● Write a program to Show an alert

```html
<!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 onclick="Alert()">Click here</button>

<script>
    function Alert() {
        alert("Hello");
    }
</script>
</body>
</html>
```

● 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"

```
<!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>
        // 1.
        document.write("1. " + (5 > 4) + "<br>");

        // 2.
        document.write("2. " + ("apple" > "pineapple") + "<br>");

        // 3.
        document.write("3. " + ("2" > "12") + "<br>");

        // 4.
        document.write("4. " + (undefined == null) + "<br>");

        // 5.
        document.write("5. " + (undefined === null) + "<br>");

        // 6.
        document.write("6. " + (null == "\n0\n") + "<br>");

        // 7.
        document.write("7. " + (null === +"\n0\n") + "<br>");
    </script>
</body>

</html>
```

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

- Yes, an alert will be shown. In JavaScript, the condition in an if statement is evaluated based on the truthiness or falsiness of the expression. The string "0" is a non-empty string, and in a boolean context, it is considered truthy. Therefore, the code inside the if block will be executed, and the alert('Hello') statement will be triggered, showing the alert with the message "Hello".

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

- To run the code alert(null || 2 || undefined); in a JavaScript environment, you can use a browser console or include it in an HTML file. The result will be an alert displaying the value

Here's an example HTML file:

```html
<!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>
        alert(null || 2 || undefined);
    </script>
</body>
</html>
```

● 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?');
            }
        }
        var userAge = prompt('Enter your age:');
        var result = checkAge(userAge);

        if (result) {
            alert('Access granted!');
        } else {
            alert('Access denied!');
        }
    </script>
</body>

</html>
```

● 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.");

} } Data Types and Objects

```html
<!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>
```

● 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.

```html
<!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>
        // a) Create an empty object user.
        let user = {};
        console.log("a) Empty object user:", user);

        // b) Add the property name with the value John.
        user.name = 'John';
        console.log("b) Added property name:", user);

        // c) Add the property surname with the value Smith.
        user.surname = 'Smith';
        console.log("c) Added property surname:", user);

        // d) Change the value of the name to Pete.
        user.name = 'Pete';
        console.log("d) Changed value of name to Pete:", user);

        // e) Remove the property name from the object.
        delete user.name;
        console.log("e) Removed property name:", user);

    </script>
</body>

</html>
```

● 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); // ?

```html
<!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>
        let fruits = ["Apples", "Pear", "Orange"];

        // Both fruits and shoppingCart reference the same array in memory.
        let shoppingCart = fruits;

        // Modifying shoppingCart will also modify fruits.
        shoppingCart.push("Banana");

        // Now, alerting the length of fruits will display 4.
        alert(fruits.length);

    </script>
</body>

</html>
```

● 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

```
<!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>
        let john = { name: "John", age: 25 };
        let pete = { name: "Pete", age: 30 };
        let mary = { name: "Mary", age: 28 };
        let users = [john, pete, mary];

        // My Code
        let names = users.map(user => user.name);
        alert(names);
    </script>
</body>

</html>
```

● 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

```
<!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>
        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
}));

console.log(usersMapped);

console.log(usersMapped[0].id);
console.log(usersMapped[0].fullName);


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

● 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

```
<!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 sumSalaries(salaries) {
  let sum = 0;

  for (let salary of Object.values(salaries)) {
    sum += salary;
  }

  return sum;
}

let salaries = { "John": 100, "Pete": 300, "Mary": 250 };
document.write(sumSalaries(salaries));
    </script>
</body>
</html>
```

● 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};

```
<!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>
        let user = { name: "John", years: 30 };

        let { name } = user;
        console.log(name);

        let { years: age } = user;
        console.log(age);

        let { isAdmin = false } = user;
        console.log(isAdmin);

    </script>
</body>

</html>
```

● 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}; Document, Event and Controls

```
<!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>
        let user = { name: "John Smith", age: 35 };

        let jsonString = JSON.stringify(user);

        console.log(jsonString);

        let parsedUser = JSON.parse(jsonString);

        console.log(parsedUser);
        var console: Console
        console.log(parsedUser.name);
        console.log(parsedUser.age);

    </script>
</body>

</html>
```

