

Modul 294 - Aufgabenblatt 03

von Lukas Meier

Unterricht vom 09.12.2025

1 Vorbereitung

Erstellen Sie einen neuen Ordner und speichern Sie die Datei `index.html` aus dem Netzwerkordner darin. Öffnen Sie die Datei anschliessend im Code-Editor Ihrer Wahl und fügen Sie ein `script`-Tag ein, falls dieses noch nicht existiert.

2 Erste Funktion

Schreiben Sie eine Funktion `begruessen(name)`, welche eine personalisierte Begrüssung in der Konsole ausgibt. Rufen Sie die Funktion zweimal mit unterschiedlichen Namen auf.

3 Return-Werte

Erstellen Sie eine Funktion `addieren(a, b)`, welche die beiden übergebenen Werte addiert und `return` zur Rückgabe des Ergebnisses nutzt.

Geben Sie das Resultat anschliessend in der Konsole aus. Erstellen Sie zusätzlich eine weitere Funktion `laenge(text)`, welche die Länge des übergebenen Strings zurückgibt.

4 Anonyme Funktion und Arrow Function

Wandeln Sie folgende normale Funktion zuerst in eine anonyme Funktion und danach in eine Arrow Function um. Testen Sie alle drei Varianten im Browser:

```
function multiplizieren(x, y) {  
    return x * y;  
}
```

5 IIFE

Schreiben Sie eine selbstaufführende Funktion (IIFE), welche beim Laden der Seite in der Konsole den Text „Programm gestartet“ ausgibt.

6 EventListener hinzufügen

Ergänzen Sie folgendes HTML in der `index.html`, falls es noch nicht vorhanden ist:

```
<button id="btn">Klick mich</button>  
<p id="ausgabe"></p>
```

Fügen Sie im JavaScript einen EventListener hinzu, welcher bei jedem Klick auf den Button im Element `ausgabe` den Text „Der Button wurde gedrückt“ setzt. Nutzen Sie dafür eine normale Funktion.

7 EventListener mit Arrow Function

Ersetzen Sie nun die Funktion aus der vorherigen Aufgabe durch eine Arrow Function und prüfen Sie, ob alles wie zuvor funktioniert.

8 EventListener entfernen

Schreiben Sie eine benannte Funktion `melden()`, die den Text „Klick erkannt“ in der Konsole ausgibt. Fügen Sie diese Funktion als `click`-EventListener auf den Button hinzu und entfernen Sie sie nach dem ersten Klick sofort wieder.

(Hinweis: `removeEventListener` funktioniert nur mit benannten Funktionen.)

9 Mehrere Events

Fügen Sie eine zweite Interaktion hinzu:

- Ein `mouseover`-Event auf dem Button soll den Text im `ausgabe`-Element auf „Maus darüber“ setzen.
- Ein `mouseout`-Event soll den Text wieder löschen.

10 Unterschiedliche Funktionstypen anwenden

Erstellen Sie zu jedem Event aus Abschnitt 9 einen anderen Funktionstyp:

- `click`: normale benannte Funktion
- `mouseover`: anonyme Funktion
- `mouseout`: Arrow Function

11 Event-Toggle

Erstellen Sie einen zweiten Button mit dem Text **Toggle**. Wenn dieser Button gedrückt wird, soll der `EventListener` auf dem Eingabefeld entweder aktiviert oder deaktiviert werden – je nach aktuellem Zustand.

(Hinweis: Nutzen Sie `classList.contains` oder eine eigene Statusvariable.)

12 Mini-Projekt: Dynamische Eingabe

Erweitern Sie das HTML um folgendes Element:

```
<input id="eingabe" type="text" placeholder="Geben Sie etwas ein...">
```

Erstellen Sie nun eine Funktion `textAktualisieren()`, welche bei jeder Eingabe den aktuellen Inhalt des Feldes in das `ausgabe`-Element schreibt.

Binden Sie diese Funktion über einen `input`-`EventListener` ein.

13 Mini-Projekt: Interaktive Farbwahlen

Erweitern Sie das HTML-Dokument um die folgenden Elemente:

```
<div id="farbBox" style="width:150px; height:150px; border:1px solid #000;">
</div>
```

```

<label for="rot">Rot:</label>
<input id="rot" type="range" min="0" max="255" value="125">

<label for="gruen">Grün:</label>
<input id="gruen" type="range" min="0" max="255" value="125">

<label for="blau">Blau:</label>
<input id="blau" type="range" min="0" max="255" value="125">

<button id="reset">Zurücksetzen</button>

```

Erstellen Sie nun eine kleine Anwendung, welche es ermöglicht, die Farbe der Box dynamisch anzupassen. Gehen Sie dabei wie folgt vor:

1. Schreiben Sie eine Funktion `farbeAktualisieren()`, welche die aktuellen Werte der drei Regler ausliest und daraus eine gültige CSS-rgb-Farbe zusammensetzt.
2. Sorgen Sie dafür, dass die Funktion bei jeder Veränderung eines der Regler ausgeführt wird. Nutzen Sie hierzu den `input`-EventListener.
3. Setzen Sie die Hintergrundfarbe der `farbBox` dynamisch anhand der berechneten Werte.
4. Schreiben Sie ausserdem eine Funktion `zuruecksetzen()`, welche alle Regler wieder auf den Ausgangswert 125 setzt und die Box wieder auf eine neutrale Farbe stellt.
5. Binden Sie diese Funktion an den Button `reset`.
6. Nutzen Sie bei der Umsetzung der Funktionen unterschiedliche Funktionstypen (normale Funktion, anonyme Funktion, Arrow Function), wie in den vorherigen Aufgaben behandelt.
7. Erweitern Sie das Projekt optional so, dass die erzeugte Farbe zusätzlich in einem Absatz unterhalb der Box als Text ausgegeben wird, z. B. „Aktuelle Farbe: rgb(125, 200, 50)“.