**ASSIGNMENT-6**

**Advance Javascript**

**MODULE: 1(Introduction and Code Quality)**

**• Write a program to Show an alert**

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

• **What will be the result for these expressions?**

1. 5>4

Ans. True

2. “apple” > “pineapple”

Ans. False

3. “2” > “12”

Ans. True

4. undefined == null;

Ans. True

5. undefined === null;

Ans. False

6. null == “\n0\n”

Ans. False

7. null === “\n0\n”

Ans. False

**• Will alert be shown? If(“0”) {alert(‘Hello’};**

Ans. No, if we put 0 in condition it is work as an empty string and code under

this condition would never execute.

**• What is the code below going to output? alert( null || 2 || undefined );**

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

**• The following function returns true if the parameter age is greater than 18.**

Otherwise it asks for a confirmation and returns its result:

Ans. function checkAge(age) { //making function to check age

if (age> 18)

{ return true; }

Else

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

}

**• 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."); } }

Ans.

Function ask(question,yes,no)

{

if (confirm(question)) yes();

else no();

}

Ask(“do you agree?”,

()=>{alert("You agreed."); }, //Arrow function

()=>{alert("You canceled the execution."); }) //Arrow function

**MODULE: 2(Data Types and Objects)**

**• Write the code, one line for each action:**

o a) Create an empty object user.

Ans. let person = new Object(); //Create an object

o b) Add the property name with the value John.

Ans. person.name = “John”; //Make property called name and assign

value ‘John’

o c) Add the property surname with the value Smith.

Ans. person.surname = “Smith”; //Make property called surname and

assign value ‘Smith’

o d) Change the value of the name to Pete.

Ans. person.name = “Pete”; //change value of property called name and

assign value ‘Pete’

o e) Remove the property name from the object.

Ans. delete person.name; //Delete property called name

**• 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 ); // ?

Ans. Yes array is copied

Fruits = ["Apples", "Pear", "Orange","Banana"] //because both variable assigns

to the same reference

The fruits length will be 4.

**• 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**

Ans.

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 );

**• 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

Ans.

let usersMapped = users.map((user) => ({

fullname: `${user.name} ${user.surname}`,

id: user.id

}));

**• 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**

Ans. let salaries = {

"John": 100,

"Pete": 300,

"Mary": 250,

};

function sumSalaries(salaries) {

var 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) );

**• Destructuring assignment We have an object: Write the Destructuring**

**assignment that reads:**

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

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

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

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

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

let { name } = user; // Extracting the 'name' property

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

let { isAdmin = false } = user; // Extracting the 'isAdmin' property with a default

value of 'false'

**• 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};**

Ans.

//JSON.parse converts object into json

let object = JSON.parse(user);

//JSON.stringyfy coverts json into object

let json = JSON.stringify(object);