

JavaScript (JS)

JavaScript je multiparadigmatický jazyk (podporuje imperativní, objektově orientované i funkcionální programování). Je standardizován pod názvem **ECMAScript**. Na rozdíl od **C++** nebo **C**, které se kompilují dopředu, JavaScript je obvykle kompilován metodou **JIT (Just-In-Time)** přímo v běhovém prostředí.

1. Historie a Evoluce

- **1995:** Brendan Eich vytvořil jazyk za pouhých 10 dní pro prohlížeč Netscape. Původně se jmenoval Mocha, pak LiveScript a nakonec JavaScript (z marketingových důvodů kvůli popularitě Javy).
- **2009:** Vznik **Node.js** umožnil spouštět JS i na serveru, což změnilo pravidla hry.
- **2015 (ES6):** Zásadní aktualizace, která přinesla moderní syntaxi (třídy, šipkové funkce, sliby - promises).

2. Jak JavaScript funguje?

JavaScript běží v tzv. **JavaScript Engine**, což je program, který kód interpretuje a optimalizuje. Nejslavnějším příkladem je **V8** (v prohlížeči Chrome a Node.js).

Event Loop (Smyčka událostí)

JavaScript je **jednovláknový**, což znamená, že může dělat jen jednu věc naráz. Přesto zvládá tisíce operací současně díky asynchronnímu modelu a **Event Loop**.

1. **Call Stack:** Zde se ukládají funkce, které se právě vykonávají.
2. **Web APIs / Node APIs:** Dlouhotrvající operace (např. načítání dat ze sítě) se předají prohlížeči/systému.
3. **Callback Queue:** Jakmile je operace hotová, výsledek čeká ve frontě.
4. **Event Loop:** Pokud je Stack prázdný, přesune úkol z fronty do Stacku.

3. Ekosystém JavaScriptu

JavaScript má největší repozitář knihoven na světě - **npm** (Node Package Manager).

Frontend

JS ovládá prohlížeč prostřednictvím **DOM** (Document Object Model), což je stromová struktura HTML stránky.

- **Frameworky:** React, Vue.js, Angular (viz článek [frontend](#)).

Backend (Node.js)

Umožňuje psát serverovou logiku v JS. Je extrémně rychlý pro I/O operace, jako jsou chatovací aplikace nebo streamování dat.

Mobilní a Desktopové aplikace

- **React Native:** Tvorba mobilních aplikací pro iOS a Android jedním kódem.
- **Electron:** Tvorba desktopových aplikací (např. VS Code, Discord, Slack) pomocí webových technologií.

4. TypeScript: "Nástupce" JavaScriptu

Protože je JS dynamicky typovaný (proměnná může změnit typ z čísla na text), u velkých projektů často dochází k chybám. **TypeScript** (od Microsoftu) přidává do JS statické typování. Kód se pak pomocí transpilátoru (např. [Babel](#) nebo TSC) přeloží zpět do čistého JavaScriptu.

5. Příklad kódu

Asynchronní načítání dat (Moderní ES6+):

```
const fetchUser = async (id) => {
  try {
    const response = await fetch(`https://api.example.com/users/${id}`);
    const data = await response.json();
    console.log(`Uživatel: ${data.name}`);
  } catch (error) {
    console.error("Chyba při stahování dat:", error);
  }
};

fetchUser(1);
```

Srovnání: JavaScript vs. Ostatní

Vlastnost	JavaScript	Python	C++
Rychlost	Vysoká (JIT)	Střední (Interpretr)	Maximální (Nativní)
Typování	Dynamické	Dynamické	Statické
Vlákna	Jedno (Event Loop)	Více (ale s GIL)	Plná podpora vláken
Hlavní užití	Web, Fullstack	AI, Data Science	Systemy, Hry

Víte, že? Navzdory podobnému názvu nemá **Java** a **JavaScript** skoro nic společného. Je to jako rozdíl mezi slovy „Auto“ a „Automat“.

Související: [frontend](#), [I/O](#), [Linux](#), [Bash](#)

From:
<https://serviceit.cz/> - **IT ENCYKLOPEDIE**

Permanent link:
<https://serviceit.cz/doku.php?id=js>

Last update: **2025/12/31 18:11**

