let number = 1425;

let numberString = number.toString();

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

    console.log(numberString[i]);

}

.toString( , ) = konvertuje numericku vrednost u string.

Math.floor

let number = 1425;

// Iteriramo kroz broj sve dok ne postane 0

while (number > 0) {

    // Izdvajamo posljednju cifru broja koristeći % 10

    let digit = number % 10;

    // Ispisujemo izdvojenu cifru

    console.log(digit);

    // Uklanjamo posljednju cifru iz broja (math.floor uklanja poslednju cifru,

    // sto znaci da u svakoj interaciji radimo sa manjim brojem)

    number = Math.floor(number / 10);

}

createString

let resultString = createString(firstName, lastName, email, password);

// Display the result in a new <p> element

document.getElementById("result").innerText = resultString;

.join - kreira string od niza

let vezba = [ime1, prezime1, sifra1, email1];

let stringNiza = vezba.join(', ');

document.write

window.open

let vezba = [ime1, prezime1, sifra1, email1];

let stringNiza = vezba.join(', ');

let noviProzor = window.open('');

noviProzor.document.write(`<p>${stringNiza}</p>`);

dugme.onclick = ispis;

Object.values da se od objekta napravi niz

<p id="demo"></p>

const person = {

  name: "John",

  age: 30,

  city: "New York"

};

document.getElementById("demo").innerHTML = Object.values(person);

.startsWith - ….

.endsWith - ako se zavrsava sa necim ?

let niz = [];

for (let i = 1; i <= 1000; i++){

    if( i.toString().endsWith("1")){

        niz.push(i);

    }

}

console.log(niz);

const email = "aida.pirusic@gmail";

if (email.endsWith(".com")) {

  console.log("valid email");

} else {

  console.log("invalid email");

}

.reset - resetuje vrednosti, obicno na kraju I za formu sluzi.

   document.getElementById("studentForm").reset();

Confirm

let book = {

    title: prompt("Unesite naslov knjige:"),

    author: prompt("Unesite ime autora:"),

    readingStatus: confirm("Da li ste već pročitali knjigu?"),

info: function() {

        if (this.readingStatus) {

            return `Već ste pročitali '${this.title}' od strane ${this.author}.`;

        } else {

            return `Još uvek treba da pročitate '${this.title}' od strane ${this.author}.`;

        }

    }

};

console.log(book.info());

Number umesto ParseInt

let userInput = prompt("Unesite broj:");

let number = parseInt(userInput);

Ili

let userInput = Number(prompt("Unesite broj:"));

Postavljanje uslova: (ako nije true ili ako nije unet, izbacice tackice)

function Student (firstName, lastName, age, email){

    this.firstName = firstName || "...";

    this.lastName = lastName || "...";

}

- myFunction.preventDefault();

let students = [];

function Student (firstName, lastName, age){

    this.firstName = firstName;

    this.lastName = lastName;

    this.age = age;

}

document.getElementById("studentForm").addEventListener("submit", function(myFunction){

myFunction.preventDefault();

    let firstName = document.getElementById("firstName").value;  
   let lastName = document.getElementById("lastName").value;  
   let age = parseInt(document.getElementById("age").value);

   let student = new Student (firstName, lastName, age);

   students.push(student);

   console.log(students);  
   document.getElementById("studentForm").reset();  
})

Uslovi

    this.firstName = firstName || "...";

   let isFormValide = true;

     if (!firstName){

        console.warn("First name not entered");

        isFormValide = false;

     }

     if(isFormValide){

        const newStudent = new Student (firstName, lastName, age, email);

        database.push(newStudent);

        console.log('database', database);

Kopiranje objekata, tj. Dupliranje bez reference

const original = new Student("Niko", "Nikic", 18); objects are copied by reference

const copy = { ...original }; losing reference !!!!!!!!!!!!!!!!!

original.displayInfo();

copy.displayInfo();

copy.years = 20;

console.log("Original years", original.years);

console.log("Copy years", copy.years);

Niz od dva objekta: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

function Student(firstName, lastName) {

//   undefined

  firstName ? firstName : "---";

  this.firstName = firstName || "---";

  this.lastName = lastName || "---";

  this.getFullName = function () {

    console.log(`${this.firstName} ${this.lastName}`);

  };

}

const students = [new Student("Aida", "P"), new Student("Miroslav", "P")];

for (let element of students) {

  element.getFullName();

}

U nizu su napravljena dva objekta iz konstruktor funkcije I interiranjem kroz njih, izbacujemo ih.

