**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 test= null;

console.log (typeof test); //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");

}

Ternarni izraz umesto IF ELSE:

var broj = 5;

var rezultat = (broj > 0) ? "Pozitivan" : "Negativan";

console.log(rezultat);

**Vazno**

parseInt – pretvara string broj

let unos = prompt ("Unesi broj dana u nedelji");

let dan = parseInt(unos);

Zagrade [] – za nizove vrednosti

* Console.log – izbacuje vrednost iz varijable ZNAK na mestu koji je odredjen ostatkom u zagradi.

if (!isNaN(godina)) {

    var znak = ["Rat", "Ox", "Tiger", "Rabbit", "Dragon", "Snake", "Horse", "Goat", "Monkey", "Rooster", "Dog", "Pig"];

    var ostatak = (unos - 4) % 12;

    console.log (znak[ostatak]);}

    else {

        console.log ("Nista");

    }

if ( !**isNaN**(unos) – kako se postavlja uslob za NaN

if (!isNaN(godina))

.toLowerCase()

let plata = 100;

let dohodak = 0;

let iznos = prompt ("Unesi ime pozicije:");

let unos = iznos.toLowerCase();

Primer:

// function rezultat (r,p) {

//     let povrsina = r\*p;

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

// }

// rezultat (2, 0.30);

.toString – pretvara unos u string

let number = 1425;

let numberString = number.toString();

for (let i = 0; i < numberString.length; i++) {

    console.log(numberString[i]); }

Zadatak : Ispisati ime kineskog znaka u horoskopu. Postoji formula kako se izracunava i broj znakova je 12 i svaki zauzima svoje mesto.

var unos = prompt ("Unesi godinu rodjenja:");

var godina = parseInt(unos);

if (!isNaN(godina)) {

    var znak = ["Rat", "Ox", "Tiger", "Rabbit", "Dragon", "Snake", "Horse", "Goat", "Monkey", "Rooster", "Dog", "Pig"];

    var znakindex = (unos - 4) % 12;

    console.log (znak[znakindex]);}

    else {

        console.log ("Nista");

    }

**CAS 3**

* **Switch**

Može se koristiti kada imamo više slučajeva koji se mogu odnositi na jednu promenljivu.

* let unos = prompt ("Unesi broj dana u nedelji");
* let dan = parseInt(unos);
* switch (dan) {
* case 1 :
* console.log ("Monday");
* break;
* case 2 :
* console.log ("Thusday");
* break;
* case 3 :
* console.log ("Wednesday");
* break;
* case 4 :
* console.log ("thursday");
* break;
* case 5 :
* console.log ("Friday");
* break;
* case 6 :
* console.log ("Sathurday");
* break;
* case 7 :
* console.log ("Sunday");
* break;
* default:
* console.log ("Unesi pravilan unos")
* break;
* }

**FUNCTIONS**

function mojafunkcija (x, y) { - TELO FUNKCIJE

    let mnozenje = x \* y;

    let sabiranje = x + y;

    console.log (mnozenje, 'Ovo je mnozenje');

    console.log (sabiranje, 'Ovo je sabiranje');

}

mojafunkcija (2, 2); - POZIVANJE FUNKCIJE

1.zadatak (izracunavanje povrsine kvadrata)

function krug (raduis) {

   let povrsina = Math.PI \* Math.pow (radius, 2);

   return povrsina;

}

 let radius = prompt("Upisi radius");

radius = parseInt(radius);

if (isNaN(radius) || radius <= 0) {

    console.log ("unesi pravilan unos")

} else {

    let kraj = krug(radius);

       console.log (kraj);

}

Matematicke operacije : ( Math.PI ) ( Math.pow ) (Math.floor)

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/PI>

- RETURNING VALUESvraca vrednost. Tj. Drugo varijabli mozemo dodeliti vrednost prve varijable ako je u prvoj varijabli stavljen RETURN. **Pogledati primer sa povrsinom kruga.**

function myFunction(x, y) {

let myVar = x \* y;

console.log(myVar);

return myVar;

