**Java Script – uvod**

* Moze se ubaciti kao eksterni file. Na taj nacin krijemo kod od korisnika.
* <script src="javascript.js"></script>
* Viselinijski komentari u kodu pisu se : */\* komentar \*/*
* Obican komentar : *//Komentar…*

Provera kojeg je tipa var :

var testvar= null;

alert (testvar); //shows null

alert (typeof testvar); //shows object

ako ne bi bilo NULL onda bi izbacilo na kraju UNDEFINED sto znaci da nije data vrednost varijabli. Ako stavimo null, onda iznacuje OBJECT. Ovo je primer kako proveravamo kog je tipa varijabla.

var first = 1;

console.log( first, "first");

//comparison operators

var x = 5;

var y = 2;

console.log(x == y); //false

console.log(x != y); //true

console.log(x > y); //true

console.log(x < y); //false

console.log(x >= y); //true

console.log(x <= y); //false

//asignment operators

var x = 5;

x += 2; // 7

console.log(x);

x -= 2; // 5

console.log(x);

x \*= 2; // 10

console.log(x);

x /= 2; // 5

console.log(x);

x %= 2; // 1

console.log(x);

**Variables** – Variables are used to store data in memory.

!= - must not be equal

Prvo se deklarise, onda joj se daje vrednost :

var number; // Declaration

number = 5; // Initialization

var number = 5; // Declaration and Initialization

**Primitivni tipovi podataka (vrednosti)**

* **Null**
* **Undefined**
* **Numbers**
* **Strings**
* **Booleah**

**a++** operacija, prvo izbacuje pocetnu vrednost, pa onda uvecava za 1.

**++a** operacija, odmah sabira.

**ASSIGNMENT OPERATORS**

+= a += b a = a + b

-= a -= b a = a - b

\*= a \*= b a = a \* b

/= a /= b a = a / b

%= a %= b a = a % b

**Cas 2**

**Strings** – konkatenacija (povezivanje stringova uz)

Mogu se ispisivati uz dvostruke ili jednostruke navodnike.

* Konkatenacija (“rezultat” **+** **“** zbog razmaka **” +** “drugi rezultat”)
* Interpolacija ( ` $ { rezultat } ` )

var rez1 = `tekst ${prom1} neki tekst ${prom2} još teksta`;  
var rez2 = "tekst " + prom1 + " neki tekst " + prom2 + " još teksta";

const fullName = firstName + " " + lastName;

var rez = prom1.concat(prom2) + "!";

– combining strings known as concatenation.

* console.log (“rezultat”)
* console.log ( ` $ { rezultat } ` )

**NaN** – not a number (matematicka operacija u kojoj nisu obe promenljive bojevi ce izbaciti NaN).

**Proverava da li je broj** console.log (Number.isNaN(b))

**Infinity** –

**LOGICAL OPERATORS**

* AND&& (svi uslovi moraju biti tacni da bi sve bilo true)
* var rez = (statusA === "open" && longitude > 0); // true and true = true
* OR|| (jedan je dovoljan kao tacan da bi sve bilo tacno)
* var rez = (statusA === "open" || latitude > 0);
* NOT! (! Ispred true daje false I obratno)

var rez2 = (statusA !== "open");

console.log("rez", rez2); //false

**FALSE/TRUE**

**FALSY VALUES:**

false

0, -0

"" (empty string)

null

undefined

NaN (invalid number)

**TRUTHY VALUES:**

true

"hello", "0"

25

[ ], [ 1, "2", 3 ] (arrays)

{ }, { a: 42 } (objects)

function foo() { .. } (functions)

everything else :

**IF ELSE**

      var x = 5;

      var a = 10;

      var b = 20;

      var z = ( x > 10 ? a : b);

Ako je vece od 10 bice a, ako nije, bice b.

let pare = prompt ("Koliko imas para?");

let kolikoimaspara = parseInt(pare);

if (kolikoimaspara <=500) {

    console.log ("Kupi nesto");

}

else {

    console.log ("Nema para");

}

parseInt – pretvara string broj

var unos = imevarijable ("Unesite broj od 0 do 11:");

// let varijablajedan = parseInt(imevarijablje);

if (**isNaN**(unos) || unos < 0 || unos > 11) {

console.log("Nevažeći unos. Molimo vas da unesete broj od 0 do 11.");

} else {

const wage = 500;

// const role = "Zeka peka";

// let bonus = 0;

// if (role === "Senior") {

//   bonus = wage \* 2.2;

// } else if (role === "Medium") {

//   bonus = wage \* 1.6;

// } else if (role === "Junior") {

//   bonus = wage \* 1.2;

// } else {

//   bonus = wage \* 1.1;

// }

// console.log(`A ${role} makes ${bonus} every month!`);

// function rezultat (r,p) {

//     let povrsina = r\*p;

//     console.log (povrsina, 'povrsina');

// }

// rezultat (2, 0.30);