

# Spremenljivke

# Spremenljivke

- simbolno ime za neko vrednost v programu
- v spremenljivko shranimo vrednost, ki jo lahko uporabimo poljubno mnogokrat
  - dokler v isto spremenljivko ne zapišemo druge vrednosti
- ime spremenljivke je lahko sestavljeno iz poljubnih črk, števk, ter znakov `_` ali `$`
  - prvi znak imena ne sme biti števka
- **JavaScript ločuje med malimi in velikimi črkami!**
  - `Stevec`  $\neq$  `stevec`  $\neq$  `STEVEC`
- **Ne uporabljamo šumnikov!**

# Spremenljivke

## ■ Spremenljivke običajno pišemo z malimi črkami

- Če je ime sestavljeno iz več besed, jih napišemo skupaj (ime spremenljivke ne sme vsebovati presledka) in vsako besedo (razen prve) napišemo z veliko začetnico

`aktivnaCelica, barvaOzadja`

- namesto presledkov pišemo znak `_`, celotno ime pa napišemo z malimi črkami

`aktivna_celica, barva_ozadja`

## ■ Pred prvo uporabo spremenljivko deklariramo

- Rezervirana beseda `var`

```
var stevec
```

```
var x = 42
```

```
beseda = "JavaScript"
```

```
var preprosto = true
```

# Spremenljivke

- Spremenljivko lahko deklariramo tudi brez rezervirane besede `var`
- V tem primeru ji moramo prirediti začetno vrednost.  
`steviloMesecev = 12`  
`trenutniMesec = "januar"`  
`seminar = null`

# Spremenljivke

- Pri deklaraciji spremenljivke nismo nikjer določili, kakšnega tipa je
  - V JavaScriptu so tipi dinamični - tip spremenljivke samodejno prilagodi vrednosti, ki jo v spremenljivko zapišemo
  - Mogoče je v spremenljivko shraniti vrednost nekega tipa, kasneje pa vrednost kakšnega drugega tipa

```
x = 12
```

```
x = "beseda"
```

```
x = true
```

# Spremenljivke

- `var` deklarira novo lokalno spremenljivko (vidna bo samo znotraj funkcije)
- brez `var` deklariramo novo globalno spremenljivko (če spremenljivka s tem imenom še ne obstaja)
- V funkcijah pred spremenljivke ne pišemo rezervirane besede `var`

## Spremenljivke – konstante

- vrednosti, ki jo dobi spremenljivka ob deklaraciji, ne moremo več spremeniti
- namesto rezervirane besede `var` uporabimo

**besedo** `const`

```
const steviloMesecev = 12
```

```
const visina = 120
```

# Podatkovni tipi



# Podatkovni tipi

- Določajo nabor vrednosti spremenljivke
  - Enostavni - informacijo predstavlja ena vrednost
  - Sestavljeni - informacija je zgrajena iz več vrednosti
- Vsak podatkovni tip ima definiran nabor operacij

# Podatkovni tipi

- osnovni tipi:
  - prazen tip - null
  - logični tip - boolean
  - število - number
  - niz - string
- sestavljeni tipi:
  - objekt - object

## Prazen tip - null

- Prazen podatkov tip ima samo eno vrednost.
- Imenujemo jo `null` (to je rezervirana beseda).

## Logični tip - boolean

- Edini vrednosti logičnega tipa (ali krajše logični vrednosti) sta `true` (res je) in `false` (ni res).
- Obe vrednosti sta rezervirani besedi.

# Število - number

- Število je eden od osnovnih podatkovnih tipov jezika
- JavaScript ne ločuje med celimi in realnimi ali predznačenimi in nepredznačenimi števili.
  - število zapišemo kot celo ali realno, a samo pozitivne vrednosti
- Celó število lahko zapišemo pri:
  - Desetiški osnovi - zaporedje desetiških števk (0–9), pri čemer prva števka ne sme biti 0 (razen ko gre za število 0)
  - Šestnajstiški osnovi - pričnemo z 0x ali 0X, nadaljujemo pa z zaporedjem šestnajstiških števk (0–9, a–f, A–F)
  - Osmiško osnovi - število pa pričnemo z 0, nadaljujemo pa z zaporedjem osmiških števk (0–7)

