



Fakultet informatike u Puli

Programsko inženjerstvo - Vježbe

2. Struktura programa

(prilagođeno prema knjizi *Eloquent Javascript*)

Sav Javascript se može isprobati u konzoli internet preglednika ili na adresi <https://jsconsole.com/>.

U prethodnom smo poglavlju upoznali osnovne Javascript tipove vrijednosti, te unarne i binarne operatore nad njima. Potrebno je kombinirati vrijednosti i operatore u složenije izraze koje stvaraju nove vrijednosti.

Vrijednosti se pohranjuju u varijablama koje predstavljaju **stanje** programa. Varijable je potrebno deklarirati (ključna riječ `let`), a zatim im i postaviti vrijednost pomoću operatora `=`.

```
1 let varijabla; // deklaracija
2 varijabla = 10; // postavljanje vrijednosti
3
4 let nova_varijabla = 11; // ... u jednom koraku
5
6 console.log(varijabla, nova_varijable)
```

Postoji i posebna vrsta varijable `const` kojoj nije dozvoljeno promijeniti jednom inicijaliziranu vrijednost.

```
1 const konstanta = 3.14;
2 konstanta = konstanta + 1; //TypeError!
```

Varijable u Javascriptu mogu početi bilo kojim slovom te znakovima `$` i `_`.

Okruženje. Skup svih pridruženih vrijednosti varijabli naziva se okruženje. Pri pokretanju programa već postoji set unaprijed definiranih varijabli, primjerice varijabla `console`. Takve predefinirane varijable omogućuju komunikaciju s ostalim djelovima sustava (npr. ispis, čitanje korisničkih podataka, ...).

Funkcije. Tip "funkcija" u Javascriptu predstavlja dio programa koji je potrebno izvršiti vezan uz neki naziv (*binding*). Za razliku od ostalih programskih jezika, funkcije su vezane uz naziv kao i varijable, s tom razlikom da ih je moguće izvršiti (pozvati). Pozivanje se izvršava tako da se nakon naziva varijable koja sadrži funkciju napišu zagrade s listom argumenata (`()` i `()`).

```
1 // prompt je funkcija za unos podataka
2 prompt(); // poziv bez argumenata
3 prompt("Unesi neki broj"); // poziv s argumentima
```

Kako su funkcije vezane uz naziv kao i varijable, moguće ih je čak promijeniti da označavaju nešto drugo.

Primjer:

```
1 var a = prompt("Unesi neki broj:");
2 console.log(a);
3 prompt = "više nisam funkcija";
4 var b = prompt("Unesi neki broj:") // TypeError!
```

Uvjetno izvođenje. Uvjetno izvođenje određeno je ključnom riječi `if` i `else` te se ponaša slično kao u programskom jeziku C.

```
1 let theNumber = Number(prompt("Odaberi jedan broj"));
2 if (!Number.isNaN(theNumber)) {
3     console.log("Odabrao si korijen od broja " +
4         theNumber * theNumber);
5 }
```

*Pitanje: čemu služi funkcija **Number** iz primjera?*

Petlje. Postoje tri vrste petlji u Javascriptu:

1. `while` petlja
2. `do while` petlja
3. `for` petlja

Primjeri:

```
1 let result = 1;
2 let counter = 0;
3 while (counter < 10) {
4     result = result * 2;
5     counter = counter + 1;
6 }
7 console.log(result);
8 // → 1024
```

```
1 let yourName;
2 do {
3     yourName = prompt("Who are you?");
4 } while (!yourName);
5 console.log(yourName);
```

```
1 for (let number = 0; number <= 12; number = number + 2) {
2     console.log(number);
3 }
```

Slično kao i u C-u, ukoliko je potrebno prekinuti izvođenje petlje, možemo to učiniti pomoću ključne riječi `break`, dok sa `continue` preskačemo ostatak trenutnog izvođenja i prelazimo na sljedeće ponavljanje.

Zadaci za vježbu

1. **(JS-201)** Napiši petlju koja će ispisati u konzolu sljedeće

```
1  #
2  ##
3  ###
4  ####
5  #####
```

2. **(JS-202)** FizzBuzz. Napiši program koji će ispisati sve brojeve od 1 do 100 uz dvije iznimke. Ukoliko je broj djeljiv s 3 umjesto njega će ispisati "Fizz", ukoliko je djeljiv s 5, umjesto njega će ispisati "Buzz", a ako je djeljiv s 3 i 5, ispisat će umjesto njega "FizzBuzz".

3. **(JS-203)** Napiši program koji ispisuje šahovsko polje koristeći razmak (" ") i znak # :

```
1  # # # #
2  # # # #
3  # # # #
4  # # # #
5  # # # #
6  # # # #
7  # # # #
8  # # # #
```

Napravi program tako da postoji na početku definirana varijabla `velicina` koja označava veličinu kvadrata.