**JQuery**

.find - ugradjena funkcija

allElements.find(".wrapper").find("p").last();

.get - pravi niz od selektovanog elementa. U ovom slucaju, od svih elemenata koji imaju klasu innerWrapper, napravice niz I sadrzati ga u varijabli. Nakon toga mozemo normalno manipulisati nizom.

let byClassElements = $(".innerWrapper").get();

console.log(byClassElements);

.split

const generateBtn = document.getElementById("generateBtn");

const colorInput = document.getElementById("colorInput");

const fontSizeInput = document.getElementById("fontSizeInput");

const itemsInput = document.getElementById("itemsInput");

function onGenerate() {

  const list = document.createElement("ul");

  const itemsArray = itemsInput.value.split(","); // ['aa', 'bbb', 'ccc'];

  const color = colorInput.value;

  const fs = fontSizeInput.value;

  let isFormValid = true;

  if (!colorInput.value) {

    console.warn("Color is required");

    isFormValid = false;

  }

  if (!fontSizeInput.value || isNaN(fontSizeInput.value)) {

    console.warn("font size is required and must be a number");

    isFormValid = false;

  }

  if (!itemsInput.value) {

    console.warn("items are required");

    isFormValid = false;

  }

  if (isFormValid) {

    itemsArray.forEach(function (element) {

      const li = document.createElement("li");

      li.style.color = color;

      li.style.fontSize = `${fs}px`;

      li.innerText = element;

      list.appendChild(li);

});

document.body.appendChild(list);

    colorInput.value = "";

    fontSizeInput.value = "";

    itemsInput.value = "";

  }

}

.toFixed(2) - u ovom slucaju zaokruzuje decimale koje se pojavljuju na 2 decimale.

  let average = (num1 + num2 + num3) / 3;

    if(average >= 10){

        div.css("color", "green");

    } else {

        div.css("color", "red");

    }

    div2.html(`<h1> ${average.toFixed(2)} </h1>`);

Math.floor - zaokruzuje decimalni broj na ceo broj. Skida decimalu.

return Math.floor(sum / 3);

parseFloat() - pretvara string u broj sa decimalom ako je unesena.

let num2 = parseFloat($("#put2").val());

Arrow function

function validateEmail(email){

    return email.includes("@");

}

const validateEmail2 = (email) => {

    return email.includes("@");

}

const validateEmail3 = (email) => email.includes("@");

const myEmail = "perapera@yahoo.com";

Arrow function and USLOVI:

const countDigits = (number) => Math.abs(number).toString().length;

const isEvenOrOdd = (number) => (number % 2 === 0 ? "Even" : "Odd");

- Kako kroz poziv funkcije uraditi potrebno:

- KORISTITI **TERNARNI** PRIKAZ I **ARROW** FUNKCIJE I MANIPULISANJE **ARGUMENTIMA I PARAMETRIMA** FUNKCIJE !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

  const header = document.getElementById("h1");

const textSizeInp = document.getElementById("textSize");

const textColorInp = document.getElementById("color");

const genTextColor = (element, color) => {

  element.style.color = color ? color : "black";

};

const genTextSize = (element, textSize) => {

  element.style.fontSize = textSize ? textSize + "px" : "24 px";

};

function handleStyle() {                          //  onclick u HTML- u se desava ova funkcija.

  genTextColor(header, textColorInp.value);

  genTextSize(header, textSizeInp.value);

}

- Ubacivanje funkcije u varijablu !!!

const countDigits = (number) => Math.abs(number).toString().length;

const isEvenOrOdd = (number) => (number % 2 === 0 ? "Even" : "Odd");

const isPositiveOrNegative = (number) => {

  // 0

  if (number > 0) {

    return "Positive";

  }

  if (number < 0) {

    return "Negative";

  }

  return "Zero";

};

const getNumberStats = (number) => {

  const numDigits = countDigits(number);

const evenOrOdd = isEvenOrOdd(number);

  const posOrNeg = isPositiveOrNegative(number);

  console.log(`${numDigits} Digits, ${evenOrOdd}, ${posOrNeg}`);

};

getNumberStats(-25);

Funkcije u nizu:

const numberState = [

  (num) => (num % 2 === 0 ? "Even" : "Odd"),

  (num) => (num >= 0 ? "Positive" : "Negative"),

];

numberState[0](17);

numberState[1](20);

Funkcija bez argumenata (nisu dostupni u arrow function)

function getLongestString() {

  console.log("arguments", arguments);

  let longest = '';

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

        if(arguments[i].length >= longest.length) {

longest = arguments[i];// "Djordje"

        }

  }

  console.log('longest string is ', longest);

}

getLongestString("Milica", "Goran", "Djordje", "Sava", "Tamaraaaaaa");