}

myFunction(1,3);

myFunction(2,5);

myFunction(100,232131);

let myresult = myFunction(7,8);

**HOW FUNCTIONS WORK**

A sandwich and a math equation

Description automatically generated with medium confidence

MISMATCH – ne podudaraju se parametri

  function calculateLoan   ( amount, months,interest,name ) {

    //lots and lots of code

    console.log(name, amount, months, interest);

  }

calculateLoan(1000, 12, 7, "John Doe");  //correct

calculateLoan(1000, 12, 7, "John Doe", "Something Extra");  //extras are ignored

calculateLoan(1000, 12);   //missing are passed as undefined

DEFAULT PARAMETER :

* https://www.tutorialspoint.com/javascript/javascript\_builtin\_functions.htm

function printValues(name = "Ivan", age, city) {

    console.log(name, age, city);

  }

  printValues("Marija", 24, "Belgrade");

  printValues("Marija ", 24, " Belgrade", 1998);

  printValues("Marija ", 24);

  printValues("Marija ", " Belgrade");

  printValues(1, 2, 3);

  printValues();

ARROW function

var multiply = (x, y) => {

    var tekst = "Ovo je funkcija";

    console.log(tekst)

};

* https://www.webprogramiranje.org/arrow-funkcija/

**RETURN IN FONCTIONS**

**Ako dodajemo return da bi vratili rezultat I ispisali, onda nakon tela funkcije moramo dodati novu varijablu u cijoj vrednosti pozivamo funkciju I onda u console.log dodajemo tu novu varijablju da bi se ona ispisala.**

let sredstva = 500;

function bankomat() {

    let unos = prompt("Unesi željnu sumu novca:");

    let iznos = parseInt(unos);

    if (isNaN(iznos) || iznos <= 0) {

        return "Molimo unesite validan iznos.";

    } else if (iznos > sredstva) {

        return "Nemate dovoljno para na računu.";

    } else {

        return "Biće isplaćeno " + iznos + " evra.";

    }

}

let rezultat = bankomat();

console.log(rezultat);

Takodje moguce je pozvati funkciju u console.log :

function temperatura (celzijus) {

   let farenhajt = celzijus \* 1.8 + 32;

   return (farenhajt)

}

let farenhajt = temperatura(50)

console.log (farenhajt);

….

function calculateAge (birthyear, currentyear) {

       let godina = currentyear - birthyear;

       return godina;

}

console.log (calculateAge (1995, 2024));

**SCOPE – Opseg**

* Scope (opseg ili domen) u JavaScript-u odnosi se na oblast koda gde se promenljive mogu koristiti ili gde im je vrednost vidljiva. Postoje dva osnovna tipa opsega u JavaScript-u: globalni opseg (global scope) i lokalni opseg (local scope).

Globalni opseg (Global Scope): Varijable koje su definisane van svih funkcija imaju globalni opseg. Ove promenljive mogu se koristiti bilo gde u kodu, kako u funkcijama, tako i izvan njih. One su dostupne globalno i nazivaju se globalne promenljive.

let globalnaPromenljiva = "Ja sam globalna!";

function primerFunkcije() {

console.log(globalnaPromenljiva);

}

primerFunkcije(); // Ispisuje: Ja sam globalna!

Globalna moze biti I ako se neka varijabla ne deklarise ali joj se da vrednost u sklopu funkcije. Ona automatski postaje globala.

myFunction();  
  
// code here can use carName  
  
function myFunction() {  
  carName = "Volvo";  
}

Lokalni opseg (Local Scope): Promenljive koje su definisane unutar funkcija imaju lokalni opseg. Ove promenljive su vidljive samo unutar funkcije u kojoj su definisane. Ponekad se lokalno referišu kao promenljive sa funkcionalnim opsegom (function scope).

function primerFunkcije() {

let lokalnaPromenljiva = "Ja sam lokalna!";

console.log(lokalnaPromenljiva);

}

primerFunkcije(); // Ispisuje: Ja sam lokalna!

