



UNIVERSITY MALAYSIA TERENGGANU
FACULTY OF COMPUTER SCIENCE & MATHEMATICS

FRONT-END PROGRAMMING
CSF 3103

LAB REPORT 4:
Javascript

Prepared by:
NUR ARIFAH BINTI MOHD HANAFIAH
S66428

Prepared for:
DR RABIEI BIN MAMAT

BACHELOR OF COMPUTER SCIENCE (MOBILE COMPUTING) WITH HONORS
SEMESTER II 2023/2024

Lab 4

Task 1 – JavaScript Function

1. Write a function to find the square of a given number

```
1  // Arifah -S66428
2
3  // The core logic of finding the square of a number
4  function findSquare(number) {
5      return number * number;
6  }
```

2. Write a function to find sum of cubes of two numbers

```
8  // Function find sumOfCubes using math.pow calculation
9  function sumOfCubes(num1, num2) {
10     return Math.pow(num1, 3) + Math.pow(num2, 3);
11 }
```

3. Write a function to reverse a number
[Hint n=12345 output : 54321]

```
13 // Function the reversedNumber
14 function reverseNumber(number) {
15     let reversed = 0;
16     while (number > 0) {
17         reversed = reversed * 10 + (number % 10);
18         number = Math.floor(number / 10);
19     }
20     return reversed; //return reversed number
21 }
34 console.log("Reverse of 12345:", reverseNumber(12345)); // Output: 54321
```

4. Write a function to print all numbers between 1 and 100 which is divisible by given number z

```
23 // Function to print all numbers between 1 and 100
24 function printDivisibleNumbers(z) {
25     for (let i = 1; i <= 100; i++) {
26         if (i % z === 0) {
27             console.log(i);
28         }
29     }
30 }
35 console.log("Numbers divisible by 5 between 1 and 100:"); // testing number by 5 1-100
36 printDivisibleNumbers(5); // Output: 5, 10, 15, 20, ..., 95, 100
37
```

Task 2 - JavaScript Recursion Function

1. Write a JavaScript function to find sum of digits of a number

```
JS Task1.js X JS Task2.js X JS Task3.js
C: > Users > Lenovo > OneDrive > Desktop > UMT > SEMESTER IV > PENGATURCARAAN FRONT-END > Lab 4 - S66428 > JS Task2.js > ...
1 // Arifah -S66428
2
3 // Function to use recursion to get the sum of a number's digits
4 function sumOfDigits(number) {
5     if (number === 0) {
6         return 0;
7     } else {
8         return number % 10 + sumOfDigits(Math.floor(number / 10));
9     }
10 }
11
22 console.log("Sum of digits of 12345:", sumOfDigits(12345)); // Output: 15
```

2. Write a JavaScript program to compute x raised to the power y using recursion

```
11
12 // Recursive function to calculate x raised to the power of y
13 function power(x, y) {
14     if (y === 0) {
15         return 1;
16     } else {
17         return x * power(x, y - 1);
18     }
19 }
23 console.log("2 raised to the power 5:", power(2, 5)); // Output: 32
```

Task 3 – JavaScript Object and Prototype

1. Write a JavaScript program to create object product,
 - a. Add the property Product Name, Quantity and price.

```
1  // Arifah -S66428
2
3  // Create a JavaScript object representing a product ..
4  let product = {
5      productName: "Scarf",
6      quantity: 10,
7      price: 10
8  };
```

- b. Access all the properties and display them.

```
10 // Access and display all properties of the product..
11 console.log("Product Name:", product.productName);
12 console.log("Quantity:", product.quantity);
13 console.log("Price:", product.price);
14
```

2. Write a JavaScript program to create object book
 - a. Add the property book name, author name

```
15 // create object and property for book ..
16 let book = {
17     bookName: "Object Oriented Programming",
18     authorName: "Kate Hermsworth"
19 };
```

- b. Add the prototype property price .

```
21 // write the prototype for price ..
22 Object.setPrototypeOf(book, { price: 80 });
```

