

JavaScript



HTML



Markup: Inhalte strukturieren

CSS



Styling: Elemente gestalten

JS



Verhalten: Elemente steuern

- erste Version „LiveScript“ 1995 von Netscape entwickelt
 - Wegen Kooperation zwischen Netscape (LiveScript) und Sun Microsystems (Java) dann in JavaScript umbenannt
- JavaScript ≠ Java
- Skriptsprache
 - Programmiersprache, die über einen Interpreter (= Übersetzer zwischen Anweisungen und Maschinencode) ausgeführt wird
- dynamisch typisierte, objektorientierte und klassenlose Skriptsprache

- ECMA Script (ES) ist eine Spezifikation (\approx standardisierendes Regelwerk) für Skriptsprachen
 - Herausgeber ist Normierungsorganisation Ecma International
- JavaScript ist die am weitesten verbreitete Implementierung von ECMA Script
- aktuelle Version ist ECMA Script 2019 (ES10)
 - letzte „große“ Version war ECMA Script 2015 (ES6)

Wofür wird JavaScript genutzt?

- Ursprünglich für dynamisches HTML entwickelt, heute extrem breites Anwendungsspektrum
 - Erweiterung der Möglichkeiten von HTML und CSS
 - Möglichkeit der Benutzerinteraktion
 - dynamisches Verändern, Nachladen und Generieren von Inhalten
 - Beispiele: Bewegung von Elementen, Parallax-Effekte, Formularvalidierung, Dialogfenster, Werbebanner, Cookies, Spiele usw.
- häufig client-seitige Anwendung im Browser
 - sowohl Nutzung von nativem JS, als auch in zahlreichen Frameworks (z.B. jQuery, React.js, Angular.js, Vue.js)
- Mittlerweile jedoch auch server-seitige Anwendung durch Frameworks (z.B. Node.js)

Internal JavaScript

- das Skript wird im HTML-Dokument in einem `<script>`-Element eingefügt
- eignet sich vor allem für kleinere Skripte (nur wenige Zeilen, z.B. Tracking-Skripte), jedoch nicht für komplexere Anwendungen

External JavaScript

- das Skript liegt in einer externen JavaScript-Datei (*.js) und wird im HTML-Dokument mittels eines `<script>`-Elements verlinkt
- eignet sich für alle Arten von Skripten von wenigen Zeilen bis hin zu komplexen Anwendungen

Allgemeiner Hinweise:

- Skripte, die mit den Elementen der Website interagieren sollen, müssen unbedingt erst direkt vor dem schließenden `</body>`-Tag eingebunden werden

```
index.html X
index.html > ...
1  <!DOCTYPE html>
2  <html lang="de">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport"
6          content="width=device-width, initial-scale=1.0">
7      <title>JavaScript</title>
8  </head>
9  <body>
10
11      <h1>Lorem ipsum</h1>
12      <p>Dolor sit, amet consectetur adipisicing elit.</p>
13
14      <script>
15          let sprache = "JavaScript";
16          if (sprache === "JavaScript") {
17              alert("Voll super!");
18          } else {
19              alter("Och nö...");
20          }
21      </script>
22  </body>
23  </html>
24
```

```
<> index.html x
index.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="de">
3  <head>
4  |   <meta charset="UTF-8">
5  |   <meta name="viewport"
6  |   |   content="width=device-width, initial-scale=1.0">
7  |   <title>JavaScript</title>
8  </head>
9  <body>
10 |
11 |   <h1>Lorem ipsum</h1>
12 |   <p>Dolor sit, amet consectetur adipisicing elit.</p>
13 |
14 |   <script src="main.js"></script>
15 </body>
16 </html>
17

JS main.js x
JS main.js > ...
1  let sprache = "JavaScript";
2  if (sprache === "JavaScript") {
3  |   alert("Voll super!");
4  | } else {
5  |   alter("Och nö...");
6  | }
```


Code Beispiel

