MODULE-1:

(Introduction and Code Quality)

1) Write a program to Show an alert

Ans.

```
<html>
<head>
<script type = "text/javascript">
function fun() {
alert ("This is an alert dialog box");
}
</script>
</head>

<body>
<p> Click the following button to see the effect </p>
<form>
<input type = "button" value = "Click me" onclick = "fun();" />
</form>
</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. null === +"\n0\n"

Ans

1. 5 > 4 → true

2. "apple" > "pineapple" → false

3. "2" > "12" → true

4. undefined == null → true

5. undefined === null → false

6. null == "\n0\n" → false

7. null === +"\n0\n" → false

Some of the reasons:

Obviously, true.

Dictionary comparison, hence false. "a" is smaller than "p".

Again, dictionary comparison, first char "2" is greater than the first char "1".

Values null and undefined equal each other only.

Strict equality is strict. Different types from both sides lead to false.

Similar to (4), null only equals undefined.

Strict equality of different types.

3 Will alert be shown? if ("0") { alert('Hello'); }

Ans

<script>

alert ("Hello");

</script>

4What is the code below going to output? alert(null || 2 || undefined);

Ans

-> alert(null || 2 || undefined);

The answer is 2, that’s the first truthy value.

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

This function is called is Old Enough To Vote (age) and has the following specifications: It takes an argument called age representing the age of the person. It checks if the age is greater than or equal to 18. If returns true or false based on that comparison.

```
let response;
var age = 18
// Add your code here
function is Old Enough To Vote(age)
{
 if (age >= 18)
{
   result; 'true'
 }
Else
{
   result; 'false'
 }
```

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 cancelled the execution.");

}

}

Ans

<!DOCTYPE html>

<script>

"use strict";

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

Ask (
  "Do you agree?",
  () => alert ("You agreed."),
  () => alert ("You cancelled the execution.")
);
</script>
```

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

Let user = {};

User .name = “John”;

User. surname = “smith”;

User.name = “pete”;

Delete User.name;

2. Is array copied?

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

// push a new value into the "copy" let shopping Cart = fruits; shopping Cart. push("Banana");

// what's in fruits? Alert (fruits .length); // ?

Ans

The result is 4:

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

let shopping Cart = fruits;

shopping Cart. push("Banana");

alert (fruits. length); // 4

That’s because arrays are objects. So both shopping Cart and fruits are the references to the same array.

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 ); // John, Pete, Mary
```

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 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 =  users.map
(user =>
({ fullName: `${user.name} $ {user.surname}` ,
id: user.id
}));
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
```

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( sum Salaries (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 (sum Salaries (salaries)); // 650

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) is Admin property into the variable is Admin (false, if no such property)

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

Ans

Let user = {

Name: "john",

Years: 30

};

Let {name, years: age, is Admin = false} = user;

Alert(name); // John

Alert(age); //30

Alert (is Admin); //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};

Ans

Let user =

{

name: "John Smith",

age: 35

};

Lets user2 = JSON.parse(JSON.stringify(user));

MODULE: 3 (Document, Event and Controls)

1. Create a program to hide/show the password

Ans

```
<!DOCTYPE html>
<html>
<body>

 <b><p>Click on the checkbox to show
        or hide password: </p></b>

 <b>Password</b>: <input type="password"
    value="geeksforgeeks" id="typepass">
```

```
 <input type="checkbox" onclick="Toggle()">
   <b>Show Password</b>


   <script>
   // Change the type of input to password or text
      function Toggle() {
         var temp = document.getElementById("typepass");
         if (temp.type === "password") {
            temp.type = "text";
         }
         else {
            temp.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

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
```

```
  <button type="button">Please Press Me</button>
</body>
<script>
For ( let i=0;i<10;i++)
{
        Document.write(“Hello World! <br>”);
}
  var pressed Button = document. Get Elements By Tag Name("button")[0];
  pressed Button.add Event Listener("click", function (event)
{
    alert ("You have pressed the button.....")
  })
</script>
</html>
```

3.Create a responsive header using proper JavaScript

Ans

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
HTML: <header>
 <div class="header-inner">
   <h2 class="logo">LO<span>GO</span></h2>
```

