

# Web-Entwicklung: Übungsblatt 6

---

## Aufgabe 1: Anlegen eines npm-Projektes

Legen Sie ein neues npm-Projekt für eine Browser-Anwendung wie in der letzten Übung gezeigt an. Anders als dort, soll hier die CSS-Datei nicht statisch sein, sondern im Zuge des Build-Prozesses aus einer (zunächst leeren) Less-Datei heraus erzeugt werden.

Für die folgenden Übungsaufgaben wird Ihnen ein Datensatz `activities.json` zur Verfügung gestellt. Kopieren Sie diese Datei in den Ordner `src/assets` Ihres Projekts.

Dieser Datensatz besteht aus einem Array von Objekten, welche jeweils eine Aktivität im Laufe eines Tages repräsentieren. Eine Aktivität enthält einen Titel, eine Notiz sowie zwei Zeitstempel, die den Beginn und das Ende der Aktivität darstellen. Darüber hinaus kann jeder Aktivität eine sortierte Menge von sog. Tags zugeordnet sein. Es gibt die Tags *Familie*, *Arbeit* und *Sport*. Beachten Sie, dass es auch Aktivitäten ohne Notiz, ohne Tags oder ohne Ende-Zeitstempel gibt und, dass die Aktivitäten im Datensatz nicht zeitlich sortiert sind.

## Aufgabe 2: Anlegen einer HTML5-Dokumentstruktur mittels DOM-Manipulation

Importieren Sie den Datensatz wie in einer früheren Übung gezeigt.

Sie sollen nun dynamisch mittels DOM-Manipulation eine semantisch korrekte HTML-Dokumentstruktur aufbauen. Überführen Sie dazu den Datensatz in ein HTML-Element (z.B. eine geordnete Liste `ol`) und zeigen Sie dieses in der Browser-Anwendung an. Das Ergebnis sollte in etwa so aussehen:



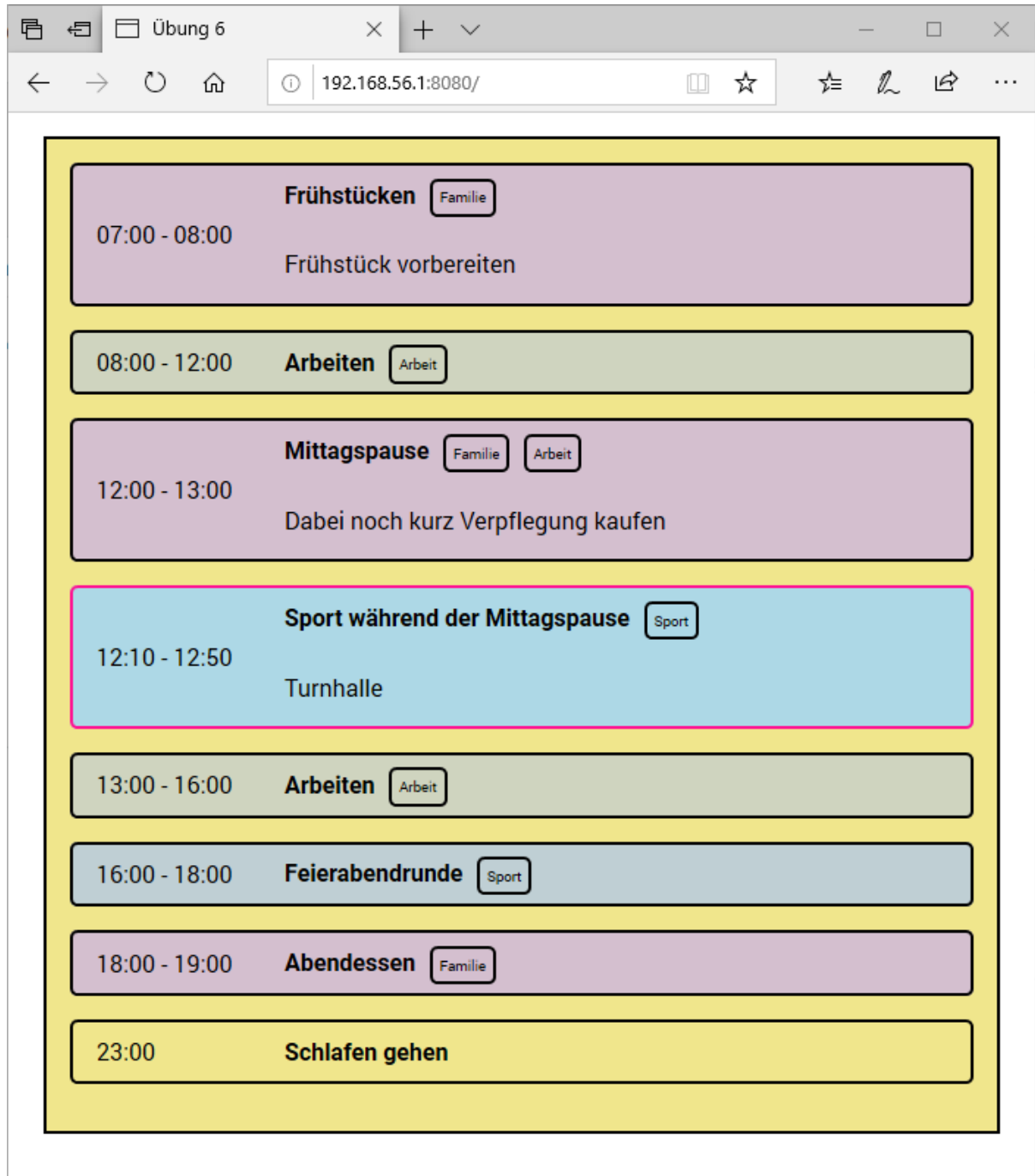
Achten Sie auf eine sinnvolle funktionale Strukturierung Ihrer Anwendung.