- c. Display all the properties.

```
24 // The display about properties of book ..
25 console.log("Book Name:", book.bookName);
26 console.log("Author Name:", book.authorName);
27 console.log("Price:", book.price);
```

3. Write a JavaScript program to create Parent object employee (Property : Employee Name , Employee Id , Salary) and Child object Manager (Property : Manager Name , Branch). Inherit all the properties of employee and display all the properties.

```
29 // create parent object for employee..
30 let employee = {
31     employeeName: "Arifah Hanafiah",
32     employeeId: "S66428",
33     salary: 7500
34 };
35
36 // create the child object manager...
37 let Manager = Object.create(employee);
38 Manager.managerName = "Jennie Kim";
39 Manager.branch = "YG branch";
40
41 //display all properties ...
42 console.log("Employee Name:", Manager.employeeName);
43 console.log("Employee ID:", Manager.employeeId);
44 console.log("Salary:", Manager.salary);
45 console.log("Manager Name:", Manager.managerName);
46 console.log("Branch:", Manager.branch);
47 |
```

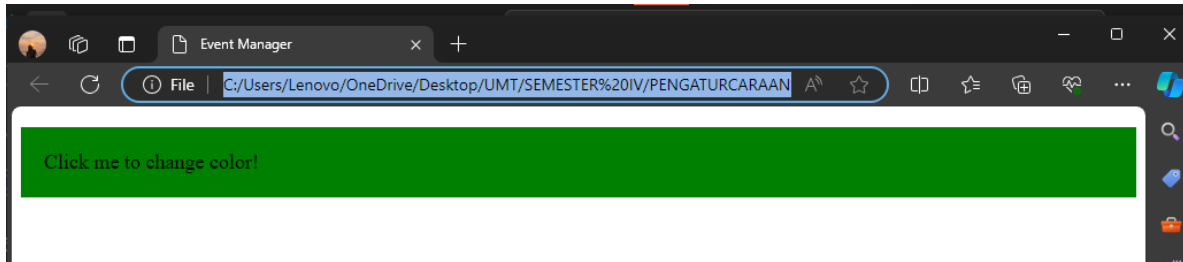
Task 4 – Event Manager

1. Create a HTML page with <p> paragraph. Change the paragraph color according to the following mouse events
 - a. Onclick, yellow background
 - b. ondblclick, blue background
 - c. onmouseover , red background
 - d. onmouseout, green background

The html code:

```
C: > Users > Lenovo > OneDrive > Desktop > UMT > SEMESTER IV > PENGATURCARAAN FRONT-END > Lab 4 - S66428 > Task 4.html > html >
1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Event Manager</title>
8 <style>
9     p {
10         padding: 20px;
11         font-size: 18px;
12         cursor: pointer;
13     }
14 </style>
15 </head>
16 <body>
17     <p id="paragraph">Click me to change color!</p>
18
19     <script>
20
21         let paragraph = document.getElementById("paragnraph");
22
23         // create handler for yellow background ...
24         paragraph.onclick = function() {
25             this.style.backgroundColor = "yellow";
26         };
27
28         // create handler for blue background ...
29         paragraph.ondblclick = function() {
30             this.style.backgroundColor = "blue";
31         };
32
33         // create handler for red background ...
34         paragraph.onmouseover = function() {
35             this.style.backgroundColor = "red";
36         };
37
38         // create handler for green backgroud ...
39         paragraph.onmouseout = function() {
40             this.style.backgroundColor = "green";
41         };
42     </script>
43 </body>
44 </html>
45
```

The Output :