// console.log(lokalnaPromenljiva); // Ovde bi izazvalo grešku, jer 'lokalnaPromenljiva' nije vidljiva izvan funkcije.

Važno je napomenuti da promenljive koje su definisane u lokalnom opsegu (npr. unutar funkcije) neće biti vidljive izvan tog opsega. Ali se sa istim imenom mogu koristiti u drugim funkcijama. Zato one imaju function scope.

Blokovski opseg (Block Scope): Uvodi se sa ECMAScript 6 (ES6) i predstavlja opseg koji vezuje promenljive unutar blokova koda, kao što su if, for, while, switch, i sl.

if (true) {

let x = 10; // Blokovski opseg

console.log(x); // Ispisuje: 10

}

// console.log(x); // Ovde bi izazvalo grešku, jer 'x' nije vidljiva izvan bloka if.

Koristite scope kako biste kontrolisali vidljivost i životni vek promenljivih. Ovo pomaže u izbegavanju nesporazuma i konflikata između različitih delova vašeg koda.

**ARRAY – NIZOVI**

**let niz = new Array(duzina)** – ovako se pise nova varijabla ako zelimo da uvedemo novi niz u program. To se radi kada mu rezervisemo mesto ali mu nisu date vrednosti.

* Pisu se u [ ] zagradama. Uglavnom su isti tipovi vrednosti u njima.
* let days = ['Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun'];
* ovde vidimo kako se ispisuje neka vrednost iz niza.

    var znak = ["Rat", "Ox", "Tiger", "Rabbit", "Dragon", "Snake", "Horse", "Goat", "Monkey", "Rooster", "Dog", "Pig"];

    var ostatak = (unos - 4) % 12;

    console.log (znak[ostatak]);}

* .length – sluzi za proveru duzine niza

 const days = [ 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun' ];

    console.log (" prikazi : ", days[2]);

    console.log (" duzina niza je: ", days.length);

* Zamena vrednosti u nizu:

    const days = [ 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun' ];

    days [2] = "Neki drugi dan:";

    console.log (days[2]);

* Zamena vrednosti na poslednjoj poziciji:
* days[days.length - 1] = 'Sunday';
* Dodavanje na kraju niza - .length

        const days = [ 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun' ];

        days[days.length] = 'No more days in the week'

        console.log (days);

* Dodavanje na kraju - .push

    let days = [ 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun' ];

    days.push("DODATAK");

    console.log(days);

* Dodavanje na pocetku : .unshift

    let days = [ 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun' ];

    days.unshift('DODATAK');

    console.log (days);

* Skidanje sa pocetka .shift

   let days = [ 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun' ];

    days.shift();

    console.log (days);

* Skidanje sa kraja .pop

    let days = [ 'Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun' ];

    days.pop ();

    console.log (days);

**Petlje –**

* **For Of**
* // Definiramo niz objekata
* let ljudi = [
* { ime: 'Ana', godine: 30 },
* { ime: 'Marko', godine: 25 },
* { ime: 'Petra', godine: 35 },
* { ime: 'Ivan', godine: 40 }
* ];
* // Prikazujemo osobe starije od 30 godina
* for (let osoba of ljudi) {
* if (osoba.godine > 30) {
* console.log(osoba.ime + ' ima više od 30 godina.');
* }
* }

**DOM:**

<html>

  <head>

    <title>My cool page</title>

  </head>

  <body>

    <div id="main" class="myDiv">

      <h1 id="myTitle">Yea! Cool page</h1>

      <p title="Short explanation" align="center" class="myParagraph">

        This page is really cool. Please don't change anything!

      </p>

    </div>

    <div class="myDiv">

      <p align="center" class="myParagraph second">

        No really, trust me dude!

      </p>

      <text align="center">

        See? It has random text!