Hinweise:

- Verwenden Sie die [Array-Methode sort\(\)](#), um den importierten Datensatz gemäß des Beginn-Zeitstempels aufsteigend zu sortieren. Dazu müssen Sie beim Aufruf von `sort()` eine Funktion übergeben.
- Der Datensatz enthält vollständige Zeitstempel inkl. Datum und Zeitzone. Angezeigt werden soll hingehend nur der Zeiteil im Format HH:MM. Dazu müssen Sie sich mit den Methoden des Standardobjekts [Date](#) auseinander setzen.
- Sollte eine optionale Eigenschaft einer Aktivität nicht gesetzt sein, so fügen Sie auch nicht unnötige leere Elemente in den DOM-Baum ein.

## Aufgabe 3: Styling mithilfe von Less

Setzen Sie nun CSS sowie die Sprach-Features, welche Less Ihnen bietet (z.B. Konstanten, Mixins, Verschachtelungen und vordefinierte Funktionen), sinnvoll ein, um schrittweise ein visuelles Erscheinungsbild der Browser-Anwendung zu erreichen, das in etwa so aussieht:



Ihre Lösung soll die folgenden Anforderungen erfüllen:

- Die Liste der Aktivitäten wird von einer Box umfasst. Diese Box besitzt einen sichtbaren Rahmen sowie eine Hintergrundfarbe, die von der Hintergrundfarbe der Web-Seite abweicht.
- Die Aktivitäten sollen innerhalb dieser Box vertikal (zeitlich aufsteigend sortiert) aufgelistet werden.
- Jede Aktivität wird wiederum von einer Box mit sichtbarem Rahmen und mit abgerundeten Ecken umgeben.
- Das Zeitintervall und die Beschreibung (bestehend aus Titel und Notiz) einer Aktivität werden nebeneinander und vertikal zentriert, Titel und Notiz innerhalb der Beschreibung untereinander und

linksbündig angeordnet.

- Die Beschreibung soll stets den gleichen Abstand vom linken Rahmen einhalten, unabhängig davon, ob der Ende-Zeitstempel gesetzt ist oder nicht (siehe *Schlafen gehen*)
- Die Hintergrundfarbe einer Aktivitätsbox hängt vom ersten zugeordneten Tag ab. Die Farben können frei gewählt werden, allerdings sollen sie den gleichen Sättigungs- und Hellwert aufweisen (siehe [Wikipedia: HSV-Farbraum](#)). Ist kein Tag gesetzt, ist die Box transparent (siehe *Schlafen gehen*).
- Falls die Maus über eine Aktivität bewegt wird, soll diese währenddessen farblich hervorgehoben werden sowie die Farbe des Rahmens sich ändern (Vergleiche *Sport während der Mittagspause* und *Feierabendrunde*).
- Die Tags in jeweils einer Box mit sichtbarem Rahmen und abgerundeten Ecken hinter dem Titel der Aktivität stehen. Dabei soll eine kleinere Schriftgröße zum Einsatz kommen.
- Achten Sie auf einheitliche Abstände.