

Uvod u JavaScript



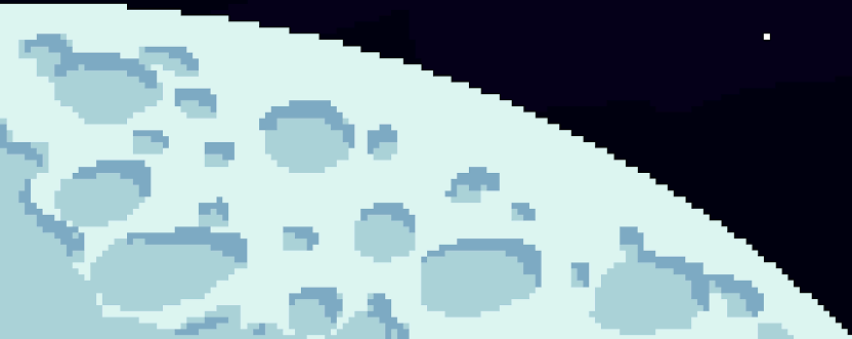
O JavaScriptu

- Nastao 1995: od strane NetScapea
- Jedan od najpopularnijih programskih jezika
- Može se koristiti za:
 - frontend (Javascript + React, Vue, Angular)
 - backend (Node.js)
 - desktop aplikacije (Electron)
 - mobilne aplikacije (React Native)



Osnovna sintaksa

- Skriptni jezik, interpreter evaluira liniju po liniju
- Nema potrebe za main funkcijom
- Osnovna sintaksa bliska C-u ili Javi (zagrada, ; itd.)
- Znak `;` nije obavezan, ali se preporučuje



JavaScript ≠ Java

Primer osnovnog Hello World programa:

Java

```
class Program {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

JavaScript

```
console.log("Hello World");
```

Variable

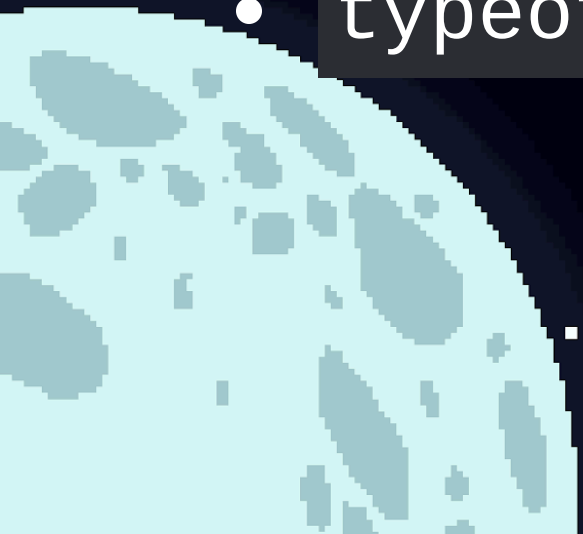
- U JS-u varijable se definišu pomoću jednog od 3 keyworda:
 - `let`
 - `const`
 - `var`
- `const` koristimo za konstante
- `var` se u poslednje vreme ne koristi, nasledio ga je `let`



+ Tipovi varijabla



- JS nije statično tipiziran
- Runtime ne brani promenu tipa tokom izvršavanja (ali je to loša ideja)
- `typeof` operator vraća string sa tipom varijable





..Vrednostni tipovi:

- Number
- BigInt
- String
- Boolean
- Object
- Function

• Tipovi koji ne sadrže vrednost:

- undefined
- null
- NaN



Ponekad tipovi maaalo nemaju smisla:

- `typeof typeof 123 == "string"`
- `typeof null == "object"`
- `typeof NaN == "number"`
- `typeof class Foo {} == "function"`
- `typeof [1,2,3] == "object"`



Operatori za poredenje

- Postoje 2 načina za poredenje varijabli:
 - `==` i `!=` → porede samo vrednosti, ali ne i tipove
 - `===` i `!==` → porede i vrednosti i tipove
(preporučljivo)



Aritmetički operatori

- `+`, `-`, `*`, `/`, `%`, `**`
- Mogu se i assignovati: `+=`, `-=`, `*=`, `/=`, `%=`, `**=`
- Takođe, postoji i increment i decrement: `i++`, `i--`
- `+` i `+=` se mogu koristiti i za spajanje stringova
- Ostali operatori automatski castuju operande u `Number`
 - Koji je output `2 + '2' - 2`?



Logički operatori

- `&&` (AND) → vraća `true` ako su obe vrednosti `true`
- `||` (OR) → vraća `true` ako je barem jedna vrednost `true`
- `!` (NOT) → negira vrednost, tj vraća kontra vrednost
- JS radi lazy evaluaciju
- `??` (Nullish) → vraća drugu vrednost ako je prva `null` ili `undefined`

Funkcije

Funkcije se mogu napisati na više načina:

Deklaracija
funkcije

```
function greet(name) {  
  return `Hello, ${name}!`;  
}
```

Izraz
funkcije

```
const greet = function(name) {  
  return `Hello, ${name}!`;  
};
```

Arrow
funkcija

```
const greet = (name) => {  
  return `Hello, ${name}!`;  
};
```



Kondisional

- `if...else` statement
- `switch` statement

if...else

```
if(uslov1) {  
    // prva akcija  
}  
else if (uslov2) {  
    // druga akcija  
}  
else {  
    // treca akcija  
}
```

Switch

- Ukoliko imamo više grananja za različite vrednosti varijable, bolje je koristiti `switch`

```
switch(varijabla) {  
    case 1:  
        // prva akcija  
        break;  
  
    case 2:  
        // druga akcija  
        break;  
  
    default:  
        // treca akcija  
}
```





Petlje

- `while i do..while`
- `for, for..in i for..of`

while i do..while

while

```
while(uslov) {  
    // akcija  
}
```

do..while

```
do {  
    // akcija  
} while(uslov)
```

- do..while će se garantovano barem jednom izvršiti



For petlje

for

```
for(init; uslov; korak) {  
    // akcija  
}
```

for..in

```
for(kljuc in iterable) {  
    // akcija  
}
```

for..of

```
for(vrednost of iterable) {  
    // akcija  
}
```

- **for..in** i **for..of** služe za prolazak kroz nizove i objekte
- **for..in** vraća index ili ključ elementa
- **for..of** vraća kopiju vrednosti elementa
 - promene vrednosti elementa **NE** utiču na objekat



Pitanja?

Hvala na pažnji!