



# Osnove programiranja

Osnove programiranja

Tipovi podataka

# Povezana literatura

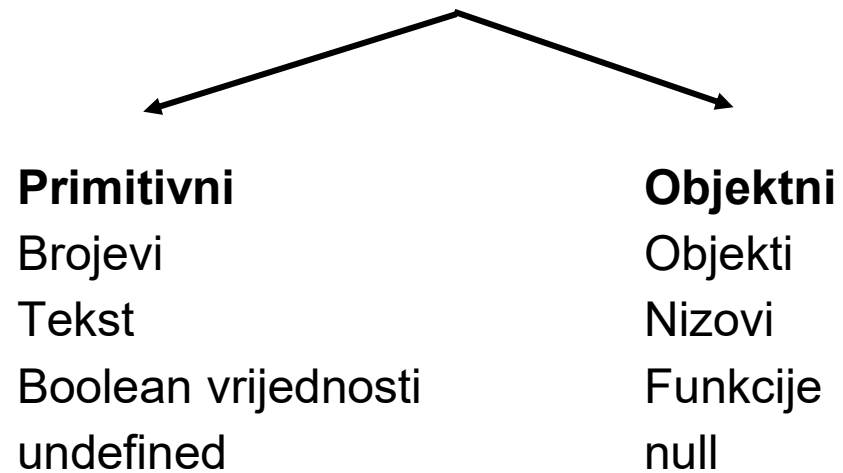
*JavaScript: The Definitive Guide*

- Part I. Core JavaScript, 3. Types, values and variables

[https://www.w3schools.com/js/js\\_numbers.asp](https://www.w3schools.com/js/js_numbers.asp)

[https://www.w3schools.com/js/js\\_strings.asp](https://www.w3schools.com/js/js_strings.asp)

# Tipovi podataka u JavaScriptu



# Promjenjive i nepromjenjivi tipovi

Također se mogu kategorizirati kao promjenjivi i nepromjenjivi tipovi. Nepromjenjivi tipovi podatka su oni čije se stanje ne može promijeniti nakon što se podatak stvori.

**Primitivni** tipovi podataka (i null) su **nepromjenjivi**.

**Objektni** tipovi podataka su **promjenjivi**.

# Brojevi

**Integer** je cjelobrojna vrijednost (bez decimalnih vrijednosti).

**Float** je broj sa decimalnom vrijednosti, obično zauzima više memorijskog prostora.

Svi brojevi u JavaScriptu su **floating point** tip broja, i spremaju se u 64-bitnom zapisu!

# Tekst

Tekst se u JavaScriptu sprema u **string** tipu podatka.  
Prepoznavamo ga po navodnicima.

```
var x = 'Lorem ipsum';  
var y = "Lorem ipsum dolor sit amet";
```

## Boolean tip podatka

Predstavlja istinu ili neistinu, uključeno ili isključeno, da ili ne. Postoje samo dvije moguće vrijednosti ove vrste: **true** ili **false**.

**Truthy** i **falsey** vrijednosti su one koje se konvertiraju u true i false kada izraz očekuje Boolean vrijednost.

```
var x = Boolean(null);    // x = false  
var x = Boolean({});      // x = true
```



## null i undefined

**null** je ključna riječ koja se obično koristi da naznači odsutnost neke vrijednosti.

**undefined** je vrijednost varijabli koje nisu inicijalizirane i vrijednost koju dobivate kada pokušavate dohvatiti vrijednost objekta ili elementa niza koji ne postoji. Također ćete dobiti undefined za izvršenje funkcije koja ne vraća nikakvu vrijednost.

# typeof operator

Možete koristiti operator `typeof` da biste pronašli vrstu podataka.

```
typeof "John"           // "string"  
typeof 3.14             // "number"  
typeof NaN              // "number"  
typeof false            // "boolean"  
typeof [1,2,3,4]        // "object"  
typeof {name:'John', age:34} // "object"
```

# Automatska konverzija podataka

JavaScript obavlja mnoge konverzije tipa podatka automatski ovisno o kontekstu u kojemu se podatak koristi, te operatoru.

