

# Zapiski iz pouka Osnove programiranja

Matej Blagšič

25. oktober 2017

# Kazalo

<b>1</b>	<b>Osnovno</b>	<b>2</b>
<b>2</b>	<b>PRAVILA</b>	<b>2</b>
<b>3</b>	<b>Spremenljivke</b>	<b>3</b>
3.1	Tipi spremenljivk . . . . .	3
<b>4</b>	<b>Izrazi</b>	<b>4</b>
<b>5</b>	<b>Operatorji</b>	<b>4</b>
<b>6</b>	<b>Stavki</b>	<b>6</b>
6.1	Diagram poteka . . . . .	6

# JavaScript

## 1 Osnovno

JavaScript je jezik, ki je integriran v neko okolje, kot je npr. HTML.

```
<script>
<!-- KODA -->
</script>
```

Območje delovanja js kode je:

```
<script src = "../pot do datoteke/imedatoteke.js"><\script>
```

Operator **console.log()**; izpiše vneseno vrednost/spremenljivko v konzolo, ki je dostopna v brskalniku z ukazom Ctrl + Shift + J

## 2 PRAVILA

- angleške črke
- desetiška števila
- podčrtaj ( \_ ) ločuje besede
- začetek stavka ne sme biti število
- loči velike in male črke(je case-sensitive)
- nedovoljena uporaba razerviranih izrazov/funkcij(&, = ...)
- "navednice" označujejo dobessedno navajanje/znak
- primer:

```
console.log(a); <!-- Izpiše vrednost spremenljivke -->
console.log("a"); <!-- Izpiše znak a -->
```

## 3 Spremenljivke

```
var = a;
```

Z enačajem se definira vrednost ali izraz spremenljivki na desni strani(var). Definicija se **VEDNO** konča s podpičjem.

Operator "=" definira spremenljivko in ji priredi neko vrednost ali izraz.

```
a = 31;
```

Spremenljivki a priredimo vrednost 31.

### 3.1 Tipi spremenljivk

Operator **typeof()** vrne tip spremenljivke/vrednosti.

#### Številski(number)

VREDNOSTI: 41, 2.15, Nan, infinity

primer:

```
console.log(typeof (13)) <!-- v konzoli se nam izpiše "number"-->
```

#### Boolov(boolean)

VREDNOSTI: TRUE, FALSE primer:

```
console.log(typeof TRUE) <-- v konzoli se nam izpiše "boolean"-->
```

#### Znakovni niz(string)

VREDNOST: "jabolko" -> string primer:

```
console.log(typeof ("jabolko")) <!-- v konzoli nam izpiše "string"-->
```

## Dedoločen tip(undefined)

VREDNOST: undefined

## 4 Izrazi

Izraz je del kode, ki razreši vrednost. Lahko vrednost dodeli spremenljivki ali pa jo ima sam.

Na primer: Izraz `x = 7` dodeli vrednost 7 spremenljivki `x`.

Pomembna lastnost izrazov je PREDNOST ali *precedence* operatorjev med seboj. To pomeni, da program bere operatorje glede na njihovo prioriteto oz. prednost.

Na primer: V izrazu `a + b * c` ima operator `*` prednost pred operatorjem `+`, zato program najprej zmnoži števili `b` in `c` in nato sešteje vsoto s številom `a`.

Operatorji istega tipa/prednosti se pa izvajajo v vrstnem redu iz leve proti desni (*associativity*).

## 5 Operatorji

### Aritmetični

<code>+, −</code>	Unarni/Binarni
<code>×, ÷</code>	
<code>%</code>	Ostanek pri deljenju

### Primerjalni

Ti operatorji vračajo le *TRUE* ali *FALSE*.

>, >=	Večje, večje ali enako (pomembno je zaporedje znakov!!)
<, <=	Manjše, manjše ali enako
==	Je enako
!=	Ni enako

Prav tako velja tudi: "5" == 5 -> *TRUE*

## Logični

Ti operatorji vračajo le *TRUE* ali *FALSE*.

&&	Logični IN
	Logični ALI
!	Negacija
=	Priredilni operator -> spremenljivki na levi strani priredi vrednost na desni strani.

Prednost in red izvajanja:

<b>Aritmetični</b>	->
<b>Primerjalni</b>	->
<b>Logični</b>	->
<b>Priredilni</b>	<-

Bljižnice (shorthands):

<b>x = x + izraz</b>	=> v spremenljivko x shranimo vsoto spremenljivke asdasdsadasdasdasdasd
<b>x += izraz</b>	=> okrajšan zgornji stavek
<b>x -= izraz</b>	spremenljivki x odštejemo vrednost izraza
<b>x++</b>	=> spremenljivki x se vrednost poveča za 1
<b>x--</b>	spremenljivki x zmanjšamo vrednost za 1

## Vejični

Ima še nižjo prioriteto, kot priredilni operator

## Pogojni

```
pogoj ? ce_je_true : ce_je_false
```

V prvi del pred vprašajem se vnese pogoj in nato izraz, ki se prebere, če je pogoj izpolnjen, po dvopičjem pa sledi izraz, če pogoj ni izpolnjen.

