

## Ausgangssituation

Sie entwickeln eine kleine Web-Anwendung, die Cafés in einer Stadt anzeigt.

Die Anwendung soll:

- eine Liste von Cafés in einer Tabelle ausgeben,
- Filtermöglichkeiten bieten (Bewertung, Entfernung, Stadt),
- einfache Statistiken zum aktuell angezeigten Ergebnis berechnen.

Sie arbeiten dabei mit einem vorgegebenen Datensatz in JavaScript und manipulieren das DOM zur Ausgabe im Browser.

### Schritt 1

Verwenden Sie das HTML-Grundgerüst in index.html. Sie finden außerdem TODO Kommentare in script.js

- Geben Sie das Array `cafes` einmal in der Konsole aus, um sich einen Überblick zu verschaffen.
- Implementieren Sie eine Funktion

```
function renderCafes(cafesArray) {  
    // ...  
}
```

- Diese Funktion soll:
  - die Tabelle mit der id `cafe-table` leeren/überschreiben und
  - eine vollständige HTML-Tabelle aus dem übergebenen Array erzeugen.
  - Anforderungen an die Tabelle:
    - Kopfzeile mit Spalten, z.B.: Name, Stadt, Bewertung, Entfernung, Preis, WLAN, Status.
    - Jede Zeile entspricht einem Café.
    - Für die Entfernung soll „km“ angezeigt werden (z.B. 1.2 km).
    - Für `priceLevel` können Sie z.B. 1–3 „€“-Symbole anzeigen.
    - Für den Status:
      - Wenn `seatsFree > 0`: Text z.B. „Plätze frei (X)“ mit CSS-Klasse `badge badge-open`.
      - Wenn `seatsFree === 0`: Text z.B. „voll“ mit CSS-Klasse `badge badge-busy`.

- Rufen Sie nach der Definition der Funktion einmal

```
renderCafes(cafes);
```

auf, damit beim Laden der Seite alle Cafés sichtbar sind.

### Schritt 2

Sie verbinden nun die Eingabefelder/Buttons mit einer Logik, die das Array der Cafés entsprechend einschränkt.

- Implementieren Sie eine Funktion

```
function applyFilters() {
    // ...
}
```

- Die Funktion soll:
  - die Werte aus den Eingabefeldern lesen:
    - min-rating (Mindestbewertung),
    - max-distance (maximale Entfernung),
    - city-select (Stadt oder „all“),
  - ausgehend von allen Cafés eine neue Liste erzeugen, in der:
    - nur Cafés mit rating  $\geq$  Mindestbewertung enthalten sind,
    - nur Cafés mit distance  $\leq$  maximale Entfernung enthalten sind,
    - nur Cafés der ausgewählten Stadt enthalten sind (oder alle, wenn „all“ ausgewählt ist).
- Speichern Sie das Ergebnis in der Variablen currentCafes und rufen Sie anschließend:

```
renderCafes(currentCafes);
updateStats(currentCafes);
```

- Verbinden Sie den Button mit der id btn-apply mit der Funktion applyFilters
- Implementieren Sie außerdem eine Funktion

```
function resetFilters() {
    currentCafes = cafes;
    renderCafes(currentCafes);
    updateStats(currentCafes);
}
```

und verbinden Sie den Button mit der id btn-reset damit.

Setzen Sie dabei auch sinnvoll die Eingabefelder zurück (z.B. auf die Default-Werte).

### Schritt 3

Erstellen Sie eine Funktion

```
function updateStats(cafes) {
    // ...
}
```

die einige Kennzahlen über das aktuell angezeigte Array berechnet und im Element mit der id stats anzeigt.

Mögliche Kennzahlen:

- Anzahl der angezeigten Cafés
  - z.B. „Anzahl Cafés: 5“.
- Durchschnittliche Bewertung
  - Durchschnitt aus allen rating-Werten der übergebenen Cafés, z.B. auf eine Nachkommastelle gerundet.
- Durchschnittliche Entfernung
  - Durchschnitt der distance-Werte.
- Anzahl Cafés mit freiem Platz
  - Anzahl der Cafés mit seatsFree > 0.
- Rufen Sie updateStats(currentCafes); sowohl nach dem ersten Rendern als auch nach jeder Filteraktion auf.

**Bonus:**

Implementieren Sie eine Funktion

```
function findBestWorkCafe(cafesArray) {  
    // ...  
}
```

Ziel: Wählen Sie aus dem Array genau ein Café aus, das sich besonders gut zum Arbeiten eignet.

Sie können dabei z.B. folgende Kriterien kombinieren:

- WLAN vorhanden (hasWifi === true),
- Bewertung möglichst hoch,
- Entfernung möglichst gering,
- es sollte mindestens ein Platz frei sein (seatsFree > 0).

Geben Sie das gefundene Café z.B. unterhalb der Statistiken kurz textuell aus, z.B.:

„Bestes Café zum Arbeiten: Latte & Code (Bewertung 4.9, 1.2 km, WLAN, 2 Plätze frei)“

Überlegen Sie selbst, wie Sie Kriterien gewichten und wie Sie aus der Liste das „beste“ auswählen.