**JavaScript Essentials And Advanced**

**Advance JavaScript**

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

1. **Write a program to Show an alert**

* <!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>

Name : <input type="text" id="name">

<br><br>

<input type="submit" onclick="validate()">

<script>

name\_tag=document.getElementById("name");

console.log(name\_tag);

function validate()

{

if(name\_tag.value=="")

{

alert("name is required......!");

}

</script>

</body>

</html>

**2.** **What will be the result for these expressions?**

**1. 5 >4**

* True. This expression compares the values 5 and 4 and checks if 5 is greater than 4, which is true.

**2."apple" > "pineapple"**

* False. In JavaScript, string comparison is based on lexicographical (dictionary) order. The expression is false.

**3."2" > "12"**

* True : The character "2" is greater than "1", so the expression is true.

**4. undefined == null**

* True. The equality operator (==) in JavaScript performs type coercion. In this case, both undefined and null are considered equal when using the equality operator, so the expression is true.

**5. undefined === null**

* False. The strict equality operator (===) in JavaScript does not perform type coercion. undefined and null are of different types, so the expression is false.

**6. null == "\n0\n"**

* False. The equality operator (==) in JavaScript does type coercion. When comparing null to a string, null is only considered equal to null or undefined. Therefore, the expression is false.

**7. null === +"\n0\n"**

* False. The strict equality operator (===) in JavaScript does not perform type coercion. The expression compares a number (7) with the result of converting the string "\n0\n" to a number using the unary plus operator. Since null and 7 are of different types, the expression is false.

**3.Will alert be shown?**

**if ("0")**

**{ alert( 'Hello'); }**

* This equation does not compare in if

Ex:

if(name.value==”0”)

{alert(‘hello’)}

**4.What is the code below going to output?**

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

* output: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?');**

**}**

**}**

* <!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>

<script>

function checkAge(age) {

if (age > 18) {

return true;

}

else {

return confirm('Did parents allow you?');

}

}

</script>

</body>

</html>

But answer is 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 canceled the execution."); }**

**}**

* <!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>

<script>

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

if (confirm(question)) {

yes();

} else {

no();

}

};

ask(

"Do you agree?",

() => {

alert("You agreed.");

},

() => {

alert("You canceled the execution.");

}

);

</script>

</body>

</html>

**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.**

* + a) const user = {};

b) user.name = "John";

c) user.surname = "Smith";

d) user.name = "Pete";

e) delete user.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 ); // ?**

* + ["Apples", "Pear", "Orange" , ”Banana”];
  + Fruits.length: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**

* + 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(user => user.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**

* 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

}));

alert(usersMapped[0].id); // 1

alert(usersMapped[0].fullName); // John Smith

**Document, Event and Controls**

1. **Create a program to hide/show the password**

<input type="password" id="passwordField">

<button onclick="togglePasswordVisibility()">Toggle Password</button>

<script>

function togglePasswordVisibility() {

var passwordField = document.getElementById("passwordField");

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

passwordField.type = "text";

} else {

passwordField.type = "password";

}

}

</script>

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

* <button id="alertButton">Show Alert</button>

<script>

document.getElementById("alertButton").addEventListener("click", function() {

var elements = document.getElementsByClassName("your-class-name");

for (var i = 0; i < elements.length; i++) {

alert(elements[i].textContent);

}

});

</script>

**3.Create a responsive header using proper JavaScript**

* <header>

<nav>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a></li>

<li><a href="#">Contact</a></li>

</ul>

</nav>

</header>

<script>

function adjustHeader() {

var header = document.querySelector("header");

if (window.innerWidth <= 768) {

header.classList.add("responsive-header");

} else {

header.classList.remove("responsive-header");

}

}

window.addEventListener("resize", adjustHeader);

</script>

**4. Create a form and validate using JavaScript**

* <form id="myForm" onsubmit="return validateForm()">

<input type="text" id="name" required>

<input type="email" id="email" required>

<button type="submit">Submit</button>

</form>

<script>

function validateForm() {

var name = document.getElementById("name").value;

var email = document.getElementById("email").value;

if (name === "" || email === "") {

alert("Please fill in all fields.");

return false;

}

if (!validateEmail(email)) {

alert("Invalid email address.");

return false;

}

return true;

}

function validateEmail(email) {

// Add your email validation logic here.

return /\S+@\S+\.\S+/.test(email);

}

</script>

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

* <button onclick="openModal()">Open Modal</button>

<div id="myModal" class="modal">

<div class="modal-content">

<span class="close" onclick="closeModal()">&times;</span>

<p>Modal content goes here.</p>

</div>

</div>

<script>

function openModal() {

var modal = document.getElementById("myModal");

modal.style.display = "block";

}

function closeModal() {

var modal = document.getElementById("myModal");

modal.style.display = "none";

}

</script>

**6. Use external js library to show slider**

**7. Prevent the browser when i click the form submit button**

**New request**

1. **What is JSON**

* JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa).

**2.What is promises**

* A Promise is an object representing the eventual completion or failure of an asynchronous operation.

**3.Write a program of promises and handle that promises also**

|  |
| --- |
| <!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> |
|  |  |
|  | <script> |
|  |  |
|  | function Emotions(emotion) |
|  | { |
|  | console.log(emotion) |
|  | } |
|  |  |
|  | const PromiseKar = new Promise(function(call,NoCall){ |
|  |  |
|  | if(true) |
|  | { |
|  | return call("Thanks for calling...!") |
|  | } |
|  | else |
|  | { |
|  | return NoCall("Bhad me ja...!"); |
|  | } |
|  |  |
|  | }) |
|  |  |
|  | console.log(PromiseKar) |
|  |  |
|  | PromiseKar.then( |
|  | // function(Khushi){console.log(Khushi)}, |
|  | // function(Galiya){console.log(Galiya)} |
|  |  |
|  | function(Khushi){Emotions(Khushi)}, |
|  | function(Galiya){Emotions(Galiya)} |
|  | ) |
|  |  |
|  |  |
|  | </script> |
|  |  |
|  | </body> |
|  | </html>  **4. Display all the product from the api in your HTML page** |

* One simple way you could do so is add an element where you want to display the data and then in your clickedEvent() method just select the element and display the result of the api request.

**5. What is JavaScript Output method?**

* JavaScript can "display" data in different ways: Writing into an HTML element, using innerHTML . Writing into the HTML output using document.write() .

**6. How to used JavaScript Output method?**

* + - **JavaScript can "display" data in different ways:**
    - Writing into an HTML element, using innerHTML
    - Writing into the HTML output using document.write() .
    - Writing into an alert box, using window.alert() .
    - Writing into the browser console, using console.log() .

**6. How to used JavaScript Events to do all examples?**

* Onload:  When your page loads, it performs accordingly.
* Onclick: When a user clicks on a button or inputs it occurs.
* Onmouseover: When a user mouses over on the button.
* Onfocus:  Certain scenarios when a user keeps the cursor in a form field.
* Onblur: If a particular form field leaves within it.