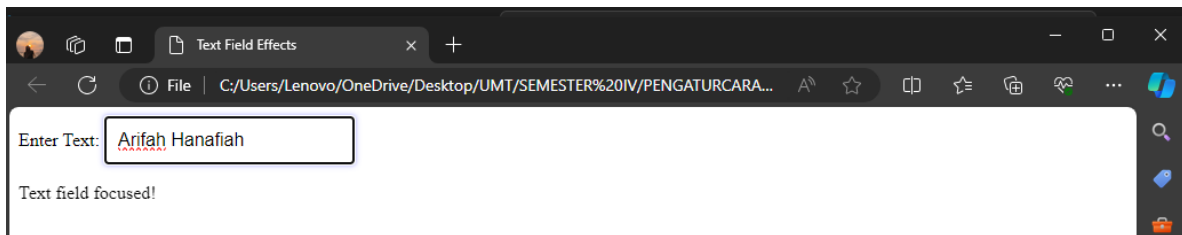


2. Create a HTML page with textfield. Show some effects on the textfield when the following events occurred:
 - a. Onchange
 - b. Onfocus
 - c. Onblur

The html code:

```
C: > Users > Lenovo > OneDrive > Desktop > UMT > SEMESTER IV > PENGATURCARAAN FRONT-END > Lab 4 - S66428 > Task4-Q2.html > html > head
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>Text Field Effects</title>
7  <style>
8      /* Style ...*/
9      input[type="text"] {
10         padding: 10px;
11         font-size: 16px;
12         border: 2px solid #ccc;
13         transition: border-color 0.3s ease; /* for smooth transaction*/
14     }
15
16     /* create Onfocus ... */
17     input[type="text"]:focus {
18         border-color: #dodgerblue;
19         box-shadow: 0 0 5px #0000ff;
20     }
21 </style>
22 </head>
23 <body>
24     <label for="textField">Enter Text:</label>
25     <input type="text" id="textField" onchange="changeEffect()" onfocus="focusEffect()" onblur="blurEffect()">
26     <p id="feedback"></p>
27
28     <script>
29         // Create the Onchange for textfield ...
30         function changeEffect() {
31             let textField = document.getElementById("textField");
32             textField.style.backgroundColor = "lightyellow";
33             document.getElementById("feedback").innerText = "Text changed!";
34         }
35
36         // Create the Onfocus for textfield ...
37         function focusEffect() {
38             let textField = document.getElementById("textField");
39             textField.style.borderStyle = "dashed";
40             document.getElementById("feedback").innerText = "Text field focused!";
41         }
42
43         // Create the Onblur for textfield ...
44         function blurEffect() {
45             let textField = document.getElementById("textField");
46             textField.style.borderStyle = "solid";
47             textField.style.borderColor = "#ccc";
48             textField.style.backgroundColor = "white";
49             document.getElementById("feedback").innerText = "Text field blurred!";
50         }
51     </script>
52 </body>
53 </html>
```


The Output :



Task 5

Given the following HTML table

| | | | |
|----|--------------|--|------------|
| 1 | Ahmad Faisal | ahmadfaisal@gmail.com | 0199088888 |
| 2. | Ismail Sabri | isabri@mail.com | 0199076760 |
| 3 | Fateh Yakin | ffateh@hotmail.com | 0176067762 |

1. Using javascript add the following record into table
 - a. Name: Mukhriz Jamil Asoka
 - b. Email: mukriz@corp.jo
 - c. Phone: 651181187223
2. Using javascript add the table header as follow:
 - a. #, Name, Email, Phone #
3. Using javascript, delete any row from table when clicked on that row

The html code:

