



Lessons[2] = 'Operators & Strings';

Operators & Strings in JavaScript

Questions, questions, questions

- Šta su to komentari? Za šta nam služe?
- Šta je to sintaksa?
- Šta su varijable?
- Šta su browseri?
- Šta je to Node.js?
- Šta je to compiler?
- Šta je to interpreter?
- Šta je to JavaScript engine?

Tipovi podataka

Svaka vrijednost u JavaScript ima tip podataka.

Primitivni tipovi podataka:

- Number
- BigInt
- String
- Boolean
- undefined
- null
- Symbol

Objektni tip podataka:

- Object

Tipovi podataka



VS Code

```
1 let number = 123;           // Number
2 let name = "Giovanni Giorgio"; // String
3 let largeNumber = 900719925474099101230123n; // BigInt
4 let isThisLightTheme = true; // Boolean
5 let doesNotHaveValue = undefined; // undefined
6 let absenceOfValue = null;      // null
```

typeof Operator

typeof operator nam omogućava da saznamo tip podatka neke vrijednosti.



VS Code

```
1 console.log(typeof 123);  
2 console.log(typeof 'Giovanni Giorgio');  
3 console.log(typeof 900719925474099101230123n);  
4 console.log(typeof true);  
5 console.log(typeof undefined);  
6 console.log(typeof null);
```

Operatori

Operator u programskom jeziku govori programu da izvrši neku operaciju. DUH

Vrste operatora u Javascript:

- Aritmetički
- Operatori poređenja
- Logički
- Operatori dodjele
- Inkrement/dekrement

Arithmetički operatori

Simbol	Operacija	Korištenje	Opis	Primjeri
+	sabiranje	$a + b$	Sabira brojeve sa obje strane znaka. (<i>suma</i>)	$7 + 5$ (12)
-	oduzimanje	$a - b$	Od lijevog broja oduzima desni. (<i>razlika</i>)	$7 - 5$ (2)
*	množenje	$a * b$	Množi brojeve sa obje strane znaka. (<i>proizvod</i>)	$7 * 5$ (35)
/	dijeljenje	a / b	Dijeli lijevi operand sa desnim. (<i>količnik</i>)	$7 / 5$ (1.4)
%	modulo	$a \% b$	Dijeli lijevi operand sa desnim, te vraća ostatak nakon tog dijeljenja. (<i>ostatak</i>)	$7 \% 5$ (2)

Arithmetički operatori



VS Code

```
1 let broj1 = 4;  
2 let broj2 = 6;  
3  
4 let razlika = broj1 - broj2;  
5  
6 console.log(razlika);
```


Zadatak

Izvršiti sljedeće operacije nad brojevima 12, 5, spremiti u varijable, zatim ispisati rezultate u konzolu.

+	sabiranje
-	oduzimanje
*	množenje
/	dijeljenje
%	modulo

Simbol	Operacija	Korištenje	Opis	Primjeri
<code>==</code> , <code>===</code>	jednakost	<code>a == b</code> , <code>a === b</code>	Provjerava da li su obje vrijednosti operanada jednake.	<code>7 == 5</code> , <code>7 === 5</code> (false)
<code>!=</code>	nejednakost	<code>a != b</code>	Provjerava da li su vrijednosti operanada različite.	<code>7 != 5</code> (true)
<code>></code>	Veće od	<code>a > b</code>	Provjerava da li je vrijednost lijevog operanda veća od vrijednosti desnog.	<code>7 > 5</code> (true)
<code><</code>	Manje od	<code>a < b</code>	Provjerava da li je vrijednost lijevog operanda manja od vrijednosti desnog.	<code>7 < 5</code> (false)
<code>>=</code>	Veće ili jednako	<code>a >= b</code>	Provjerava da li je vrijednost lijevog operanda veća ili jednaka vrijednosti desnog.	<code>7 >= 5</code> (true)
<code><=</code>	Manje ili jednako	<code>a <= b</code>	Provjerava da li je vrijednost lijevog operanda manja ili jednaka vrijednosti desnog.	<code>7 <= 5</code> (false)

Operatori poređenja



VS Code

```
1 let broj1 = 4;  
2 let broj2 = 6;  
3  
4 let poredjenje2 = broj1 !== broj2;  
5  
6 console.log(poredjenje2);
```

Zadatak

Izvršiti sljedeće operacije nad proizvoljnim brojevima spremiti u varijable, zatim ispisati rezultate u konzolu.