Primer:

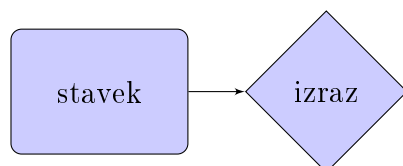
```
var x = 3;  
var y = 0;  
x > 2 ? y = 1 : y = 2;  
console.log(y); --> konzola v tem primeru izpiše 1
```

## Primerjalni

Črke: Črke primerja po abecedi, velike črke so pred malimi. Primerja se od prve do zadnje.

# 6 Stavki

## 6.1 Diagram poteka



Obstaja prazen stavek, ki vsebuje le podpičje. Podpičja so neobvezna v Javascriptu, a jih je vseeno dobro uporabljati.

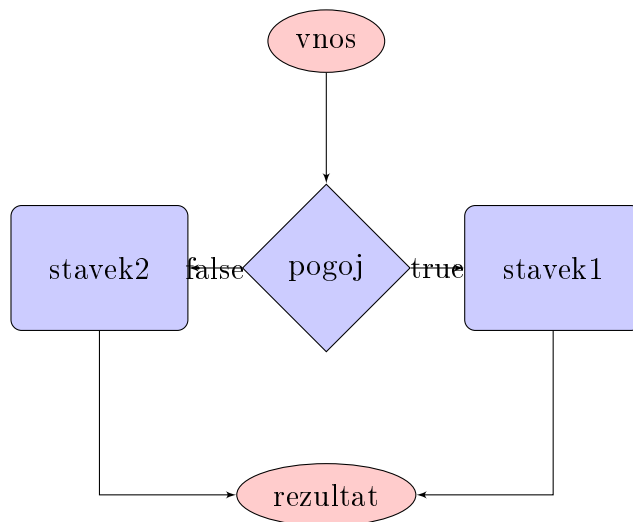
Primer stavka:

```
{  
  stavek1;  
  stavek2;  
  ...  
  stavekN;  
}
```

## Stavek if/else

```
if (pogoj) stavek1 else stavek2
```

Po pogoju, sledi glede na rezultat pogoja **LE EN STAVEK!!** Za več kot en stavek, se uporabi zaviti oklepaj.





Primer stavkov: Računanje idealne teže s podanim podatkom o spolu in višini. Izvozi podatek idealne teže.

```
<script>

    var teza;
    var visina;
    var spol;

    spol = prompt("Vnesi spol (m/z)");

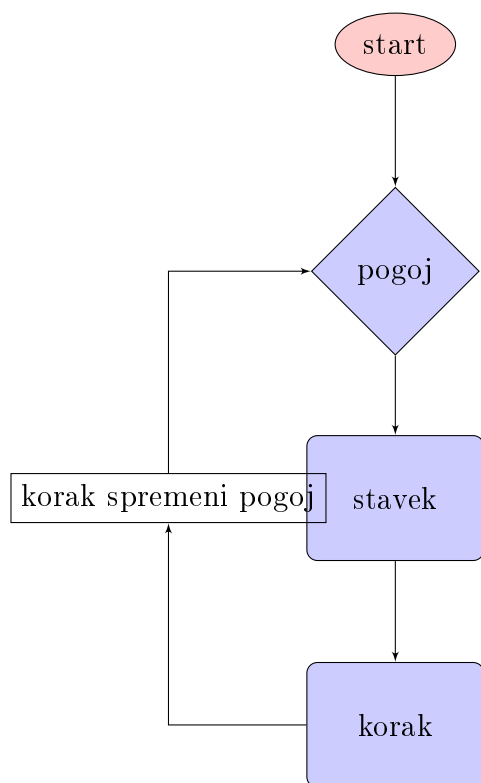
    if(spol == "m"){
        teza = 48 + (visina - 150) * 0.9;
    }
    else{
        teza = 43 + (visina - 150) * 0.7;
    }
    console.log("Tvoja idealna teža je" + teza + "kg");

</script>
```

## Stavek for

for (start; pogoj; korak) stavek

For stavek izvaja določen stavek, dokler je pogoj uresničen. V for stavek se vnese šartni parameter. Ta se preveri v pogoju in se spreminja po koraku.



Npr: začnemo s številom  $i$ , katerega vrednost je 0. Če hočemo stavek ponoviti 4-krat, potem bomo povečevali naš  $i$  do števila 3 (štetje se začne s številom 0) s korakom  $i(i++)$ . To pomeni, da vsakič ko se bo izvedel stavek (več stavkov z  $\{\}$ ) v for stavku, se po glede na nastavljen korak spremenil pogoj za 1 več ( $i++$ ). Torej je v drugem (1-tem) krogu pogoj  $i = 1$  in tako naprej.

```
for(var i = 0; i < 4; i++) console.log("Zdaj se izvajam v " + i + "-tem krogu");
```