● Create a program to hide/show the password

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Password Toggle</title>
  <style>
    #passwordToggle {
      margin-top: 10px;
    }
  </style>
</head>
<body>
  <label for="password">Password:</label>
  <input type="password" id="password" placeholder="Enter your password">
  <input type="checkbox" id="passwordToggle"> Show Password

  <script>
    document.getElementById('passwordToggle').addEventListener('change', function() {
      var passwordInput = document.getElementById('password');
      passwordInput.type = this.checked ? 'text' : 'password';
    });
  </script>
</body>
</html>
```

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

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Class Selector</title>
  <style>
    .sample-class {
      margin: 10px;
      padding: 10px;
      border: 1px solid #ccc;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <div class="sample-class">Element 1</div>
  <div class="sample-class">Element 2</div>
  <div class="sample-class">Element 3</div>

  <button id="alertButton">Click me</button>

  <script>
    document.getElementById('alertButton').addEventListener('click', function() {
      var elements = document.querySelectorAll('.sample-class');

      elements.forEach(function(element) {
        element.addEventListener('click', function() {
          alert('Clicked: ' + this.textContent);
        });
      });
    });
  </script>
</body>
</html>
```

- Create a responsive header using proper JavaScript

● Create a form and validate using JavaScript

```
<!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>
    <form action="" onsubmit="return ValidData(input, msg)">
        <table cellpadding="5px" align="center">
            <tr>
                <td><label for="">Name</label></td>
                <td><input type="text" name="" id="name" onblur="ValidData(this,'nameerror')"
                        onkeydown="NameValid(this, 'nerror')">
                    <span id="nameerror"></span>
                    <span id="nerror"></span>
                </td>
            </tr>
            <tr>
                <td><label for="">Email</label></td>
                <td><input type="email" name="" id="email" id="name" onblur="ValidData(this,'emailerror')"
                        onkeydown="EmailValid(this, 'eerror')">
                    <span id="emailerror"></span>
                    <span id="eerror"></span>
                </td>
            </tr>
            <tr>
                <td><label for="">Password</label></td>
                <td><input type="password" name="" id="password" minlength="0" maxlength="8"
                        onblur="ValidData(this,'Passworderror')" onkeydown="PasswordValid(this, 'Perror')">
                    <span id="Passworderror"></span>
                    <span id="Perror"></span>
                </td>
            </tr>
            <tr>
```

```
                <td><label for="">Mobile No</label></td>
                <td><input type="text" name="" id="number" minlength="0" maxlength="10"
                        onblur="ValidData(this,'Mobileerror')" onkeydown="MobileValid(this, 'Merror')">
                    <span id="Mobileerror"></span>
                    <span id="Merror"></span>
                </td>
            </tr>
            <tr>
                <td colspan="2" align="center"><input type="submit" name="" id=""></td>
            </tr>
        </table>
</body>

</html>
<script>
    function ValidData(input, msg) {
        if (input.value == "") {
            document.getElementById(msg).style.color = "red";
            document.getElementById(msg).innerHTML = "Please Enter Data!";
            return false;
        } else {
            document.getElementById(msg).innerHTML = "";
            return true;
        }
    }
    function NameValid(input, msg) {
        var name = /^[a-zA-Z]+$/
        if (name.test(input.value)) {
            document.getElementById(msg).innerText = "";
            return true;
        }
        else {
            document.getElementById(msg).style.color = "red";
            document.getElementById(msg).innerText = "Please Enter Character Only!"
            return false;
        }
```

```
}
function EmailValid(input, msg) {
    var email = /^([a-z0-9.-_]+@+([a-z0-9-._]).([a-z]{3}))/
    if (email.test(input.value)) {
        document.getElementById(msg).innerText = "";
        return true;
    }
    else {
        document.getElementById(msg).style.color = "red";
        document.getElementById(msg).innerText = "Please Enter Valid Email!"
        return false;
    }

}
function PasswordValid(input, msg) {
    var password = /^[a-zA-z0-9.-]{3,5}$/
    if (Password.test(input.value)) {
        document.getElementById(msg).innerText = "";
        return true;
    }
    else {
        document.getElementById(msg).style.color = "red";
        document.getElementById(msg).innerText = "Please Enter Min 3 and max 5 Characters!"
        return false;

    }
}
function MobileValid(input, msg) {
    var mobile = /^[0-9]{10}$/
    if (mobile.test(input.value)) {
        document.getElementById(msg).innerText = "";
        return true;
    }
    else {
        document.getElementById(msg).style.color = "red";
        document.getElementById(msg).innerText = "Please Enter Valid Mobile Number !"
        return false;
```