>	Veće od
<	Manje od
==, ===	Jednakost
!=	Nejednakost
<=	Manje ili jednako

Operatori dodjele

Simbol	Operacija	Korištenje	Opis	Primjeri (a je 10)
=	dodjela	$a = b$	Dodjeljuje vrijednost desnog operanda lijevom.	$a = 5$ (5)
+=	Sabiranje i dodjela	$a += b$	Sabira vrijednost lijevog operanda sa desnim, te rezultat dodjeljuje lijevom operandu.	$a += 5$ (15)
-=	Oduzimanje i dodjela	$a -= b$	Oduzima vrijednost desnog operanda od lijevog, te rezultat dodjeljuje lijevom operandu.	$a -= 5$ (5)
*=	Množenje i dodjela	$a *= b$	Množi vrijednost lijevog operanda sa desnim, te rezultat dodjeljuje lijevom operandu.	$a *= 5$ (50)
/=	Dijeljenje i dodjela	$a /= b$	Dijeli vrijednost lijevog operanda sa desnim, te rezultat dodjeljuje lijevom operandu.	$a /= 5$ (2)
%=	Modulo i dodjela	$a \% = b$	Dijeli vrijednost lijevog operanda sa desnim, te ostatak dodjeljuje lijevom operandu.	$a \% = 5$ (0)

Operatori dodjele



VS Code

```
1 let b = 12;  
2 let c = 5;  
3  
4 b %= 5;  
5  
6 console.log(b);
```

Zadatak

Izvršiti sljedeće operacije nad proizvoljnim brojevima, zatim ispisati rezultate u konzolu.

=	Dodjela
%=	Modulo i dodjela
*=	Množenje i dodjela
/=	Djeljenje i dodjela

Logički operatori

Simbol	Operacija	Korištenje	Opis	Primjeri
&&	Logičko “i” (AND)	a && b	Rezultira sa true samo u slučaju kada su oba operanda true .	true && false (false)
	Logičko “ili” (OR)	a b	Rezultira sa true kada je bar jedan od operanda true .	true false (true)
!	Logička negacija (NOT)	!a	Unarni operator. Rezultira sa true ako je operand false , tj rezultira sa false ako je operand true .	!true (false)

Logički operatori

AND (&&)

A	B	A && B
<i>false</i>	<i>false</i>	false
<i>false</i>	<i>true</i>	false
<i>true</i>	<i>false</i>	false
<i>true</i>	<i>true</i>	true

OR (||)

A	B	A B
<i>false</i>	<i>false</i>	false
<i>false</i>	<i>true</i>	true
<i>true</i>	<i>false</i>	true
<i>true</i>	<i>true</i>	true

NOT (!)

A	!A
<i>false</i>	true
<i>true</i>	false

Logički operatori

VS Code

```
1 let broj1 = 12;  
2 let broj2 = 7;  
3 let broj3 = 5;  
4  
5 console.log((broj1 > broj2) && (broj2 > broj3));  
6  
7 console.log(broj1 < broj2 && broj2 < broj3);
```

Inkrement/Dekrement

Simbol	Operacija	Korištenje	Opis	Primjeri (a = 5)
++	Post inkrement	a++	Poveća vrijednost a za 1 nakon korištenja vrijednosti a .	a++ (6)
--	Post dekrement	a--	Smanji vrijednost a za 1 nakon korištenja vrijednosti a .	a-- (4)
++	Pre inkrement	++a	Poveća vrijednost a za 1 prije korištenja vrijednosti a .	a++ (6)
--	Pre dekrement	-- a	Smanji vrijednost a za 1 prije korištenja vrijednosti a .	a-- (4)

