

# LU01b - JavaScript in HTML einbinden

## Einleitung

Es gibt verschiedene Möglichkeiten, JavaScript in eine HTML-Seite einzubinden. Die wichtigsten sind:

1. Inline im HTML-Tag
2. Intern im `<script>`-Tag
3. Extern in einer separaten .js-Datei
4. Mit `defer` und `async`

## 1. Inline im HTML-Tag

Der JavaScript-Code wird direkt als Attribut in einem Element reingeschrieben. ☐ Das ist nicht die eleganteste Lösung, aber für schnelle Tests praktisch.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Inline JavaScript</title>
  </head>
  <body>
    <!-- JavaScript-Code mit einer-alert-box wird ausgegeben, wenn auf Button
geklickt -->
    <!-- Bitte Browser wie Chrom, Firefox, .... verwenden zum Testen. -->
    <button onclick="alert('Hallo, Welt!')">Button für Alert Box</button>
  </body>
</html>
```

**Hiweis:** Der obige sollte in einem regulären Browser getestet werden, *VS Code Live Preview* erlaubt keine PopUp Fenster wie alertbox, etc.

## 2. Intern im `<script>`-Tag

JavaScript wird direkt im HTML-Dokument, meist im `<head>` oder besser am Ende von `<body>` platziert, damit das DOM schon geladen ist.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Internes Script</title>
  </head>
  <body>
    <h1>Beispiel mit Script-Tag</h1>
```

```
<h3>Beim Laden der Seite wird die Hintergrundfarbe auf "lightblue"
geändert.</h3>
<!-- JS-Code-Bereich wird irgendwo definiert, hier direkt dort wo es
gebraucht wird, und gleich ausgefüllt.
In diesem Falle wird einfach Hintergrundfarbe mittels JS verändert
--->
<script>
  document.body.style.backgroundColor = "lightblue";
</script>
</body>
</html>
```

### 3. Extern in einer separaten .js-Datei

Dieser Ansatz ist der saubere Standardweg: Gemäss der *Maxime Teile und herrsche* wird HTML und JS in separaten Dateien gehalten. So bleibt der HTML-Code übersichtlich und JavaScript kann wiederverwendet werden.

#### HTML-Datei

```
<!DOCTYPE html>
<html>
  <head>
    <title>Externes Script</title>
    <script src="script.js" defer></script>
  </head>
  <body>
    <h1>Externes JavaScript</h1>
  </body>
</html>
```

#### Script.js

```
document.addEventListener("DOMContentLoaded", () => {
  console.log("Externes Script wurde geladen!");
  alert("Hallo aus script.js!");
})
```

### 4. Mit defer und async

- **defer:** Script wird nach dem Laden des HTML ausgeführt (Reihenfolge bleibt erhalten).
- **async:** Script wird parallel geladen und sobald fertig, sofort ausgeführt (Reihenfolge nicht garantiert).

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>Async & Defer</title>
    <script src="defer-script.js" defer></script>
    <script src="async-script.js" async></script>
  </head>
  <body>
    <h1>Async vs. Defer</h1>
  </body>
</html>
```

## Zusatzmaterial

- [W3School-Where\\_to\\_bind](#)
- [SelfHTML-JS einbinden](#)



Volkan Demir

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/de/modul/m288/learningunits/lu01/02?rev=1756459386>

Last update: **2025/08/29 11:23**

