

# JavaScript Grundlagen

---

## Lernziele

- Eine JavaScript-Datei mit `<script>` verbinden
  - In die Konsole loggen
  - Elemente (elements) mit `querySelector` auswählen
  - CSS-Klassen beim `click` mit `addEventListener` hinzufügen, entfernen und umschalten
- 

## Eine JavaScript-Datei verbinden

```
<head>
  ...
  <script src="./index.js" defer></script>
</head>
<body>
  ...
</body>
```

Das `script`-Tag hat zwei Attribute: `src="./index.js"` legt die URL zu unserer JavaScript-Datei fest. `defer` teilt dem Browser mit, das Laden des Skripts zu verzögern, bis alle HTML-Elemente geladen sind.

💡 Alternative: `script`-Tag am Ende des Body-Elements, sodass das `defer`-Attribut nicht notwendig ist. Weniger modern.

```
<head>
  ...
</head>
<body>
  ...
  <script src="./index.js"></script>
</body>
```

---

## Hallo Welt: `console.log()`

In JavaScript können wir Text in die Konsole des Webbrowsers ausgeben. Dies können wir zum Debuggen oder zum Protokollieren von Fehlern verwenden.

```
console.log("Hallo Welt!");
// loggt in die Konsole
console.clear();
// löscht die Konsole
```

```
console.error("Fehler!");  
// loggt als Fehler in die Konsole
```

## HTML-Elemente auswählen: `.querySelector()`

Bevor wir Interaktivität hinzufügen können, müssen wir die benötigten HTML-Elemente auswählen:

```
<body>  
  <main class="main" id="main" data-js="main">...</main>  
</body>
```

Es gibt mehrere Möglichkeiten, den obigen Main-Abschnitt in unserem JavaScript auszuwählen. Eine gute Praxis ist es, ein `data-*` Attribut, wie das **data-js** im folgenden Beispiel zu verwenden:

```
const mainElement = document.querySelector('[data-js="main"]');
```

Andere CSS-Selektoren funktionieren ebenfalls, aber die `data-*`-Attributselektoren sollten bevorzugt werden.

```
// Tag als Identifikator const mainElement =  
document.querySelector("main"); // Klasse als Identifikator -> . const  
mainElement = document.querySelector(".main"); // ID als Identifikator ->  
#  
const mainElement = document.querySelector("#main");
```

💡 Wir versuchen, unsere Verantwortlichkeiten zu trennen: Klassen sind für CSS und `data-*` Attribute sind für JavaScript.

## Interaktion hinzufügen: `.addEventListener()`

Wir können auf **Ereignisse (events)** wie **Klicks (click)** auf ein Element hören und Code ausführen, wenn das Ereignis ausgelöst wird. Die Methode `addEventListener` wird verwendet, um auf Ereignisse zu reagieren.

```
<button type="button" data-js="button">In die Konsole loggen</button>
```

```
const button = document.querySelector('[data-js="button"]');  
button.addEventListener("click", () => {});
```

Zuerst spezifizierst du die Art des Ereignisses, z.B. **click**, dann definierst du, welcher Code ausgeführt werden soll, wenn das Ereignis ausgelöst wird. Du schreibst diesen Code zwischen die `{}`-Klammern, z.B. ein `console.log```

```
const button = document.querySelector('[data-js="button"]');
button.addEventListener("click", () => {
  console.log("Yeah");
});
```

Es gibt verschiedene Ereignisse, auf die du hören kannst, zum Beispiel:

```
button.addEventListener("mouseover", () => {});
```

```
button.addEventListener("keydown", () => {});
```

💡 Hier findest du eine [Liste der Ereignistypen](#). 💡 Du musst die Syntax jetzt noch nicht verstehen, das behandeln wir in einer späteren Sitzung.

---

## Klassen hinzufügen/entfernen & umschalten: `.classList`.

Du kannst Klassen hinzufügen, entfernen und umschalten, z.B. um das Styling eines Elements zu ändern.

```
<main data-js="main">
  <button type="button" data-js="button">Eine Klasse hinzufügen</button>
</main>
```

Füge die Klasse **page--primary** zu dem oben stehenden Main-Abschnitt hinzu, indem du die Methode (method) `selectedElement.classList.add` verwendest:

```
const main = document.querySelector('[data-js="main"]');
const button = document.querySelector('[data-js="button"]');

button.addEventListener("click", () => {
  main.classList.add("page--primary");
});
```

Ein Klick auf den Button fügt dem Main-Element die Klasse **page--primary** hinzu:

```
<main data-js="main" class="page--primary">
  <button type="button" data-js="button">Eine Klasse hinzufügen</button>
```

```
</main>
```

Du kannst auch eine Klasse auf dieselbe Weise entfernen oder umschalten:

```
main.classList.remove("page--primary");
```

```
main.classList.toggle("page--primary");
```

---

## Ressourcen

Eine JavaScript-Datei verbinden

[Das Script-Element](#)

Hallo Welt

[Konsole](#)

HTML-Elemente auswählen

[Document](#)

[Verwendung von data-Attributen](#)

[document.querySelector](#)

[data-\\* Attribut](#)

Interaktion hinzufügen

[.addEventListener\(\)](#)

[Ereignisreferenz Klassen](#)

hinzufügen/entfernen & umschalten

[classList](#)