

Übungsblatt 1

Veröffentlicht am	10.04.2017
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 20./25.04.2017

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 WE2-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>
- https://developer.mozilla.org/en/docs/Web/HTML/Element/picture#Browser_compatibility
- <https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>
- <https://developer.mozilla.org/en-US/docs/Web/Events/mouseover>
- <https://developer.mozilla.org/en/docs/Web/CSS/position>
- <http://www.w3.org/TR/html5/document-metadata.html>
- <https://onpagelove.com/gallery/application>
- https://www.w3schools.com/css/css_rwd_mediaqueries.asp

Aufgabe 0 (Pflicht, 0.5 Punkte)

Füllen Sie die Moodle Kurzumfrage im Kurs zu WE2 aus.

<https://lms.beuth-hochschule.de/moodle/mod/feedback/view.php?id=321326>

Achtung: Für Zug 2 erst ab Di, 11.04.2017 11:30 Uhr verfügbar (nach dem Unterricht Zug2) und auszufüllen bis spätestens Di, 18.04.2017 18:00 Uhr.

Aufgabe 1 (Pflicht, 1 Punkt)

Erstellen Sie ein Standard-konformes HTML5-Seitenskelett das mindestens 8 Bilder auflistet.

Als Bilder können Sie eigene Quellen verwenden oder die Beispiele von <http://lorempixel.com/>.

Nutzen Sie gerne verschiedene Formate (PNG, IMG, GIF, SVG, ..).

- Die Bilder sollten mindestens zwei verschiedene Seitenverhältnisse haben. (Überlegen Sie sich eine passende Anordnung)
- Alle Bilder werden in der Übersicht in gleichen Breite dargestellt (z.B. 236px wie bei Pinterest)
- Stellen Sie maximal 5 Bilder in einer Reihe dar (also mindestens 2 Reihen).

Die HTML-Seite sollte mindestens eine Kopfzeile, Fußzeile, sowie einen [Rand]Seitenbereich und einen Hauptbereich haben. Verwenden Sie auch hier semantische Tags aus HTML5.

- Im Hauptbereich werden die Bilder angezeigt.
- Im Seitenbereich können sie zusätzliche Buttons/Links angeben, beispielsweise eine Navigation zu Sprungmarken innerhalb der Seite, Hinzufügen neuer Bilder (ohne Funktion), ..

Aufgabe 2a (Pflicht, 2 Punkte)

Hauchen Sie der Seite mittels JavaScript mehr Leben ein und bauen sie zusammen mit CSS einen *Like-Button* und einen *Details-Button* ein, welche sichtbar werden, sobald sich der Mauszeiger über ein Bild bewegt (mouseover). Folgende Funktionalität ist zu implementieren: Der Like-Button sollte anklickbar sein und sich seinen Status „like/not liked“ merken und visuell sichtbar sein (via JS/CSS). Der Details-Button sollte das Bild bei einem Klick in Groß in einem Overlay über dem bestehenden Inhalt der Seite zentriert anzeigen (siehe Beispiel in Referenzen).

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()`)

Aufgabe 2b (Bonus, 1Punkt)

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

Aufgabe 2c (Bonus, 0.5 Punkte)

Unterstützen Sie mit JavaScript auch die Touch-Bedienung. Während am Desktop ein Mouseover-Event passend ist, gibt es am Tablet oder Smartphone eher die touchstart/touchend-Events. Wie unterscheiden Sie nun einen Klick von einem per touch auszulösenden mouseover?

Aufgabe 3a (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 3b (Bonus, 0.5 Punkte)

Gestalten Sie die CSS so, dass die Seite auch auf kleinen Bildschirmen (Smartphone-Screens) besser benutzbar ist (Responsive Design). Dabei sollten Sie die Anzahl an dargestellten Spalten der Bildschirmbreite anpassen. Überlegen Sie auch, welche Teile von Kopf-, Seiten-, und Fuss-Bereich sich anders darstellen. Ob Sie diese Styles mit `@media` in die `app.css` einbauen oder eine neue CSS-Datei in HTML einbinden können Sie wählen.