# Celo število

- Celo število lahko zapišemo pri desetiški, šestnajstiški ali osmiški osnovi.
  - zaporedje desetiških števk (0–9), pri čemer prva števka ne sme biti 0 (razen ko gre za število 0)
  - šestnajstiško celo število pričnemo z 0x ali 0X, nadaljujemo pa z zaporedjem šestnajstiških števk (0–9, a–f, A–F)
  - osmiško celo število pa pričnemo z 0, nadaljujemo pa z zaporedjem osmiških števk (0–7)

# Celo število - primeri

## ■ Celo število:

### ■ Decimalno:

- 25
- 0
- 812376348

### ■ Šestnajstiško:

- 0x5ab7
- 0Xa0F
- 0xFFFF

### ■ Osmiško:

- 0126
- 05

# Realno število

- Sestavljeno iz (nekatero lahko izpustimo):
  - celi del / decimalni simbol / decimalni del / eksponent
- Celi in decimalni del - zaporedji desetiških števk (0–9)
  - celi del ne sme pričeti z ničlo, razen če je to edina števka v tem delu
- Decimalni simbol je pika
- Eksponent je sestavljen iz znaka  $e$  ali  $E$ , morebitnega predznaka plus ali minus in vsaj ene desetiške števke (0–9)
- Realno število mora vsebovati vsaj eno števko v celem in decimalnem delu skupaj, ter decimalni simbol ali eksponent



## Realno število - primeri

- 3.1415
- 10.
- .5
- 3.1E12
- .1e12
- 2e-12

## Niz - string

- poljubna zaporedja znakov, ki jih zapišemo v narekovajih
- ni pomembno, ali uporabimo enojne ali dvojne narekovaje " ali '
  - niz moramo končati z enakim narekovajem, kot smo ga pričeli
- Koristno je tudi (čeprav ni obvezno), da v celem programu uporabljamo samo eno vrsto narekovajev za zapis nizov
- Obstaja tudi prazen niz – niz ki ne vsebuje nobenega znaka
  - Npr. " "

## Niz - string – primeri

- ' '
- " "
- "Janez"
- 'Danes je lep dan.'
- '123'
- "-13.23,,

'Janez" **ni pravilno!**

## Niz - string

- Poleg običajnih znakov (črke, številke, ločila, oklepaji, operatorji ...) lahko niz vsebuje tudi posebne znake, kot so tabulator, znak za novo vrstico ali znaki s posebnim pomenom.
- Poseben znak v nizu opišemo z uporabo znaka \ (backslash), ki mu sledi dejanski opis znaka. P
- V nizu, ki ga napišemo v dvojnih narekovajih, ne moremo uporabiti dvojnega narekovaja, ker bi pomenil konec niza.
  - Rešitev – poseben znak \
  - Primeri \'
  - Če potrebujemo znak\ v nizu, ga moramo zapisati kot \\

## Niz - string

ZNAK	OPIS
<code>\b</code>	znak za skok eno mesto v levo (backspace)
<code>\f</code>	znak za skok na novo stran (form feed)
<code>\n</code>	znak za skok v novo vrstico (new line)
<code>\r</code>	znak za skok na začetek vrstice (carriage return)
<code>\t</code>	znak za skok v naslednji stolpec (tab)
<code>\v</code>	znak za skok v naslednji odstavek (vertical tab)
<code>'</code>	enojni narekovaj (single quote)
<code>"</code>	dvojni narekovaj (double quote)
<code>\\</code>	nagibnica (backslash)
<code>\XXX</code>	znak, predstavljen s kodo Latin1 (tri osmiške številke)
<code>\xXX</code>	znak, predstavljen s kodo Latin1 (dve šestnajstiški številki)
<code>\uXXXX</code>	znak, predstavljen s kodo Unicode (štiri šestnajstiške številke)

## Niz - string

- ' ' ' '
- " " ' "
- "Znak \" je poseben."
- "C:\\Temp\\podatki.dat"

# Objekt - object

- Objekt je edini sestavljeni podatkovni tip
- Vrednosti tega tipa si najlažje predstavljamo kot zbirko poimenovanih ali indeksiranih vrednosti, ki so lahko spet kateregakoli tipa (lahko so tudi objekti)
- Med objekte spadajo datumi, tabele, funkcije
- Več o objektih kasneje v predavanjih!