- 1. da se ispise ime svih studenata.

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();

}

students.forEach(function (el) {

  console.log(el);

});

1. Create three inputs for numbers. Print the average of the three numbers in an h1 element If the average is larger or the same as 10 the result should be in green.If the average is smaller than 10 the result should be red.

    <div>

        <input type="text" id="put1">

         <input type="text" id="put2">

        <input type="text" id="put3">

        <button id="btn">Button</button>

    </div>

<div id="div2"> </div>

$(document).ready(function(){

    let button = $("#btn");

let div2 = $("#div2");

    button.click(function(){

    let num1 = parseFloat($("#put1").val());

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

    let num3 = parseFloat($("#put3").val());

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

    if(average >= 10){

        div2.css("color", "green");

    } else {

        div2.css("color", "red");

    }

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

});

});

Ovo je drugi nacin

let input = $(".input-form");

let result = $("#result")

function calculateAverage(num1, num2, num3) {

  let sum = num1 + num2 + num3;

  return Math.floor(sum / 3);

}

$("button").click(function () {

  let numOne = parseInt(input[0].value);

  let numTwo = parseInt(input[1].value);

  let numThree = parseInt(input[2].value);

  let average = calculateAverage(numOne, numTwo, numThree);

- ovde se pozivanjem funkcije daje prosek, ne moze se uzeti sum jer se ta varijabla

nalazi u drugoj funkciji, lokalnog opsega.

  if (average > 10) {

    result.text(average).css("color", "yellow");

  } else {

    result.text(average).css("color", "red");

  }

})

Ovde vidimo da mozemo uvek napraviti posebno funkciju za izracunavanje nekog rezultata I onda tu funkciju ubaciti u varijablu u daljem kodu ako nam je potrebna kroz pozivanje funkcije.

1. Create a header generator. Create two inputs, one for text and one for color. Create a button that says: generate h1. Create an h3 element for messages. When the button is clicked create a new header below the inputs and button. The new header should have the text and color from the inputs

If the person enters an invalid color or an empty text show an error message to the message element

You must use JQuery to complete the task.

$(document).ready(function(){

    let button = $("#btn");

let h3 = $("#h3");

button.click(function(){

        let text = $("#text").val();

        let color = $("#color").val();

        let isValid = true;

        if (!text || !color || !isNaN(color) || !isNaN(text)){

            h3.text("input invalid");

            isValid = false;

        }

        if (isValid){

        let newH1 = $("<h1>").text(text).css("color", color);

        button.after(newH1);

}

})

})

- Kreiranje tabele

function createTable() {

    let rows = parseInt(document.getElementById("rows").value);

    let cols = parseInt(document.getElementById("cols").value);

    let table = document.createElement("table");

    for (let i = 1; i <= rows; i++) {

      let row = table.insertRow();

      for (let j = 1; j <= cols; j++) {

        let cell = row.insertCell();

        cell.textContent = "Row-" + i + " Column-" + j;

      }

    }

    let tableContainer = document.getElementById("tableContainer");

    tableContainer.innerHTML = "";

    tableContainer.appendChild(table);}

- moze I ovako

function tabela(){

    let rows = parseInt(document.getElementById("rows").value);

let columns = parseInt(document.getElementById("columns").value);

let tabelaHTML = "<table>";

    for (let i = 1; i <= rows; i++){

        tabelaHTML += "<tr>";

        for (let j = 1; j <= columns; j++){

        tabelaHTML += "<td>Row-" + i + "Column-" + j + "</td>";

        }

        tabelaHTML += "</tr>";

    }

    tabelaHTML += "</table>";

    document.getElementById("divid").innerHTML = tabelaHTML;}

1. - Sledeci zadaci se nalaze na VIsual class3 - JS advanced.

- Create a button When the button is clicked, call the StarWars api for the first person. Print the person name in an **h1** tag. Print the person stats in a **table**: Height, Weight, Eye color, Hair color. **URL:** <https://swapi.dev/api/people/1>  
**NOTE:** JQuery will autmatically parse this call (js will not).

- Create a button When the button is clicked, get the data from a given url with an AJAX call. Print the name of the academy in an **h1** tag. Print all student names in an **unordered list**. **URL:** <https://raw.githubusercontent.com/Drakso/AJS2019/master/Class1/students.json>  
**NOTE:** You need to parse this data before using it.

- Create a button with text Get dog image When the button is clicked, call the Dog API to get random dog image. Create image and set its source as the received one from the API. Display image in the body of page. Remember to add some height and width to that image. **NOTE:** Each time button is clicked there should be created and shown new image for dog, so we got many dog images. **URL:** <https://dog.ceo/api/breeds/image/random>

1. Create 3 inputs: Color, FontSize, Text. Create a button for generating titles. When the button is clicked generate a new h1 element with the color, font size, and text from the inputs.

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 = "";

  }

}

// function onGenerate2() {

//   let list = '<ul>';

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

//   console.log(itemsArray);

//   itemsArray.forEach(function(element){

//     list += `<li>${element}</li>`;

//   });

//   list += '</ul>';

//   console.log(list);

//   document.body.innerHTML += list;

// }

// generateBtn.addEventListener("click", onGenerate);

1. Create an array of 5 names. Create an HTML page with: A header, An empty unordered list, A button. When the button is clicked it should fill in the empty unordered list with the names of the array.

let list = document.getElementById("list");

let niz = ["Milica", "Sava", "Djordje", "Zarko", "Sava"];

function newList(){

    for (let ime of niz){

        list.innerHTML += `<li> ${ime}  </li>`

    }

}

const names = ["Alice", "Bob", "Charlie", "David", "Eve"];

function fillList() {

  let list = document.getElementById("name-list");

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

    let item = document.createElement("li");

    item.textContent = names[i];

    list.appendChild(item);

  }

}

6 Create 2 variables with arrow functions. 1. First arrow function will accept two parameters, one for element and one for color. The function should change the given element text color with the color given from the second color parameter. If no parameter is passed for color, the default value is **black**. 2. Second arrow function will accept two parameters, one for element and one for textSize. The function should change the given element text size to the number given from the second textSize parameter. If no parameter is passed for textSize, the default value is 24. Create an HTML document with two inputs, a button and an h1 header with some text. The first input should be for text size and the second for color. When the button is clicked the h1 header should change according to the input values ( change size as the first input value and color as the second input value ). Use the functions that we declared earlier and use arrow function for the event listener of the button.

 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);