

Übungsblatt 1

Veröffentlicht am	11.10.2016
Anzahl der Seiten	3
Anzahl Punkte im Pflichtteil (entspricht maximal erreichbaren Punkten)	5
Anzahl Punkte im Bonusteil	2
Abgabetermin und Demonstration in der Übung	Übungen 25./27.10.16

Anmerkungen

Bitte beachten Sie die folgenden Hinweise zur Bearbeitung der Übungsaufgaben und dem Ablauf im aktuellen Semester.

- Lesen Sie bei einem Übungsblatt stets alle Aufgaben durch, bevor Sie beginnen.
- Nach dem Unterricht wird vor der Übung das ggf. neue Übungsblatt in Moodle veröffentlicht.
- Sofern zum Aufgabenblatt Code-Bausteine (Vorgaben) dazugehören, werden diese ebenfalls auf Moodle zum Download angeboten und sind als Ausgangsbasis bei der Bearbeitung zu verwenden.
- Ihre Lösung der Aufgaben laden Sie *vor* Ihrer persönlichen Demonstration in Moodle hoch. Dateiname: Ü[Nr]__Nachnamen__Matrikelnummern.ZIP
Beispiel: Ü2__Mueller_Meier__12345678__87654321.ZIP
- Erfordern die Teil-Aufgaben eines Übungsblattes, dass Sie mehrere Anwendungen, HTML-Seiten oder Code-Pakete erstellen, dann legen Sie bitte Unterordner in Ihrem ZIP mit den Nummern der Aufgaben an.
- Persönliche Demonstration und Erklärung in der Übung durch alle Gruppenmitglieder. Jedes Gruppenmitglied kann die Abgabe erläutern, sonst keine Punkte.
- Bei den Aufgaben ist jeweils angegeben, ob diese Pflicht- oder Bonus-Aufgaben sind, sowie die maximal erreichbaren Punkte der Teil-Aufgabe.
- Eine Übung gilt als bestanden, wenn mind. 50% der Pflichtpunkte erreicht wurden, sonst gibt es 0 (Null) Punkte.
- Bei verspäteter Abgabe von bis zu maximal 2 Wochen können nur noch 50% der möglichen Punkte des Übungsblattes erreicht werden (bei einer Woche verspäteter Abgabe 75% der möglichen Punkte).

Ziel und Zweck der Übung:

Die Übung beinhaltet eine kurze Umfrage, damit ich als Dozent besser weiß, welche Vorkenntnisse und Interessen Sie mitbringen. Da der Unterricht nicht „fertig“ ist, kann ich so auf Ihre Antworten eingehen und den ME2-Inhalt etwas anpassen.

Hauptziel der Übung: Sie frischen Ihre Kenntnisse zum Zusammenspiel von HTML5/CSS3 und JavaScript auf bzw. erlernen diese. Achten Sie auf saubere, professionelle Metadaten in allen Dateien (Kopfzeilen, Funktionen und Parameter mit JSDoc beschriften). Es ist für diese Übung nicht erlaubt, irgendwelche Frameworks oder fertigen Bibliotheken (wie backbone, jQuery etc) zu verwenden. Die Ordner/Dateistruktur für die Aufgaben sollte wie folgt aussehen (HTML, JS, CSS):

```
▼ css
  app.css
▼ js
  app.js
  index.html
```

Referenzen:

- <http://www.w3schools.com/html/default.asp>
- <http://www.w3schools.com/css/default.asp>
- <http://tools.webdirections.org/#video>
- <https://msdn.microsoft.com/de-de/library/hh924823%28v=vs.85%29.aspx>
- <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>
- <http://www.w3.org/TR/html5/document-metadata.html>
- <https://onpagelove.com/gallery/application>
- <http://www.webmasterpro.de/coding/article/css-responsive-webdesign-media-queries-fuer-iphone-ipad-und-smartphones.html>

Aufgabe 0 (Pflicht, 0.5 Punkte)

Füllen Sie die Moodle Kurzumfrage im Kurs zu ME2 aus.

<https://lms.beuth-hochschule.de/moodle/mod/feedback/view.php?id=290943>

Achtung: Erst ab Di, 11.10.2016 11:30 Uhr verfügbar (nach dem Unterricht Zugl) und auszufüllen bis spätestens Di, 18.10.2016 20:00 Uhr.

Aufgabe 1 (Pflicht, 1 Punkt)

Erstellen Sie ein Standard-konformes HTML5-Seitenskelett das mindestens 3 Videos auflistet. Als Videos können Sie eigene Quellen verwenden oder die Beispiele von <http://techslides.com/sample-webm-ogg-and-mp4-video-files-for-html5/>. Nutzen Sie gerne verschiedene Formate (WebM, mp4/H264 und OGG).

Die Einzelseite (SinglePageApp) sollte mindestens eine Kopfzeile, Fußzeile, sowie einen [Rand]Seitenbereich und einen Hauptbereich haben. Verwenden Sie dafür semantische Tags aus HTML5.

- Im Hauptbereich werden die drei Videos angezeigt/abgespielt.
- Im Seitenbereich können sie zusätzliche Informationen angeben, beispielsweise eine Navigation zu weiteren Seiten, Sprungmarken innerhalb der Seite, etc.

Aufgabe 2 (Pflicht, 2 Punkte)

Hauchen Sie dem Template mittels JavaScript mehr Leben ein und bauen sie eigene mit CSS gestaltete Play/Pause, Stop-Buttons ein, die per JavaScript das jeweilige Video steuern.

Achtung: Das Javascript ist dafür komplett in der externen Datei `app.js` zu schreiben. Binden Sie ihren JavaScript Code so ein, dass er erst ausgeführt wird, wenn das Dokument fertig geladen wurde (Experimentieren sie dazu mit Events `load` und `DOMContentLoaded` auf dem Objekt `window` bzw. `document`). Nutzen Sie `addEventListener()` und vermeiden Sie direkt in HTML irgendwelches JavaScript (wie bspw. mit `onclick=myFunction()`)

Tipp: Flexibilisieren Sie den Code so, dass er für alle Videos der Seite die passenden Buttons findet und per Event „click“ die Buttons mit den Funktionsaufrufen zum Starten/Stoppen des jeweiligen Videos verbindet.

Aufgabe 2 Bonusteil (1.5 Punkte)

Erstellen Sie Ihr JavaScript so flexibel, dass der gleiche Code für alle Videos der Seite die Buttons iterativ initialisiert (bspw. mit `getElementsByClassName()`). Sie könn(t)en dann also ein viertes Video im HTML hinzufügen, ohne dass Ihr JavaScript-Code geändert werden muss. *Tipp: Eine Lösung ohne Verwendung von IDs ist möglich.*

Aufgabe 3 (Pflicht, 1.5 Punkte)

Gestalten Sie mit CSS3 in einem externen Stylesheet namens `app.css` die Abstände, Hintergründe und Rahmen für Kopf, Fuss, Seitenbereich und Hauptbereich.

Aufgabe 3 Bonusteil (0.5 Punkte)

Gestalten Sie die CSS so, dass die Seite auch auf kleinen Bildschirmen (Smartphone-Screens) besser benutzbar ist (Responsive Design). Ob Sie diese Styles mit `@media` in die `app.css` einbauen oder eine neue CSS-Datei in HTML einbinden können Sie wählen.