Slovenska univerza za tretje življenjsko obdobje

Sofinancira program
Evropske unije
Erasmus+

JavaScript za uporabo na spletnih straneh – operatorji

Računski operatorji

Operator	Opis
+	Seštevanje
-	Odštevanje
*	Množenje
/	Deljenje
%	Modulus (ostanek od deljenja)
++	Povečanje za 1
--	Pomanjšanje za 1

Dodelitveni operatorji

Operator	Primer	Pomeni enako, kot
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y

Operatorji nizov

- Operator + lahko uporabimo tudi za združevanje nizov.

- Primer 1:

```
var txt1 = „Neznana";
var txt2 = „oseba";
var txt3 = txt1 + " " + txt2; //rezultat je Neznana oseba
```

- Primer 2:

```
var txt1 = „Kako lep, ";
txt1 += „sončen dan!";  // rezultat bo Kako lep sončen dan!
```

Združevanje nizov in številk

- Rezultat seštevka dveh števil bo vsota, toda rezultat združevanja niza in številke bo niz.

- Primeri:

```
var x = 5 + 5;                  // rezultat bo 10
var y = "5" + 5;                // rezultat bo 55
var z = „Zdravo" + 5;           // rezultat bo Zdravo5
var z1 = 5 + 5 + „Zdravo“ + 5 + 5 // rezultat bo 10Zdravo55
```

Primerjalni operatorji

Operator	Opis
==	je enako
===	enaka vrednost in enaka vrsta podatka
!=	ni enako
!==	ni enaka vrednost ali ni enaka vrsta podatka
>	večje kot
<	manjše kot
>=	večje kot ali enako
<=	manjše kot ali enako
?	pogojni operator

Pogojni operator

- Pogojni operator dodeli vrednost spremenljivki na podlagi danega pogoja. Sintaksa:

  imespremenljivke = (pogoj) ? vrednost1:vrednost2

- Primer: var lahkoglasujem = (leta < 18) ? „Premlad/a":„Dovolj star/a";

Pogojni operator - primer

```
<!DOCTYPE html>
<html>
<body>
	<p>Vnesite leta in kliknite na gumb:</p>
	<input id="leta" value="18" />
	<button onclick="mojaFunkcija()">Poskusi</button>
	<p id="demo"></p>
	<script>
		function mojaFunkcija() {
		var leta, lahkoglasujem;
		leta = document.getElementById("leta").value;
		lahkoglasujem = (leta < 18) ? "Premlad/a,":"Dovolj stara,";
		document.getElementById("demo").innerHTML = lahkoglasujem + " da glasuješ.";
	}
	</script>
</body>
</html>
```

Primerjalni operatorji

Če je x = 5, si v spodnji tabeli poglejmo kakšne rezultate nam dajo primerjalni operatorji:

Operator	Opis	Primerjava	Rezultat	
==	je enako	x == 8	false	false (slov. ne drži) true (slov. drži)
		x == 5	true	
		x == "5"	true	
===	enaka vrednost in enaka vrsta podatka	x === 5	true	
		x === "5"	false	
!=	ni enako	x != 8	true	
!==	ni enaka vrednost ali ni enaka vrsta podatka	x !== 5	false	
		x !== "5"	true	
		x !== 8	true	
>	večje kot	x > 8	false	
<	manjše kot	x < 8	true	
>=	večje ali enakogreater than or equal to	x >= 8	false	
<=	manjše ali enako	x <= 8	true	

Logični operatorji

Operator	Opis
&&	logično in
||	logično ali
!	ni logično

Logični operatorji

Če je x = 6 in y = 3, si v spodnji tabeli poglejmo rezultate logičnih operatorjev:

Operator	Opis	Primer	
&&	in	(x < 10 && y > 1) is true	true (slov. drž) false (slov. ne drži)
||	ali	(x == 5 || y == 5) is false	
!	ni	!(x == y) is true	

Operatorji tipov (vrst)

- V JavaScriptu ločimo 5 vrst podatkov, ki vsebujejo vrednost:
  - Niz (angl. string)
  - Številka (angl. number)
  - boolean (drži, ne drži; angl. true, false)
  - Objekt (angl. object)
  - Funkcija (angl. function)
- Tri vrste objektov:
  - Objekt (angl. object)
  - Datum (angl. date)
  - vrsta nizov (angl. array)
- Dve vrsti tipa podatkov, ki ne morejo vsebovati vrednosti:
  - Nič (angl. null)
  - Nedefinirano (angl. undefined)

Operatorji tipov (vrst)

Operator	Opis
typeof	Pove, kakšen tip podatka je spremenljivka
instanceof	Vrne true (slov. drži), če je objekt instanca tipa objekt

Rezultat typeof operatorja je vedno niz.

