

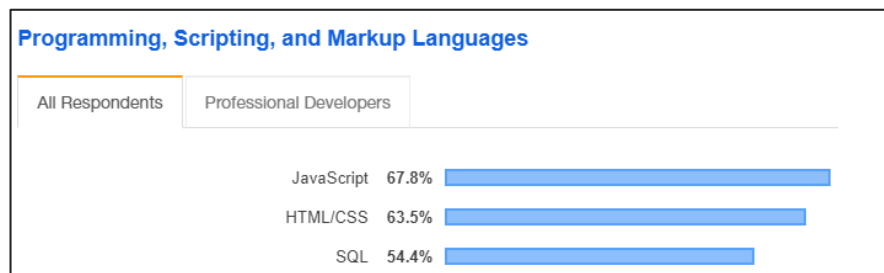
OSNOVE JAVASCRIPT

Patrik Rek,
Viktor Taneski



KAJ JE JAVASCRIPT?

- Objektno usmerjen programski jezik
- Dodajanje interaktivnih možnosti spletnim stranem
- Logični nivo spletnih strani
- Lahko se izvaja na strežniku (s pomočjo izvajalnega okolja kot je Node.js) ALI v **brskalniku**
- Že 7 let zaporedoma [najpogosteje uporabljen programski jezik](#)
- Dinamično tipiziran programski jezik (za strogo tipizirano „nadgradnjo“ poznamo TypeScript)



VKLJUČEVANJE JAVASCRIPT V HTML

- S pomočjo značke **<script>**
 - Koda znotraj značke (deluje samo v trenutnem HTML dokumentu)
 - Koda v ločeni datoteki, ki jo naslovimo s pomočjo atributa **src** (npr. *<script src='js.js'></script>*)

SPREMENLJIVKE IN TIPI

- ~~var~~: zastarel, odsvetovan zapis. Obseg: funkcija.
- **let**: obseg: blok.
- **const**: obseg: blok. Ni možno spreminjanje.
- Tipi: dinamično dodeljeni, preverjanje tipov možno s pomočjo **typeof**

POLJA

- `= new Array()` ALI `[]`
- Dodajanje elementov: **`polje.push(element)`**
- Odstranjevanje elementov: **`polje.splice(pozicija, kolikoElementov)`**
- Ali element obstaja v polju? **`polje.includes(element)`**
- Na katerem mestu v polju se nahaja element? **`polje.indexOf(element)`**
- Dolžina polja: **`polje.length`**

FUNKCIJE

- `function imeFunkcije(parametri) {}`

ALI

- `const imeFunkcije = (parametri) => {}`

- Funkcija lahko vrača vrednost (**return**) ali pa je tipa **void** in je ne vrača.

OBJEKTI IN RAZREDI

```
let objekt = new Object();  
objekt.lastnost = vrednost;
```

```
class Razred {  
    constructor(parameter) {  
        this.parameter = parameter;  
    }  
}  
  
let objekt = new Razred(parameter);
```

LOCALSTORAGE IN SESSIONSTORAGE

- **localStorage** → shramba podatkov, ki je dostopna v vseh sejah
- **sessionStorage** → shramba podatkov, ki je dostopna zgolj v trenutni seji (do osveževanja)
- `localStorage.setItem('ključ', 'vrednost')` → nastavi vrednost za določen ključ
- `localStorage.getItem('ključ')` → pridobi vrednost za določen ključ
- `localStorage.removeItem('ključ')` → odstrani določeno vrednost iz shrambe
- `localStorage.clear()` → pobriše celotno shrambo

PREVERJANJE ENAKOSTI

- == preveri zgolj vrednost
- === preveri vrednost in tip

ZANKE

- **for** (let i = 0; i < 10; i++) {} → zanka, ki povečuje vrednost i od 0 do 10
- polje.**forEach**(element => console.log(element)) → zanka, ki gre skozi vse elemente polja
- **for** (const element **of** polje) {} → zanka, ki gre skozi vse elemente polja
- **for** (const lastnost **in** objekt) {} → zanka, ki gre skozi vse lastnosti objekta
- Pozor: ne zamenjajte **for..of** in **for..in**! Pogostejše se uporablja **for..of**.

DOM (1/2)

- HTML **D**ocument **O**bject **M**odel
- JavaScript lahko s pomočjo tega modela spremeni vse HTML elemente, attribute, CSS stile ...
- **document.getElementById("id")** → pridobi HTML element z določenim ID-jem
- **document.getElementsByClassName("razred")** → pridobi vse HTML elemente z določenim razredom
- **document.forms** → pridobi vse obrazce v trenutnem HTML dokumentu
- **element.innerHTML = "Poljubna vrednost";** → nastavi vsebino elementa
- **element.atribut = vrednost** → nastavi vrednost atributa elementa
- **element.style.lastnost = nov stil** → spremeni stil

DOM (2/2)

- **document.createElement("DIV")** → ustvari nov element
- **element.appendChild(novElement)** → doda nov element na konec obstoječega
- Dogodki:
 - **onclick** v katerem koli HTML elementu → sproži se ob kliku
 - **onchange** v elementu **input** → sproži se vsakič ob spremembi vrednosti
 - **onmouseover** in **onmouseout** v katerem koli elementu → sprožita se, ko je miška na elementu oz. ko ga zapusti
 - **onsubmit** v elementu **form** → sproži se ob sprejemanju obrazca

DODATNO

- [W3Schools](#)
- [Mozilla MDN](#)
- [TutorialsPoint](#)
- Microsoft Teams skupina