Zasto nema else?

const isPositiveOrNegative = (number) => {

  // 0

  if (number > 0) {

    return "Positive";

  }

  if (number < 0) {

    return "Negative";

  }

  return "Zero";

};

Pitati za cas 7 - poslednji zadatak I cas 8, linija koda 210

   this.addRating = (rating) => {

      const smallRating = ({ user = "unknown", grade, comment = "" } = rating);    //.....    mora da se stave zagrade() da bi se posle pusalo i da bi to bilo kao celina, gradi novi objekat

      this.ratings.push(smallRating);

    };

Get and set kod inheritance classes

Cim vidi jedan strign, sve brojeve pretvara u string (console.log(“3” 4)) - ispisace 34

string, number, undefined, boolean primitivni tipovi podataka

Null, false, prazan string “”, broj 0 - to su falsy vrednost;

Console.log(isNaN “test”) - vratilo bi true;

Console.log(isNan “7”) - iako je strign on prepoznaje da je broj pa izbacuje false

Member operator to je ona tacka = c.name = ddsa;

http metode???

PROVEZBATI

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);}

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

let x = "";

const myObj = {

  name: "John",

  age: 30,

  cars: [

    {name:"Ford", models:["Fiesta", "Focus", "Mustang"]},

    {name:"BMW", models:["320", "X3", "X5"]},

    {name:"Fiat", models:["500", "Panda"]}

  ]

}

for (let i in myObj.cars) {

  x += "<h2>" + myObj.cars[i].name + "</h2>";

  for (let j in myObj.cars[i].models) {

    x += myObj.cars[i].models[j] + "<br>";

  }

}

document.getElementById("demo").innerHTML = x;

Get set u objektima

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.getFullN

  <div class="wrapper">

        <h1 id="mainTitle">Hey this is my page</h1>

        <div class="innerWrapper">

          <h3 id="firstTitle">This is some subheader</h3>

          <p>Hello!</p>

          <p>Yes it is a sub header!</p>

        </div>

        <div class="innerWrapper">

          <h3 id="SecondTitle">This is some other subheader</h3>

          <p>Hello again!</p>

          <p>Yes it is another sub header!</p>

        </div>

        <button>It does nothing :</button>

        <input type="text" value="Cool">

      </div>

let allElements = $("\*");

let byIdElement = $("#mainTitle");

let byClassElements = $(".innerWrapper");

let byTagNameElements = $("p");

let wrapperAndInnerWrapper = $(".wrapper,.innerWrapper");

let firstParagraph = $("p:first");

let lastParagraph = $("p:last");

let firstChildHeaders = $("h3:first-child");

let lastChildParagraphs = $("p:last-child");

let secondChildParagraphs = $("p:nth-child(2)");

console.log(firstParagraph.text());

// selecting

allElements.first();

allElements.last();

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

allElements[0];

// Get all DOM elements from a JQuery array

byClassElements.get();

// Get next or previous element

firstParagraph.next();

firstParagraph.prev();

What are the final values of C and D into the object, when following code is executed?

const object1 = {

     a: 1,

     b: 2,

     c: 3

};

const object2 = Object.assign({c: 4, d: 5}, object1);

console.log(object2.c, object2.d);

C:3, D:5

C:4, D:5

Both of the above

None of above

What is the difference between display: none and visibility: hidden?

They do the same thing

display:none hides the element completely, visibility: hidden hides the element, but leaves empty space on the spot

visibility: hidden hides the element completely, display:none hides the element, but leaves empty space on the spot

None of the above

You are developing a customer web form that includes the following HTML:

<input id = "txtValue" />

A customer must enter a value in the text box prior to submitting the form.

You need to add validation to the text box control.

Which HTML should you use?

<input id="txtValue" type="text" required="required" />

<input id="txtValue" type="text" pattern="[A-Za-z]{3}" />

<input id="txtValue" type="required" />

<input id="txtValue" type="required" autocomplete="on" />

What will be the value of the result variable and of what type is it?

let names = ["bob", "jill", "greg", "john", "anne"];

let result = names

     .filter(x => x[0] === x[x.length-1])

    .reduce((res, cur) => res + " " + cur, "[") + " ]";

[ bob, greg ] of type array

[ bob greg ] of type array

[ bob, greg ] of type string

[ bob greg ] of type string