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

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Modal Box</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            background-color: #f4f4f4;
        }

        .modal {
            display: none;
            position: fixed;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background-color: rgba(0, 0, 0, 0.5);
            justify-content: center;
            align-items: center;
            z-index: 1;
        }

        .modal-content {
            background-color: #fff;
            padding: 20px;
            border-radius: 5px;
            text-align: center;
```

```
        }

        .button-container {
            margin-top: 20px;
        }

        .button {
            padding: 10px 20px;
            margin: 0 10px;
            cursor: pointer;
            border: none;
            border-radius: 5px;
        }

        .open-button {
            background-color: #4caf50;
            color: white;
        }

        .close-button {
            background-color: #f44336;
            color: white;
        }

        .submit-button {
            background-color: #2196f3;
            color: white;
        }
    </style>
</head>

<body>

    <button class="button open-button" onclick="openModal()">Open Modal</button>

    <div class="modal" id="myModal">
        <div class="modal-content">
```

```
            <p>This is a modal box. Click the buttons below!</p>

            <div class="button-container">
                <button class="button close-button" onclick="closeModal()">Close Modal</button>
                <button class="button submit-button" onclick="submitForm()">Submit</button>
            </div>
        </div>
    </div>

    <script>
        function openModal() {
            var modal = document.getElementById('myModal');
            modal.style.display = 'flex';
        }

        function closeModal() {
            var modal = document.getElementById('myModal');
            modal.style.display = 'none';
        }

        function submitForm() {
            alert('Form submitted!');
            closeModal();
        }
    </script>
</body>

</html>
```

- Use external js library to show slider

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Image Slider</title>
    <link rel="stylesheet" type="text/css"
        href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick.min.css" />
    <link rel="stylesheet" type="text/css"
        href="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick-theme.min.css" />
</head>

<body>
    <div class="slider-container">
        <div class="slider">
            <div><img src="pexels-bri-schneiter-346529.jpg" height="650" width="1500" alt="Image 1"></div>
            <div><img src="pexels-pixabay-147411.jpg" height="650" width="1500" alt="Image 2"></div>
            <div><img src="pexels-ákos-szabó-440731.jpg" height="650" width="1500" alt="Image 3"></div>
        </div>
    </div>

    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick.min.js"></script>
    <!-- JS File Path-->
    <script src="Q19.js"></script>
</body>

</html>
```

```
$(document).ready(function(){
    $('.slider').slick({
      autoplay: true,
      autoplaySpeed: 2000,
      dots: true,
      arrows: true
    });
  });
```

- Prevent the browser when i click the form submit button New Request

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prevent Form Submission</title>
</head>
<body>
    <table align="center" cellspacing="10" cellpadding="5">
        <form id="myForm">
            <tr>
                <td> <label for="username">Username:</label></td>
                <td> <input type="text" id="username" name="username" required></td>
            </tr>
            <tr>
                <td><label for="password">Password:</label></td>
                <td> <input type="password" id="password" name="password" required></td>
            </tr>
            <tr>
                <td></td>
                <td><button type="submit" onclick="submitForm()">Submit</button></td>
            </tr>
        </form>
    </table>
    <script>
        function submitForm(event) {
            event.preventDefault();
            var username = document.getElementById('username').value;
            var password = document.getElementById('password').value;
            alert('Form submitted! Username: ' + username + ', Password: ' + password);
        }
        document.getElementById('myForm').addEventListener('submit', submitForm);
    </script>
</body>

</html>
```