



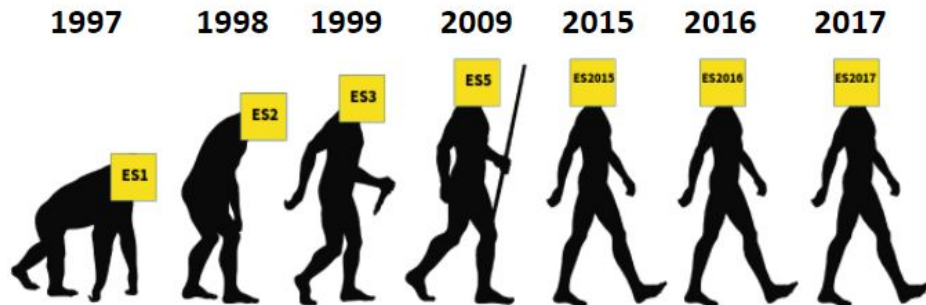
Programarea aplicatiilor in NodeJS

Curs 2



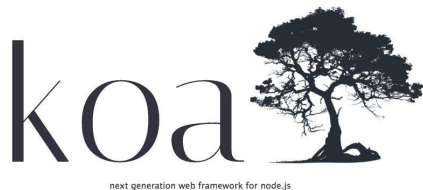
◀ Istoria JavaScript

- Creat de Brendan Eich în 1995 pentru Netscape Navigator pentru a adauga interactivitate în aplicațiile web
- Inițial a fost făcut în 10 zile cu intenția de a fi folosit complementar cu Java, dar destul de rapid a evoluat într-un limbaj de programare foarte puternic și de sine statator
- În 1997 s-a creat ECMAScript, un standard pe care JS îl respecta, pentru a asigura interoperabilitatea paginilor web pe cât mai multe browsere diferite.



◀ Rolul JavaScript in Web Dev

- Singurul limbaj de programare nativ pentru browsere web, este folosit pentru a facilita conținut dinamic și pentru a crea aplicații web interactive
- Cu Node.js (introdus in 2009), JavaScript s-a extins si pe partea de backend/servers, astfel JS devenind un limbaj full-stack
- Are un ecosistem foarte mare de librarii si frameworks precum React, Angular si Vue pentru frontend, sau Express sau Koa pentru backend.



Express



◀ Javascript Engines

- Un engine este un program care poate executa cod JavaScript.
- V8 (Chrome si Node.js), SpiderMonkey (Firefox), JavaScriptCore (Safari)



◀ Istoria ECMAScript

- ES1 (1997)
 - Var keyword
 - Functions
 - Objects
- ES3 (1999)
 - Regular expressions
 - try/catch
 - error handling
 - Object.create, Object.keys, etc.
- ES5 (2009)
 - “strict mode”
 - JSON support
 - Array.forEach, Array.map, Array.reduce, Array.filter, etc
 - multe improvements pe partea de securitate

◀ Istoria ECMAScript

- ES6 (2015) - printre cele mai importante updates
 - Classes
 - Modules
 - Arrow functions
 - Promises
 - Template literals
 - let and const
 - destructuring

◀ Istoria ECMAScript

- Incepand cu 2016, ECMAScript a inceput sa scoata updates anual
- ES2017
 - Async/await
- ES2018
 - rest/spread operator (... operator)
- ES2020
 - Optional chaining, nullish coalescing (?, ??)

◀ Cum este introdusă o propunere în Standard?

- Stage 0: Strawperson/Strawman
 - Propunerea urmează să fie prezentată comisiei TC39 care decide dacă trece în stage 1 sau nu
- Stage 1: Proposal
 - Propunerea reprezintă un interes pentru comisie, se începe definirea problemei, impactul asupra standardului, căutarea mai multor soluții pentru problema
- Stage 2: Draft
 - Se creează draftul propunerii, cu soluția, sintaxa, și tot ce mai este nevoie
- Stage 3: Candidate
 - Soluția este completă, se așteaptă feedback extern
- Stage 4: Finished
 - Poate fi adăugat în ECMAScript
- <https://www.proposals.es/>

◀ JavaScript Runtime Environment

- Un runtime environment ne ofera toata infrastructura necesara pentru a executa cod javascript.
 - JavaScript engine
 - Parseaza, compileaza si executa codul
 - Transforma codul in machine language
 - E.g. V8 Engine (Chrome, Node.js), e rapid si eficient
 - APIs
 - Web APIs (DOM, AJAX, etc.)
 - Node.js APIs (file system, HTTP requests, streams, etc)
 - Event loop

◀ JavaScript in Browser vs Server

- Cea mai mare diferenta: runtime environment si APIs disponibile
- Browser:
 - DOM, Geolocation, Clipboard, Fetch, Canvas, etc
 - History, location
 - Obiectul global se numește “window”
- Node.js:
 - HTTP, OS, Path, Crypto, Process, Stream, etc.
 - Obiectul global se numește “global”

◀ JavaScript in Browser vs Server

Sistemul de module diferă destul de mult dintre browser si server.

În browser se folosește modelul “ES Modules”, adica import/export, si e supported de toate modern web browsers.

In Node.js se foloseste modelul “CommonJS”, adica require/module.exports, dar in ultima perioada lumea incearca sa treaca la ES Modules și pe server.

◀ JavaScript basics

- JavaScript este “weakly typed”, adică programul își da seama singur de tipul de date pe care o variabila o are.
- Tipuri de date: number, string, bigint, boolean, null, undefined, object, symbol
 - Unde este array?
- Variabile: var, let si const
- If, else if, else, switch
- For, while, for x in obj, for y of array
- Functii:

```
function greet() {  
  console.log("Hello World!");  
}
```

```
const greet = function() {  
  console.log("Hello World!");  
};
```

```
const add = (a, b) => {  
  return a + b;  
};
```