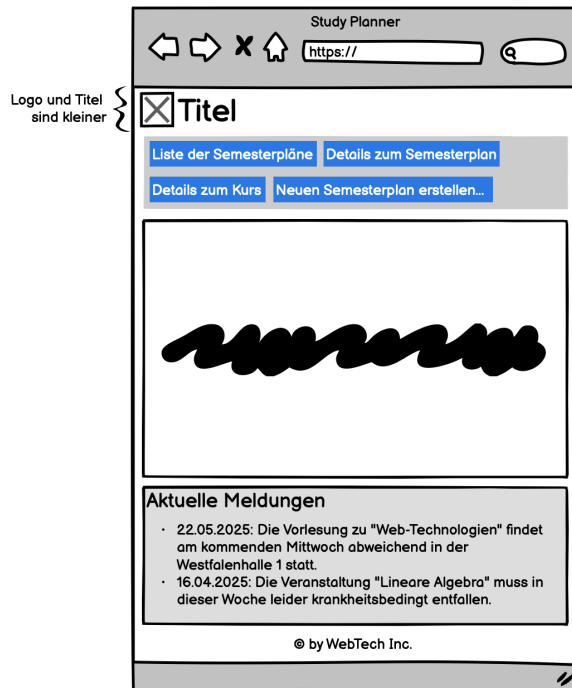
Dabei werden Kopf- und Navigationsbereich nebeneinander dargestellt. Der gesamte Navigationsbereich soll vertikal zentriert sein. Die Navigationslinks (blaue Boxen) sollen weiterhin linksbündig angezeigt werden, wie in Praktikum 6 realisiert.

5. Für *Viewport-Breiten* $\leq 1024 \text{ Pixel}$ soll folgendes Layout gelten:



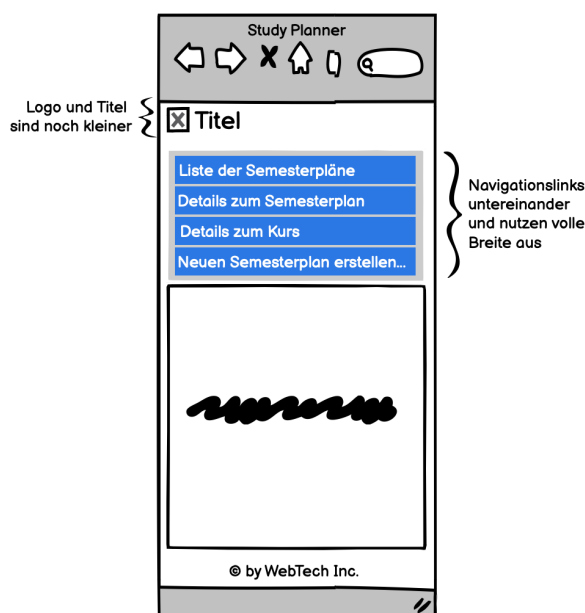
Kopf- und Navigationsbereich werden nun untereinander angezeigt. Der Navigationsbereich nimmt die volle verfügbare Breite ein. Die Navigationslinks (blaue Boxen) werden weiterhin linksbündig angezeigt.

6. Für *Viewport-Breiten* ≤ 768 Pixel soll folgendes Layout gelten:



Dabei werden Logo und Titel verkleinert dargestellt, um Platz zu gewinnen. Außerdem wird der Bereich mit den Zusatzinformationen ("Aktuelle Meldungen") nun unter dem Hauptinhalt angezeigt und nutzt die volle verfügbare Breite.

7. Für *Viewport-Breiten* ≤ 480 Pixel soll folgendes Layout gelten:



Logo und Titel werden nun noch kleiner dargestellt. Die Navigationslinks (blaue Boxen)

werden untereinander angezeigt und nutzen jeweils die volle Breite aus. Der Bereich mit den Zusatzinformationen ("Aktuelle Meldungen") ist vollständig ausgeblendet.

Aufgabe 2: JavaScript-Vorbereitung

Wir bereiten nun das nächste Praktikum vor, indem wir ein einfaches Skript einbinden (`assets/js/script.js`). Gehen Sie dabei wie folgt vor:

1. Binden Sie das Skript als externes JavaScript zunächst auf der Seite *Liste der Semesterpläne* (`index.html`) ein.
2. Geben Sie mittels JavaScript die Breite des Layout-Viewports auf der Konsole aus. Gehen Sie dazu folgendermaßen vor:

1. Verwenden Sie folgende Funktion zum Ermitteln der Viewport-Breite:

```
const getViewportWidth = () => window.innerWidth ||  
                                document.documentElement.clientWidth
```

2. Nutzen Sie `console.log([...])`, ein *Template-Literal* (keine String-Konkatenation mit `+`!) und einen Aufruf der Funktion aus 1., um eine Ausgabe der folgenden Form auf der Konsole zu produzieren:

```
Die Viewport-Breite beträgt: 2560 Pixel.
```

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.

Zusatzaufgabe 2: Capture the Flag (CTF)

Die folgende Aufgabe ist freiwillig – es gibt für diese Aufgabe keine Bonuspunkte (nur Ruhm, Ehre, Spaß, etc. pp.)!

In dieser freiwilligen Zusatzaufgabe gibt es eine neue Challenge vom Typ *Capture the Flag* (CTF, siehe Praktikum 3 für die grundlegende Beschreibung des Prinzips). Auch dieses Mal ist das Flag wieder eine Zeichenkette, die mit `ctf_` beginnt (z.B.

```
ctf_This_is_the_Flag_forma7
```

).

Finden Sie das Flag, das hier versteckt ist:

Challenge 2: "Hexy Colors" (Schwierigkeitsgrad: leicht)

<https://labs.inf.fh-dortmund.de/ctfd-challenge-1/hexy-colors-everywhere.html>

Wenn Sie das Flag gefunden haben, schicken Sie es per E-Mail an sven.joerges@fh-dortmund.de (gerne inklusive Lösungsweg, den Sie gewählt haben). Und nichts verraten!