      </text>

    </div>

    <script src="dom.js"></script>

  </body>

</html>

* document.getElementById(“MyTitle”)
* let pera = document.getElementById("myTitle");
* console.log(pera);
* console.log(pera.textContent);
* document.getElementByClassName(“myParagraph”)

let paragraphs = document.getElementsByClassName("myParagraph");

let secondParagraph = document.getElementsByClassName("second");

console.log(paragraphs); // Will give us an array with 2 paragraphs

console.log(paragraphs[0]); // Will give us the first paragraph

console.log(secondParagraph); // Will give us an array with 1 paragraph

console.log(secondParagraph[0]); // Will give us the only paragraph

* document.getElementByTagName("p");
* let pera = document.getElementsByTagName("p");
* console.log (pera);
* console.log (pera.textContent);
* document.querySelectorAll !!!!!!!!
* let pera1 = document.querySelectorAll("p");
* let pera2 = document.querySelector(".myParagraph");
* let pera3 = document.querySelector("#myTitle");
* .elementSibling, .nextElementSibling, .previousElementSibling.

let paragraph = document.getElementsByClassName("myParagraph")[0];

let sibling = paragraph.nextElementSibling; // We add the next sibling element here

console.log(paragraph); // The first paragraph

console.log(sibling); // The second paragraph

console.log(sibling.previousElementSibling); // Will log the first paragraph again

* .parentElement
* let paragraph = document.getElementsByClassName("myParagraph")[0];
* let parentDiv = paragraph.parentElement; // We take the parent of the paragraph
* console.log(paragraph); // The first paragraph
* console.log(parentDiv);
* .children, .firstElementChild, .lastElementChild.

let myDiv = document.getElementById("main");

let divChildren = myDiv.children; // All children of myDiv

let divFirstCh = myDiv.firstElementChild; // Get first child of myDiv

let divLastCh = myDiv.lastElementChild; // Get last child of myDiv

console.log(divChildren); // Array of all children of myDiv

console.log(divFirstCh); // The first child element of myDiv

console.log(divLastCh); // The last child element of myDiv

* .Innertext (gets or changes only the text without white spaces)
* .textContent (with white spaces)

 let header = document.getElementById("myTitle");

 console.log(header.innerText); // Checking what the text is

 header.innerText = ""; // Changing the text to nothing

 console.log(header.innerText);

 header.innerText = "New TEXT!";

 console.log(header.innerText);

 header.innerText += " Ultra new text";

 console.log(header.innerText);

* .innerHTML += <p>…..</p>

let myDiv = document.getElementById("main");

console.log(myDiv);

