LU01a - JavaScript Einführung

Einleitung

JavaScript (kurz: JS) ist heute eine der wichtigsten Programmiersprachen im Bereich der Webentwicklung. Ursprünglich in den 1990er-Jahren entwickelt, um Webseiten dynamischer zu gestalten, hat sich die Sprache längst über ihre anfängliche Rolle hinaus entwickelt. Sie ist mittlerweile universell einsetzbar – **sowohl im Browser als auch auf Servern und sogar in Desktop- oder Mobile-Apps**. Der große Vorteil: Nahezu jeder Browser unterstützt JavaScript ohne zusätzliche Installation, was es zu einer global verfügbaren Technologie macht.

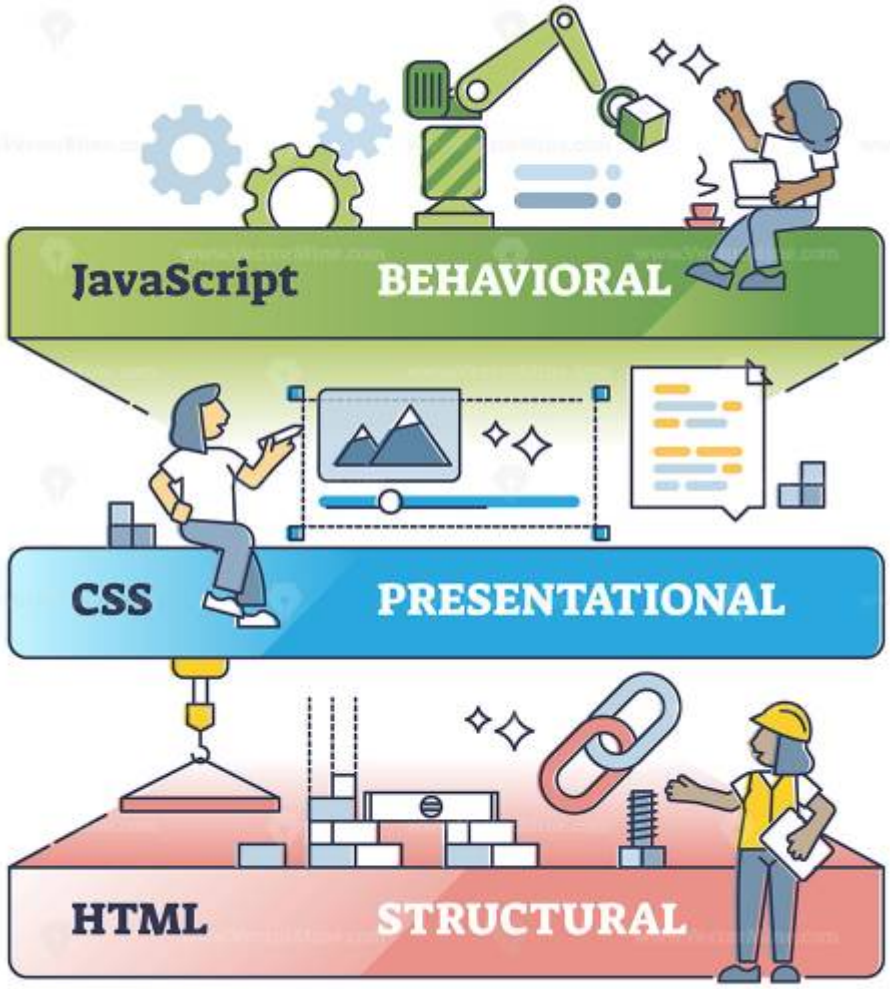
Was ist JavaScript?

JavaScript ist eine interpretierte, dynamisch typisierte Skriptsprache, die eng mit HTML und CSS zusammenarbeitet. Sie ermöglicht das Erstellen von interaktiven, dynamischen und reaktiven Webinhalten. Technisch gesehen handelt es sich um eine Multi-Paradigmen-Sprache – man kann objektorientiert, funktional oder imperativ programmieren. Die Syntax ist C-ähnlich, die Sprache ist jedoch flexibel und erlaubt schnelles Prototyping.

Zusammenspiel mit HTML und CSS

Man kann sich HTML, CSS und JavaScript als drei Säulen moderner Webentwicklung vorstellen:

- **HTML**: Struktur der Inhalte (Texte, Bilder, Überschriften, Formulare usw.)
- **CSS**: Gestaltung und Layout (Farben, Abstände, Animationen, Schriftarten)
- **JavaScript**: Interaktivität und Logik (z. B. Formulareingaben prüfen, Inhalte dynamisch nachladen, Animationen steuern)