```
    <i id="bars" class="fas fa-bars bars"></i>
    <nav class="nav-menu">
      <a href="#" class="nav-link">Home</a>
      <a href="#" class="nav-link">About</a>
      <a href="#" class="nav-link">Services</a>
      <a href="#" class="nav-link">Contact</a>
    </nav>
  </div>
</header><nav id="mobileMenu" class="mobile-menu">
  <a href="#" class="nav-link">Home</a>
  <a href="#" class="nav-link">About</a>
  <a href="#" class="nav-link">Services</a>
  <a href="#" class="nav-link">Contact</a>
</nav>


CSS: *{
  padding: 0;
  margin: 0;
  box-sizing: border-box;
}body{
  height: 1000px;
}header{
  position: relative;
  z-index: 2;
  background-color: #333;
}.header-inner{
  width: 80%;
  margin: 0 auto;
  padding: 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}.logo{
  color: #fff;
  cursor: default;
}.logo span{
  color: #FFFF00;
}.bars{
  font-size: 26px;
```

```
  color: #fff;
  display: none;
  cursor: pointer;
  transition: color 0.6s ease;
}.bars:hover{
  color: #FFFF00;
}.nav-link{
  margin-left: 30px;
  color: #fff;
  text-decoration: none;
  transition: color 0.6s ease;
}.nav-link:hover{
  color: #FFFF00;
}.mobile-menu{
  position: absolute;
  top: 0;
  left: -100%;
  width: 100%;
  height: 100%;
  z-index: 1;
  background-color: #222;
  opacity: 0;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  margin-top: 30px;
  pointer-events: none;
}@media screen and (max-width: 768px){
  .bars{
    display: block;
  }

  .nav-menu{
    display: none;
  }

  .active{
  left: 0;
  opacity: 1;
```

```
  pointer-events: auto;
  transition: left 0.6s ease-in-out;
  }

  .nav-link{
    font-size: 24px;
    margin: 30px 0;
  }
}
```

```
</body>
```

Javascript:

```
<script>
 const bars = document.getElementById('bars');
const mobileMenu =
document.getElementById('mobileMenu');bars.addEventListener('click', function()
{
  mobileMenu.classList.toggle('active')
})
</script>
</html>
```

● Create a form and validate using JavaScript

Ans

```
<script>
function validateform(){
var name=document.myform.name.value;
var password=document.myform.password.value;

if (name==null || name==""){
  alert("Name can't be blank");
```

```
  return false;
}else if(password.length<6){
  alert("Password must be at least 6 characters long.");
  return false;
  }
}
</script>
<body>
<form name="myform" method="post" action="abc.jsp" onsubmit="return validateform()" >
Name: <input type="text" name="name"><br/>
Password: <input type="password" name="password"><br/>
<input type="submit" value="register">
</form>
```

● Create a modal box using css and Js with three buttons

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
```

```
  padding-top: 100px; /* Location of the box */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: rgb(0,0,0); /* Fallback color */
  background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
}

/* Modal Content */
.modal-content {
  background-color: #fefefe;
  margin: auto;
  padding: 20px;
  border: 1px solid #888;
  width: 80%;
}

/* The Close Button */
.close {
  color: #aaaaaa;
  float: right;
  font-size: 28px;
  font-weight: bold;
}
```

```
.close:hover,
.close:focus {
  color: #000;
  text-decoration: none;
  cursor: pointer;
}
</style>
</head>
<body>

<h2>Modal Example</h2>

<!-- Trigger/Open The Modal -->
<button id="myBtn">Open Modal</button>

<!-- The Modal -->
<div id="myModal" class="modal">

  <!-- Modal content -->
  <div class="modal-content">
    <span class="close">&times;</span>
    <p>Some text in the Modal..</p>
  </div>

</div>

<script>
```

```
// Get the modal
var modal = document.getElementById("myModal");

// Get the button that opens the modal
var btn = document.getElementById("myBtn");

// Get the <span> element that closes the modal
var span = document.getElementsByClassName("close")[0];

// When the user clicks the button, open the modal
btn.onclick = function() {
  modal.style.display = "block";
}

// When the user clicks on <span> (x), close the modal
span.onclick = function() {
  modal.style.display = "none";
}

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>
```

```
</body>
</html>
```