Pročitajte više o konverziji  
primitivnih vrijednosti u  
JavaScript: The Definite Guide

```
> '5' - 3
< 2
> '5' + 3
< "53"
> '5' - '4'
< 1
> '5' + + '5'
< "55"
> 'foo' + + 'bar'
< "fooNaN"
> '5' + - '2'
< "5-2"
> '5' + - + - - + - - + + - + - + - - - '2'
< "5-2"
> x = 3
< 3
> '5' + x - x
< 50
> '5' - x + x
< 5
```

# Tipovi podataka - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 20min

Osnove programiranja

Rad sa tekstom

# Povezana literatura

*JavaScript: The Definitive Guide*

- Part I. Core JavaScript, 3. Types, values and variables

[https://www.w3schools.com/js/js\\_numbers.asp](https://www.w3schools.com/js/js_numbers.asp)

[https://www.w3schools.com/js/js\\_strings.asp](https://www.w3schools.com/js/js_strings.asp)

# String

String je niz znakova (0 ili više) okružen navodnicima.

Navodnici mogu biti jednostruki ili dvostruki.

Brojanje pozicije znaka u stringu počinje od **nule!**

```
var myString = "";           // prazan string
myString = "Lorem ipsum";
myString = "  Lorem ipsum"; // string sa prazninama na početku
```

# Konkatenacija

Operacija spajanja dva ili više stringova.  
U JavaScriptu se koristi operator `+`

```
x = "Hello, " + "World";  
x = "Hello, " + 5;
```

```
var x = "Ovo je vrlo dugi string" +  
        "koji je u dva reda " +  
        "jer se inače ne može čitati.";
```

```
var x = "Ovo je vrlo dugi string \  
        "koji je u dva reda \  
        "jer se inače ne može čitati.";
```



# Posebni znakovi

Neki znakovi unutar stringa se moraju "izbjeći" kako bi se prikazali.  
To radimo sa znakom \ (backslash).

\'	Jednostruki navodnici
\"	Dvostruki navodnici
\\	Backslash
\n	Nova linija
\r	Carriage Return

```
var x = "Ovo je moj \"string\"";
```

# Metode

length	Duljina stringa
charAt	Vraća znak na poziciji
trim	Miće praznine na početku i kraju stringa
indexOf	Vraća poziciju stringa unutar stringa
split	Pretvara string u niz - više o tome kasnije!
replace	Vraća novi string sa zamijenjenim izrazom
toUpperCase	Pretvara stringu u sva velika slova

**Pazite!** Neke metode mijenjaju izvorni string, a neke vraćaju novi string.

# Rad sa tekstom - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 30min

Osnove programiranja

Operatori

# Povezana literatura

*JavaScript: The Definitive Guide*

- Part I. Core JavaScript, 4. Expressions and operators

[https://www.w3schools.com/js/js\\_operators.asp](https://www.w3schools.com/js/js_operators.asp)

[https://www.w3schools.com/js/js\\_comparisons.asp](https://www.w3schools.com/js/js_comparisons.asp)

[https://www.w3schools.com/js/js\\_booleans.asp](https://www.w3schools.com/js/js_booleans.asp)

# Tipovi operatora

*Po tipu operacije*

Logički	&&    !
Arifmetički	+ - * / % ** ++ --
Operatori dodjeljivanja	= += -= *= /= %= **=
Relacijski	== === != !== > < >= <= ?
Bitovni (bitwise)	&   ~ ^ << >> >>>
Operator konkatencije	+
Ostali	delete, typeof, void, instanceof, in

# Tipovi operatora

*Po broju operandata*

1. Unarni
2. Binarni
3. Ternarni - kombinira 3 operacije u jednu

```
var x += 1; // Unarni operator
var x = y + 1; // Binarni operator
var x = y ? y : x + 1; // Ternarni operator
```

# Aritmetički operatori

-	oduzimanje	$x = x - 1$	$x -= 1$
+	zbrajanje	$x = x + 1$	$x += 1$
*	množenje	$x = x * 2$	$x *= y$
/	podjela	$x = x / 2$	$x /= 2$
%	modulo (ostatak dijeljenja)	$x = x \% 2$	$x \% = 2$
**	potenciranje (ES6)	$x = x ** 2$	$x ** = 2$
--	dekrement	$x--$ ili $--x$	Različito ponašanje ovisno o kontekstu
++	inkrement	$x++$ ili $++x$	



