

## (Mini-)Praktikum 8 (JavaScript einbinden)

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: JavaScript einbinden

---

Wir legen nun eine erste Grundlage für die folgenden Praktika, 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 *Liste der abonnierten Podcasts* ( `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 (in Pixel):

```
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.
```

3. Ergänzen Sie das Skript, so dass es eine Warnung ausgibt, sobald die Viewport-Breite weniger als 30% der verfügbaren Bildschirmbreite beträgt. Gehen Sie wie folgt vor:

1. Ermitteln Sie die Bildschirmbreite mit dem Ausdruck `screen.width`.
2. Falls die Viewport-Breite weniger als 30% der verfügbaren Bildschirmbreite beträgt, zeigen Sie eine Warnung mit dem Befehl `alert($HierEinenTextEinsetzen)` an.

**Hinweis:** Die Nutzung der `alert`-Funktion ist in produktiven Anwendungen kein guter Stil. Daher dürfen Sie die entsprechende Code-Zeile für die folgenden Praktika auskommentieren, wenn die Meldung lästig wird.

## 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.