Operator typeof

```
typeof "Janez"                    // izpiše "string"
typeof 3.14                       // izpiše "number"
typeof NaN                        // NaN pomeni ni številka (angl. not a number.
                                     izpiše "number"
typeof false                      // izpiše "boolean"
typeof [1,2,3,4]                  // izpiše "object"
typeof {ime:'John', starost:34}   // izpiše "object"
typeof new Date()                 // izpiše "object"
typeof function () {}             // izpiše "function"
typeof avto                       // izpiše "undefined"
typeof null                       // izpiše "object"
```

Lastnost constructor

- Lastnost constructor pove funkcijo konstruktorja spremenljivke (ali je to niz, številka, boolean, objekt, vrsta nizov, datum, funkcija,…). Typeof operator nam pove samo, da gre za objekt, medtem ko nam lastnost konstruktor pove, za kakšno vrsto objekta gre.
- Primeri:

```
„Janez".constructor                          // izpiše function String()  {[native
                                                code]}
(3.14).constructor                           // izpiše function Number()  {[native code]}
false.constructor                            // izpiše function Boolean() {[native code]}
[1,2,3,4].constructor                        // izpiše function Array()   {[native code]}
{ime:'John',starost:34}.constructor          // izpiše function Object()  {[native code]}
new Date().constructor                       // izpiše function Date()    {[native code]}
function () {}.constructor                   // izpiše function Function(){[native code]}
```

Vrste podatkov

- Vrste v JavaScriptu so dinamične, kar pomeni, da lahko ista spremenljivka vsebuje več vrednosti (vrst podatkov):

```
var x;                  // zdaj je nedefinirana (angl. undefined)
x = 5;                  // zdaj je številka (angl. number)
x = „Janez";            // zdaj je niz (angl. string)
```

- Nize lahko zapisujemo znotraj enojnih ali dvojnih narekovajev, vendar:

```
var answer = "It's alright";                 // enojni narekovaj znotraj
                                                dvojnih narekovajev
var answer = "He is called 'Johnny'";        // enojna narekovaja znotraj
                                                dvojnih narekovajev
var answer = 'He is called "Johnny"';        // dvojni narekovaji znotraj
                                                enojnih narekovajev
```

Vrste podatkov

- Zapis številk:

```
var x1 = 34.00;          // zapisano z decimalko
var x2 = 34;             // zapisano kot celo število
```

- Booleani:

```
var x = 5;
var y = 5;
var z = 6;
(x == y)      // rezultat je true (slov. drži)
(x == z)      // rezultat je false (slov. ne drži)
```

Vrste podatkov

- **Vrste nizov (angl. arrays)** uporabljamo za deklariranje več spremenljivk istočasno, ki jih zapišemo v oglate oklepaje:

var avtomobili = ["Saab", "Volvo", „Fičo“];

Nizi so indeksirani na temelju ničle, kar pomeni, da je prva postavka [0], druga [1] itd.

- Vrednost **nedefinirano (angl. undefined)** nam vedno da vrednost undefined in vrsto podatka (typeof operator) undefined:

var car;

car = undefined;

V obeh primerih je vrednost spremenljivke in vrsta podatka undefined.

- Prazna vrednost nima nič skupnega z nedefiniranim. Spremenljivka var car = „“; ima vrednost „“ in vrsta podatka, ki ga vsebuje je string.

Vrste podatkov

- Razlika med vrednostima null in undefined: JavaScript vrednost null obravnava kot objekt, medtem ko vrednost undefined obravnava kot nedefinirano.
  - var oseba = {ime:„Neznana", priimek:„oseba", starost:50, barva_oci:„modra"};
    oseba = null; // zdaj je vrednost spremenljivke osebe nič, vendar
    je vrsta podatka, ki ga nosi, še vedno objekt
  - var oseba = {ime:„Neznana", priimek:„oseba", starost:50, barva_oci:„modra"};
    oseba = undefined; // zdaj sta vrednost in vrsta spremenljivke
    nedefinirani
  - typeof undefined // nedefinirano
    typeof null // objekt

    null === undefined // izpiše false (slov. ne drži), ker vrednost in tip nista
    enaka
    null == undefined // izpiše true (slov. drži), drži, ker je vrednost obeh
    enaka