# Relacijski operatori

Testiraju relativni poredak (brojčani ili abecedni) njihova dva operanda:

==	jednaka vrijednost
===	jednaka vrijednost i jednaki tip podatka
!=	Nije jednako
!==	nije jednaka vrijednost ili nije jednaki tip podatka
>	više od
<	manje od
>=	više ili jednako
<=	manje ili jednako

# Logički izrazi

&&	i
	ili
!	ne

$$\neg(p \ \&\& \ q) \equiv \neg p \ || \ \neg q$$

$$\neg(p \ || \ q) \equiv \neg p \ \&\& \ \neg q$$

# Operatori - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 30min

Osnove programiranja

Izjave

## Povezana literatura

*JavaScript: The Definitive Guide*

- Part I. Core JavaScript, 5. Statements

[https://www.w3schools.com/js/js\\_if\\_else.asp](https://www.w3schools.com/js/js_if_else.asp)

[https://www.w3schools.com/js/js\\_switch.asp](https://www.w3schools.com/js/js_switch.asp)

[https://www.w3schools.com/js/js\\_loop\\_for.asp](https://www.w3schools.com/js/js_loop_for.asp)

[https://www.w3schools.com/js/js\\_loop\\_while.asp](https://www.w3schools.com/js/js_loop_while.asp)

## Vrste izjava

**Uvjeti** izvršavaju ili preskaču druge izjave ovisno o vrijednosti određenog izraza, tj. granaju kod na više mogućnosti.

**Petlje** repetitivno izvršavaju dijelove koda.

Izjave **skokova** (poput break, return, throw) naređuju interpreteru da skoči na neki drugi dio programa.

# Uvjeti

*If, else if*

Ako je izraz truthy, onda se izvrši izjava/e unutar zagrada. Inače se izjava preskače.

**Else** nam daje alternativu za slučaj da je izraz falsey.

Možemo koristiti i **else if** ako želimo postaviti dodatan uvjet.

```
// If - else
if (izraz) {
    // Izvrši kod u bloku #1
} else {
    // Izvrši kod u
    bloku #2
}
```

# Uvjeti

*If, else if*

```
// Jednostavan if
if (izraz) {
    // Izvrši kod u bloku #1
}
```

```
// If - else if
if (izraz) {
    // Izvrši kod u bloku
    #1
} else if (izraz2) {
    // Izvrši kod u
    bloku #2
}
```



# Uvjeti

*switch*

Switch izjava je bolja alternativa kompliciranim if izjavama.

```
switch(n) {  
    case 1: // Execute code block #1. break;  
    case 2: // Execute code block #2. break;  
    default: // Execute code block #3. break;  
}
```



Smjer provjere

# Petlje

*while, do ... while*

```
var count = 0;
while (count < 10) {
    console.log(count);
    count++;
}
```

```
var count = 0;
do {
    console.log(count);
    count++;
} while (count < 10);
```

# Petlje

*for, for ... in*

```
for (var count = 0; count < 10; count++) {  
    console.log(count);  
}
```


For ... in petlja se koristi sa objektnim tipovima podataka.

```
for (var i = 0; i < a.length; i++) {  
    console.log(a[i]);  
}
```

# Izjave

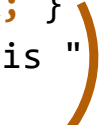
*break, continue*

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { continue; }  
    text += "The number is " + i + "<br>";  
}
```



**continue** izjava preskače obradu trenutnog člana petlje i nastavlja sa sljedećim

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { break; }  
    text += "The number is " + i + "<br>";  
}
```



**break** izjava prekida petlju i izlazi iz nje

# Izjave - vježba

Pratite upute u Upute.txt

Vrijeme trajanja vježbe: 20min

Osnove programiranja - 1.dio

# Vježbe

# Zadaci

Pratite Upute.txt

Trajanje vježbe: 45min