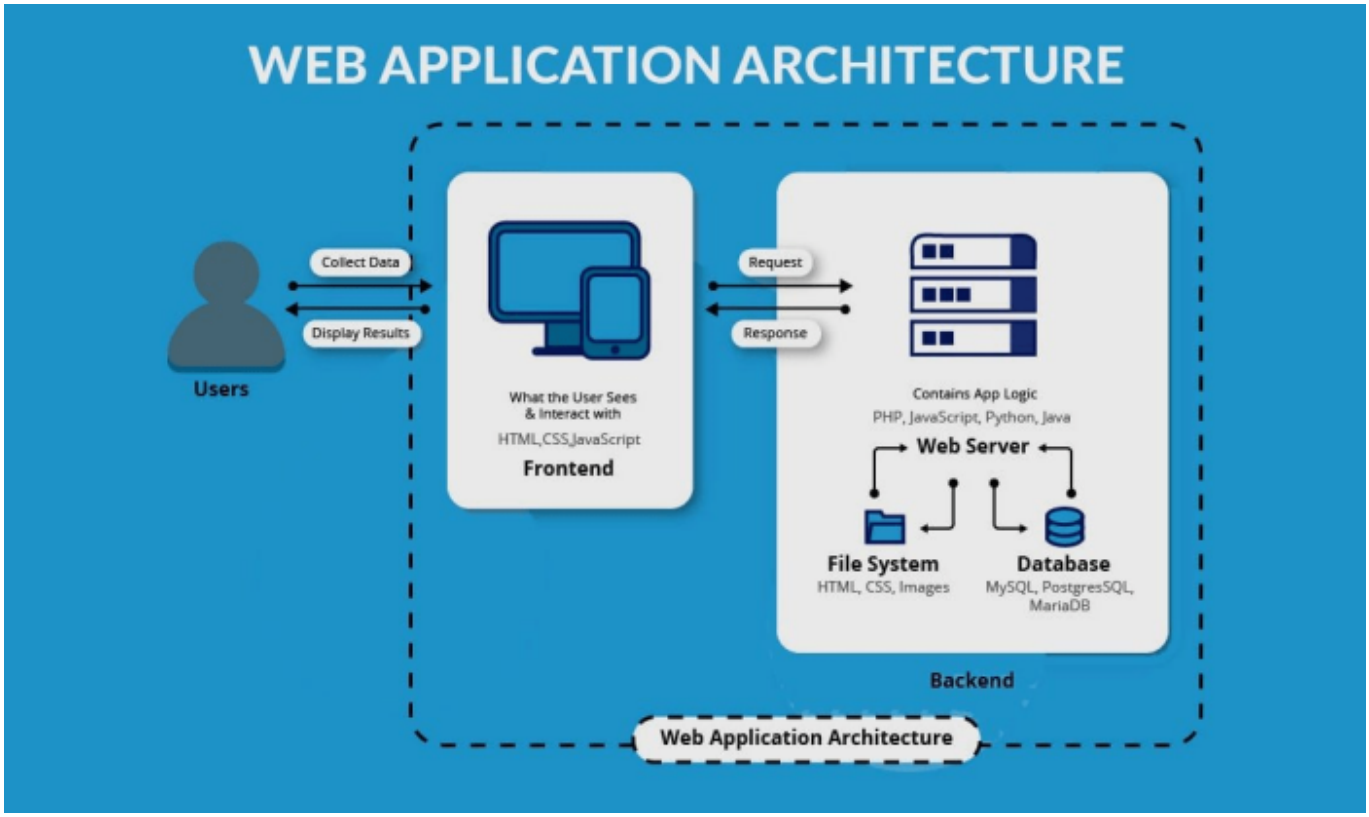
Beispiel:

- HTML stellt ein Eingabefeld dar.
- CSS sorgt dafür, dass es optisch ansprechend aussieht.
- JavaScript überprüft, ob der Benutzer eine gültige E-Mail-Adresse eingegeben hat, und gibt ggf. eine Fehlermeldung aus – ohne die ganze Seite neu zu laden.

JavaScript in einer Client-Server-Architektur

In einer klassischen*Client-Server-Architektur* wird JavaScript vor allem auf der Client-Seite (also im Browser) ausgeführt. Hier interagiert es direkt mit dem DOM (Document Object Model), um Inhalte zu ändern oder Benutzeraktionen auszuwerten.

Mit der Entwicklung von Node.js kann JavaScript aber auch auf der Server-Seite laufen. Damit kann dieselbe Sprache sowohl die Präsentationsschicht (Frontend) als auch die Logik- und Datenspeicherebene (Backend) bedienen.



In modernen Architekturen wie Single Page Applications (SPA) oder Serverless-Ansätzen wird JavaScript oft an mehreren Punkten gleichzeitig eingesetzt.

Anwendungen – Clientseitig

- Dynamische Anpassung von Webseiteninhalten
- Validierung von Formulareingaben
- Interaktive Benutzeroberflächen (z. B. Drag & Drop, Sliders, Menüs)
- Echtzeitkommunikation mit Servern über AJAX oder WebSockets
- Animationen und visuelle Effekte
- Rendering von Daten (z. B. Tabellen, Diagramme) ohne Neuladen der Seite

Anwendungen – Serverseitig

- Webserver-Implementierung mit Node.js
- API-Entwicklung (REST, GraphQL)
- Verarbeitung von Datenbankabfragen
- Authentifizierung und Zugriffskontrolle
- Echtzeit-Anwendungen wie Chats oder Multiplayer-Spiele
- Automatisierungs- und Skriptaufgaben auf Servern
- Generierung dynamischer HTML-Seiten (Server Side Rendering, SSR)

Aktuelle JS-Version

Was meinen viele Entwickler damit, wenn sie „JS-Version“ sagen? Gemäss Wikipedia ist

- **Version 17 (ES17), auch bekannt als ECMAScript 2026 (ES2026) die aktuellste Version**.
- Am häufigsten dürfte aktuell die **ES14er - 2023** Version verwendet werden.

ES14 entspricht übrigens ECMAScript 2023 (es gibt eine leichte Nummer-Verschiebung: ES14 = Jahr 2023, ES15 = 2024, ES17 = 2026)

Zusatzmaterial

- W3School/JS-Introduction
- Wikipedia/JS-Version

Volkan Demir / Elena Mastrandrea

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m288/learningunits/lu01/01

Last update: **2026/08/11 15:21**