// Adding new element in the div

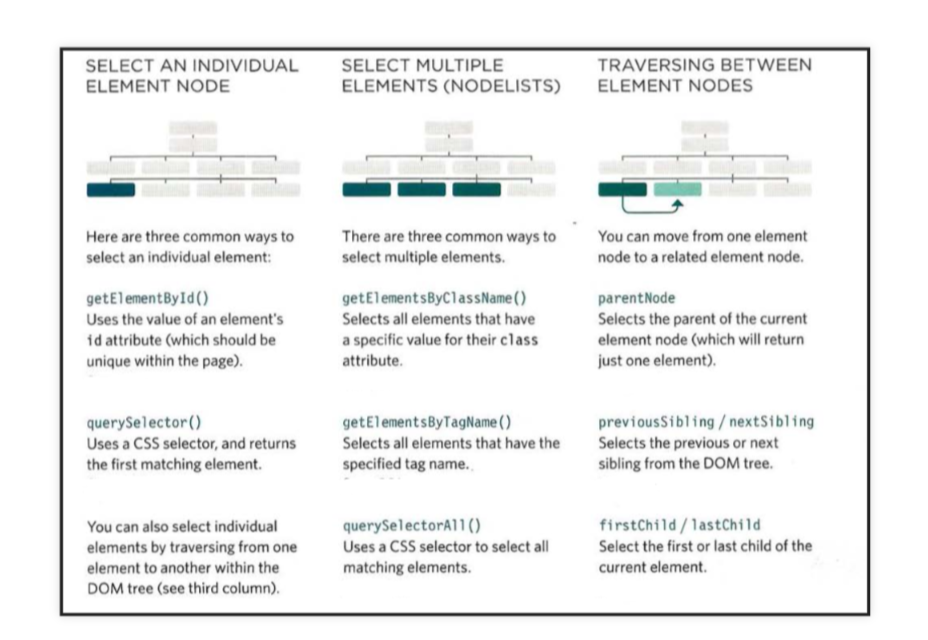
myDiv.innerHTML += `<p class="new">

Paragraph generated from JavaScript

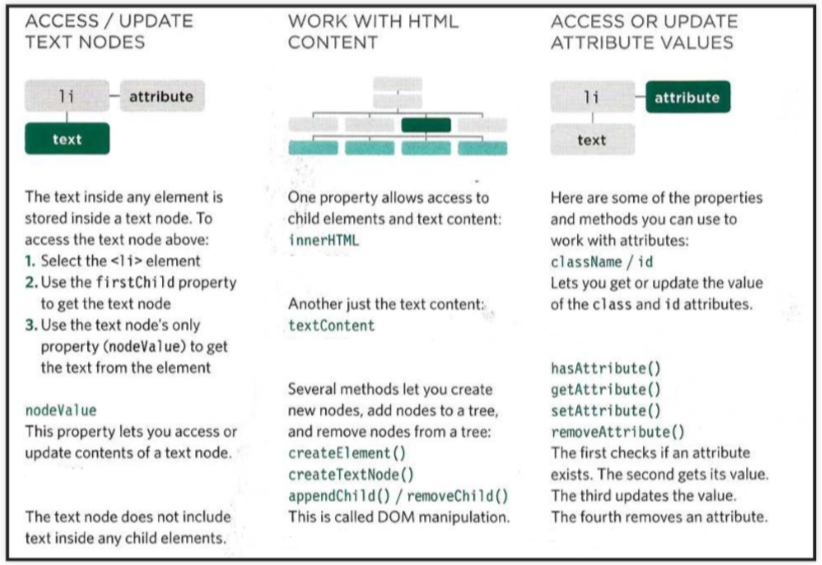
</p>`;

console.log(myDiv);

myDiv.innerHTML = ""; // removing all code from myDiv



* IZUCITI !!!!



<div id="first" class="aDiv">

  <h1 id="myTitle">Yea! Cool page</h1>

  <p class="paragraph">

    This is an exercise. It's pretty easy

  </p>

</div>

<div class="anotherDiv">

  <p class="paragraph second">

    No really, It's easy!

  </p>

  <text>

    It's about selecting elements and

  </text>

</div>

<div>

  <h1>This should be changed</h1>

  <h3>And also this!</h3>

</div>

**MANIPULATE THE DOM**

* Select the first div
* Select all paragraphs
* Select the last div
* Select the header 3 in the last div
* Select the header 1 in the last div
* Get the text from the first paragraph in the second div
* Add the word "text" to the text element in the second div
* Change the text of the header 1 in the last div
* Change the text of the header 3 in the last div

let firstDiv = document.querySelector("div");

let allParagraphs = document.getElementsByTagName("p");

let lastDiv = document.getElementsByTagName("body")[0].lastElementChild;

let header3last = lastDiv.getElementsByTagName("h3")[0];

let header1last = header3last.previousElementSibling;

let paragraphSecondText = document.getElementsByClassName("second")[0].innerText;;

document.querySelector("text").innerText += " text!";

header3last.innerText = "Text added from JS";

header1last.innerText = "Text added from JS also";

* Nauciti MODULO - <https://skolakoda.github.io/upotreba-modulo-operatora>
* Nauciti - <https://www.webprogramiranje.org/arrow-funkcija/> !!!!!!
* 01.02. – preci predavanje 5, mozda neki domaci, ako ne domaci, onda besplatan kurs.
* 01.02. –W3School I vise preci o java scriptu u besplatnim predavanjima
* getElementS ili getElement ???
* .elementSibling, .nextElementSibling, .previousElementSibling.