● Use external js library to show slider

Ans

Using Javascript library to add a slider This is (according to me) the best way to add a Image/Normal Slider with good Animations in your website. In this we will use a JS library called SwiperJS.

```
<img src="images/1.jpg" name="slide"  width="100%" height="368" />
<script>
<!--
   var image1=new Image()
   image1.src="images/1.jpg"
   var image2=new Image()
   image2.src="images/4.jpg"
   var image3=new Image()
   image3.src="images/3.jpg"

   //variable that will increment through the images
   var step=1
   function slideit(){
      //if browser does not support the image object, exit.
      if (!document.images)
         return
      document.images.slide.src=eval("image"+step+".src")
      if (step<3)
         step++
```

```
        else
            step=1
        //call function "slideit()" every 2.5 seconds
        setTimeout("slideit()",2500)
    }
    slideit()
//-->
</script>
```

● Prevent the browser when i click the form submit button

Ans

An event listener can be used to prevent form submission. It is added to the submit button, which will execute the function and prevent a form from submission when clicked. For example, element.addEventListener("submit", function(event) { event.preventDefault(); window.history.back(); }, true);

```
$j('form#userForm .button').click(function(e) {
    if ($j("#zip_field").val() > 1000){
        $j('form#userForm .button').attr('onclick','').unbind('click');
        alert('Sorry we leveren alleen inomstreken hijen!');
        e.preventDefault();
        return false;
    }
});
<button class="button vm-button-correct" type="submit"
 onclick="javascript:return myValidator(userForm,
'savecartuser');">Opslaan</button>
```

MODULE: 4 (New Request)

● 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

● What is promises?

"Producing code" is code that can take some time

"Consuming code" is code that must wait for the result

A Promise is a JavaScript object that links producing code and consuming code

● Write a program of promises and handle that promises also

Ans

```
var promise = new Promise(function(resolve, reject) {
  const x = "jainikforjainik";
  const y = "jainikforjainik"
  if(x === y) {
    resolve();
  } else {
    reject();
  }
});


promise.
    then(function () {
```

```
    console.log('Success, You are a JAINIK);
}).
catch(function () {
    console.log('Some error has occurred');
});
```

● Use fetch method for calling an api https://fakestoreapi.com/products

Ans

fakeStoreApi can be used with any type of shopping project that needs products, carts, and users in JSON format. you can use examples below to check how fakeStoreApi works

```
fetch('https://fakestoreapi.com/products')
    .then(res=>res.json())
    .then(json=>console.log(json))
```

output:

```
[
{
Id:1,
Title:'….',
Price:'….',
Category:'….',
Description:'….',
Image:'….'
}
/*…..*/
{
Id:30,
Title:'….',
```

Price:’….’,

Category:’….’,

Description:’….’,

Image:’….’

}

]

● Display all the product from the api in your HTML page

Ans

How to display api**: .**

```
<script>
    function fetchdata() {
    $.get("http://10.10.35.138:5000/data", function (data) { //The link of this line is my api link
            $("#visitor").html('Visitor Count : ' + data.people);
            $("#time").html('Time : ' + data.time);
        });
      }
</script>
```

****HTML PART****

```
<div class="details">
<p id="visitor">Person Count:</p>
<p id="time">Time:</p>`enter code here`
</div>
```

Display API Data in Html:

```
<!DOCTYPE html>
```

```
<html>
<body>

<h1>API Data</h1>

<div id="container">
    <div id="api">Nothing Yet</div>
</div>

<br>

<button type="button" onclick="loadAPI()">Change Content</button>

<script>
function loadAPI() {
  var xhttp = new XMLHttpRequest();
  xhttp.open("GET", "API URL with Token here", false);
  xhttp.addEventListener("load", loadData);
  xhttp.send();
}

function loadData() {
  document.getElementById('api').innerText = JSON.parse(this.responseText);
}
</script>

</body>
```

</html>