Advance JavaScript

MODULE: 1 (Introduction and Code Quality)

(1) Write a program to Show an alert.

Ans -

```
<!DOCTYPE html>
<html>
<body>
<button onclick="myFunction()">Click here</button>

<script>
function myFunction() {
  alert("Hello! I am an alert box!");
}
</script>

</body>
</html>
```

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

1. 5 > 4
2. "apple" > "pineapple"
3. "2" > "12"
4. undefined == null
5. undefined === null
6. null == "\n0\n"
7. 7. null === +"\n0\n"

Ans -

1. true
2. false
3. true
4. true
5. false
6. false
7. false

(3) Will alert be shown?

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

Ans -

Hello

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

alert(null || 2 || undefined);

Ans -

2

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

```
function
checkAge(age)
{
if (age> 18) { return true; }
else {
// ...return confirm (‘did parents allow you?');
 }
 }
```

Ans -

```
function checkAge(age) {
      return age > 18 ? true : confirm("Did parents allow you?");
    }

    // console.log(checkAge(age));
```

(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."); }
 )
```

Ans -

```
function ask(question, yes, no) {
  if (confirm(question)) yes();
  else no();
}

ask(
  "Do you agree?",
  () => alert("You agreed."),
  () => alert("You canceled the execution.")
);
```

MODULE: 2 (Data Types and Objects)

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

a) Create an empty object user.

b) Add the property name with the value John.

c) Add the property surname with the value Smith.

d) Change the value of the name to Pete.

e) Remove the property name from the object.

Ans -

```
(a) const obj = {};
    console.log(obj);

(b) const person = {name: "John"};

(c) const person = { surname: "Smith" };

(d) const person = { name: "Smith" };
    person.name = "Pete";

(e) const person = { name: "Smith" };
    delete person.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); // ?

Ans -

(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

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);
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**

Ans -

```
let usersMapped = 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
```

Ans -

```
function sumSalaries(salaries) {
let sum = 0;
for (let salary of Object.values(salaries)) {
sum += salary;
  }
return sum; // 650
}

let salaries = {
  "John": 100,
  "Pete": 300,
  "Mary": 250
};
alert( sumSalaries(salaries) );
```

(6) 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, years: age, isAdmin = false} = user;

alert( name ); // John
alert( age ); // 30
alert( isAdmin );
```

(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};

Ans -

```
user = { name: "John Smith", age: 35 };
let json = JSON.stringify(user);
alert(json);  //it's show that {"name":"John Smith","age":35}
```

MODULE: 3 (Document, Event and Controls)

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

Ans -

```
<!DOCTYPE html>
<html>
 <body>
  Password: <input type="password" value="FakePSW" id="myInput" /><br /><br />
  <input type="checkbox" onclick="myFunction()" />Show Password

  <script>
   function myFunction() {
    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.

Ans -

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src=
       "https://code.jquery.com/jquery-3.6.0.min.js"
       integrity=
"sha256-/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4="
       crossorigin="anonymous">
    </script>
</head>

<body>
    <button id="btn">Click me!</button>

    <script>
       $(document).ready(function () {
          $("#btn").click(function () {
             alert("This is an alert message!");
          });
       });
    </script>
</body>
  </html>
```

(3) Create a responsive header using proper JavaScript.

Ans - Solution available in Responsive header folder.

(4) Create a form and validate using JavaScript.

Ans - Solution available in Form and Validate folder.

(5) Create a modal box using css and Js with three buttons.

Ans - Solution available in Model Box folder.

(6) Use external js library to show slider.

Ans - Solution available in Slider folder.

(7) Prevent the browser when i click the form submit button.

Ans - Solution available in Submit button folder.

MODULE: 4 (New Request)

(1) What is JSON?

Ans -

JSON stands for JavaScript Object Notation.
JSON is a lightweight format for storing and transporting data.
JSON is often used when data is sent from a server to a web page.
JSON is "self-describing" and easy to understand.

(2) What is promises?

Ans -

A promise is an object that may produce a single value some time in the future: either a resolved value, or a reason that it’s not resolved (e.g., a network error occurred). A promise may be in one of 3 possible states: fulfilled, rejected, or pending. Promise users can attach callbacks to handle the fulfilled value or the reason for rejection.

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

Ans -

```
myPromise.then(

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

function(error) { /* code if some error */ }

);
```

(4) Use fetch method for calling an api https://fakestoreapi.com/products.

Ans -

```
<script>
    fetch("https://fakestoreapi.com/products")
      .then(
        // view the  complete content of the response
        (response) => response.json()
      )
      .then(
        //access the actual data
        (data) => console.log(data)
      );
  </script>
```