Inkrement/Dekrement

VS Code

```
1 let a = 5;  
2  
3 console.log(a++);  
4 console.log(++a);  
5 console.log(a--);  
6 console.log(--a);
```

Stringovi



VS Code

```
1 let string1 = "Neki string.";
2 let string2 = 'Jos jedan string.';
3 let string3 = `Necete vjerovati, ali jos jedan string.`;
```



Chars	N	e	k	i		s	t	r	i	n	g	.
Index	0	1	2	3	4	5	6	7	8	9	10	11

Chars	N	e	k	i		s	t	r	i	n	g	.
Index	0	1	2	3	4	5	6	7	8	9	10	11

→ `.length` → Vraća dužinu stringa

→ `[index]` ili `.charAt(index)` → Vraća karakter na određenom indexu


```
VS Code

1 let string = 'Neki string.';
2
3 console.log(string.length);
4
5 console.log(string[3]);
6 console.log(string.charAt(3));
```

Dodatne metode

- `.concat` → Spaja jedan string sa drugim
- `.includes` → Vraća true ili false u zavisnosti od toga da li jedan string sadrzi drugi
- `.indexOf` → Vraća index na kojem se nalazi neki karakter
- `.replace` → Mijenja jedan određeni karakter u stringu sa nekim drugim
- `.toUpperCase` → Vraća isti string sa svim velikim slovima
- `.toLowerCase` → Vraća isti string sa svim malim slovima
- `.trim` → Uklanja prazna mjesta sa početka i kraja stringa

Dodatne metode



VS Code

```
let name = 'Jon';  
let lastName = 'Snow';  
  
let fullName = name.concat(lastName);  
  
console.log(fullName);  
  
console.log(fullName.toUpperCase());  
  
console.log(fullName.replace('J', 'D'));
```


Concatenation

Vidjeli smo kako možemo sastaviti dva stringa koristeći se `.concat()` metodom.

Isti efekat možemo postići koristeći se već poznatim znakom `“+”`.




VS Code

```
let name = 'Jon';  
let lastName = 'Snow';  
  
let fullName = name + lastName;  
  
console.log(fullName);
```

Interpolation

U JavaScript, možemo ubaciti, ili *interpolirati*, varijable u string koristeći se *template literals*.



VS Code

```
let num1 = 3;  
let num2 = 6;  
  
let message = `Suma brojeva ${num1} i ${num2} je ${num1 + num2}`;  
  
console.log(message);
```

Explicit Conversion

JavaScript nam omogućava da pretvaramo jedan tip podatka u drugi.

Bilo koji tip podataka je moguće pretvoriti u jedan od sljedećih:

- Number koristeći: *Number(value)*
- String koristeći: *String(value)*
- Boolean koristeći: *Boolean(value)*

To Number Conversion



VS Code

```
1 Number("42");           // 42
2 Number("");              // 0
3 Number(true);            // 1
4 Number(false);           // 0
5 Number(null);            // 0
6 Number(undefined);       // NaN
```

To String Conversion



VS Code

```
1 String("42");           // "42"  
2 String(true);           // "true"  
3 String(false);          // "false"  
4 String(null);           // "null"  
5 String(undefined);      // "undefined"
```

To Boolean Conversion



VS Code

```
1 Boolean(' ');           // false
2 Boolean(0);              // false
3 Boolean(-0);             // false
4 Boolean(NaN);            // false
5 Boolean(null);           // false
6 Boolean(undefined);     // false
7 Boolean(false);         // false
```

Implicit Conversion -> Coercion

U slučaju kada vršimo operacije nad vrijednostima različitog tipa, desava se Coercion.

Najčešći primjer s kojim ćete se susretati je kada koristite operator `==`

NE morate znati sva pravila na pamet, ali morate biti svjesni da Coercion postoji.

Implicit Conversion -> Coercion



VS Code

```
1 console.log(1 == '1');           // true
2 console.log(1 === '1');          // false
3 console.log('1' == true);        // true
4 console.log('1' === true);       // false
5 console.log(1 + '1');             // 11
6 console.log('1' + true);          // 1true
```


JS

That's it folks!

Questions?

That's all Folks!