**Introduction and Code Quality**

**(1) Write a program to Show an alert**

**A.** document.write(alert(“Any message you can put here”));

**(2) What will be the result for these expressions?**

1. 5 > 4

A. True

2. "apple" > "pineapple"

A. False

3. "2" > "12"

A. True

4. undefined == null

A. True

5. undefined === null

A. False

6. null == "\n0\n"

A. False

7. null === +"\n0\n”

A. False

**(3) Will alert be shown?**

**if ("0") { alert( 'Hello'); }**

**A.** No. if we put 0 in condition it is work as an empty string and code under this condition would never execute.

**(4) What is the code below going to output?**

**alert( null || 2 || undefined );**

1. Result will be 2. Because null and undefined refers to empty string.

**(5) The following function returns true if the parameter age is greater than 18. Otherwise it asks for a confirmation and returns its result:**

1. **function checkAge(age) {// making function to check age**

**if (age> 18)**

**{return true;}**

**Else**

**{return confirm (‘did parents allow you?');} // open confirm box**

**}**

**(6) Replace Function Expressions with arrow functions in the code below:**

**Function ask(question, yes, no) {**

**if (confirm(question))yes();**

**else no();**

**}**

**ask("Do you agree?",**

**function() { alert("You agreed."); },**

**function() { alert("You canceled the execution."); }}**

1. const ask = (question, yes, no) => {

if (confirm(question)) yes ();

else no ();

};

Ask (

"Do you agree?",

() => alert ("You agreed."),

() => alert ("You canceled the execution.")

);

**Data Types and Objects**

**(1) Write the code, one line for each action:**

**(a) Create an empty object user.**

**A.** let person = new object (); // Create an object

**(b) Add the property name with the value John.**

**A.** person.name = “john”; // make property called value ‘john’

**(c) Add the property surname with the value Smith.**

**A.** person.surname = “smith”; // make property called assign value ‘smith’

**(d) Change the value of the name to Pete.**

**A.** person.name = “pete”; // change value of property assign value ‘pete’

**(e) Remove the property name from the object.**

**A.** delete person.name; // delete property called name

**(2) Is array copied?**

**let fruits = ["Apples", "Pear", "Orange"];**

**// push a new value into the "copy"**

**let shoppingCart = fruits;**

**shoppingCart.push("Banana");**

**// what's in fruits?**

**alert( fruits.length ); ?**

1. yes array is copied

Fruits = [“apples”, “pear”, “orange”, “banana”]

The fruits length will be 4.

**(3) Map to names**

**let john = {name: "John", age: 25};**

**let pete = {name: "Pete", age: 30};**

**let mary = {name: "Mary", age: 28};**

**let users = [john, pete, mary];**

**let names = /\* ... your code \*/ alert( names ); // John, Pete, Mary**

**A.** let john = {name: "John", age: 25};

let Pete = {name: "Pete", age: 30};

let Mary = {name: "Mary", age: 28};

let users = [john, Pete, Mary];

let names = users .map((item => {item.name})); // carrying out name value from objects.

Alert(names);

**(4) Map to objects**

**let john = {name: "John", surname: "Smith", id: 1};**

**let Pete = {name: "Pete", surname: "Hunt", id: 2};**

**let Mary = {name: "Mary", surname: "Key", id: 3};**

**let users = [ john, Pete, Mary];**

**let usersMapped = /\* ... your code ... \*/ /\***

**usersMapped = [ {fullName: "John Smith", id: 1},**

**{fullName: "Pete Hunt", id: 2},**

**{fullName: "Mary Key", id: 3} ]**

**\*/ alert( usersMapped[0].id ) // 1 alert( usersMapped[0].fullName ) // John Smith**

**A.** let userMapped = users.map((user) => ({

Fullname: `${user.name} ${user.surname}`, id: user.id}));

**(5) Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries(salaries) that returns the sum of all salaries using Object.values and the for..of loop.If salaries is empty, then the result must be 0.**

**let salaries = {"John": 100, "Pete": 300, "Mary": 250};**

**alert(sumSalaries(salaries)); // 650**

1. let salaries = {“john”:100, “pete”:300, “mary”:250};

Function sumSalaries(salaries){

Let sum = 0; //made sure that sum will be 0

For (const salary of object.values(salaries)){

Sum+=salary; //addition of salaries

}

Return sum; //Output as sum

}

alert(sumSalaries(salaries));

**(6) Destructuring assignment We have an object: Write the Destructuring assignment that reads:**

**let user = {name: "John", years: 30};**

**(a) Name property into the variable name.**

**A. l**et {name} = user; // Extracting the 'name' property

**(b) Year’s property into the variable age.**

**A.** let {years: age} = user; // Extracting the 'years' property and assign it to 'age'

**(c) isAdmin property into the variable isAdmin (false, if no such property)**

**A.** let {isAdmin = false} = user; // Extracting the 'isAdmin' property with a default value of 'false'

**(7) Turn the object into JSON and back Turn the user into JSON and then read it back into another variable.**

**user = {name: "John Smith", age: 35};**

1. let object = JSON.parse(user); //JSON.parse converts object into json

let json = JSON.stringify(object); //JSON.stringyfy coverts json into object

**Document, Event and Controls**

**(1) Create a program to hide/show the password**

**A.** <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    PASSWORD: <input type="password" placeholder="Password" value="FAKEpsw" id="myInput">

    <input type="checkbox" onclick="togglePassword()">Show Password

    <script>

       function  togglePassword() {

            var x = document.getElementById('myInput');

            if (x.type === "password") {

                x.type = "text";

            }

            else {

                x.type = "password";

            }

        }

    </script>

</body>

</html>

**(2) Create a program that will select all the classes and loop over and whenever i click the button the alert should show**

**New Request**

**(1) What is JSON**

1. JavaScript Object Notation is standard file format used to exchange data internally. Object data are stored and transmitted using key-value pair and array data types. JSON format is text only, which makes easy to read and use with any programming language.

**(2) What is promises.**

**A.** JavaScript Promise Object combination of both the producing code and calls to the inner code

Promise Object can be:

Pending

Fulfilled

Rejected

myPromise.then{

function(value){/\*code if successful\*/},

function(error){/\*code if failed\*/}

};

**(3) Write a program of promises and handle that promises also**

**A.**