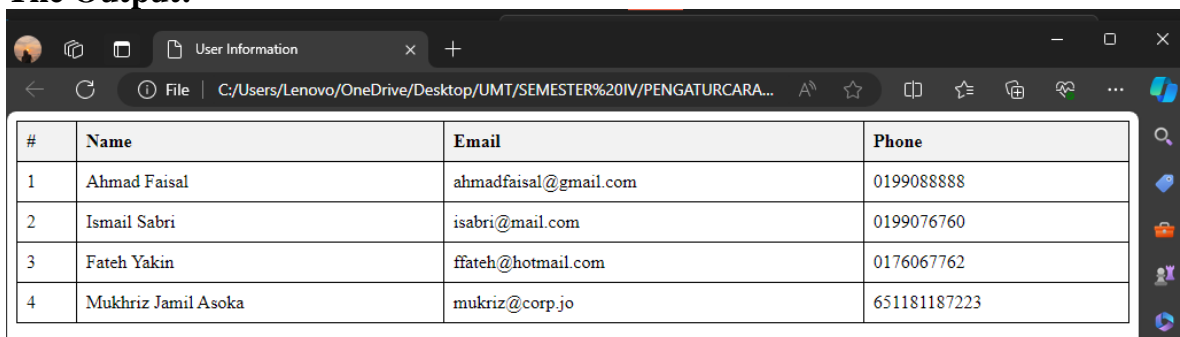
```
JS Task5.js Task 5.html X JS Task 6.js
Task 5.html > html > body > table > tbody
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>User Information</title>
7   <style>
8     table {
9       width: 100%;
10      border-collapse: collapse;
11    }
12    th, td {
13      border: 1px solid black;
14      padding: 8px;
15      text-align: left;
16    }
17    th {
18      background-color: #f2f2f2;
19    }
20  </style>
21 </head>
22 <body>
23   <table>
24     <thead>
25       <tr>
26         <th>#</th>
27         <th>Name</th>
28         <th>Email</th>
29         <th>Phone</th>
30       </tr>
31     </thead>
32     <tbody>
```

```
32     <tbody>
33         <tr>
34             <td>1</td>
35             <td>Ahmad Faisal</td>
36             <td>ahmadfaisal@gmail.com</td>
37             <td>0199088888</td>
38         </tr>
39         <tr>
40             <td>2</td>
41             <td>Ismail Sabri</td>
42             <td>isabri@mail.com</td>
43             <td>0199076760</td>
44         </tr>
45         <tr>
46             <td>3</td>
47             <td>Fateh Yakin</td>
48             <td>ffateh@hotmail.com</td>
49             <td>0176067762</td>
50         </tr>
51     </tbody>
52 </table>
53
54
55     <script src="Task5.js"></script>
56 </body>
57 </html>
58
```

The js code:

```
JS Task5.js x Task 5.html JS Task 6.js
JS Task5.js > ...
1 // create table tbody ...
2 var tableBody = document.querySelector("table tbody");
3
4 // create the row in the table ...
5 var newRow = document.createElement("tr");
6
7 // new row for new data ...
8 newRow.innerHTML = `
9     <td>4</td>
10    <td>Mukhriz Jamil Asoka</td>
11    <td>mukriz@corp.jo</td>
12    <td>651181187223</td>
13 `;
14
15 tableBody.appendChild(newRow);
16
17 var rows = document.querySelectorAll("table tbody tr");
18
19 // create click event listener ...
20 rows.forEach(function(row) {
21     row.addEventListener("click", function() {
22         this.remove(); // delete the row when click it ...
23     });
24 });
25
```

The Output:



The screenshot shows a web browser window with a single tab titled "User Information". The address bar shows the file path: "C:/Users/Lenovo/OneDrive/Desktop/UMT/SEMESTER%20IV/PENGATURCARA...". The main content area displays a table with four columns: "#", "Name", "Email", and "Phone". The table contains four rows of data, with the last row being the one added by the JavaScript code.

| # | Name | Email | Phone |
|---|---------------------|-----------------------|--------------|
| 1 | Ahmad Faisal | ahmadfaisal@gmail.com | 0199088888 |
| 2 | Ismail Sabri | isabri@mail.com | 0199076760 |
| 3 | Fateh Yakin | ffateh@hotmail.com | 0176067762 |
| 4 | Mukhriz Jamil Asoka | mukriz@corp.jo | 651181187223 |

Task 6

Write a JavaScript program to move two small squares inside one big square in a random manner. User should be able to start and stop this animation using button based events

`Math.floor(Math.random() * Math.floor(max))` will give you a random number that is less than max value

The html code:

```
Task 6.html > html > body > script
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>Square Animation</title>
7  <style>
8      #container {
9          position: relative;
10         width: 400px;
11         height: 400px;
12         border: 2px solid black;
13     }
14
15     .square {
16         position: absolute;
17         width: 50px;
18         height: 50px;
19         background-color: red;
20     }
21
22     #square1 {
23         left: 0;
24         top: 0;
25     }
26
27     #square2 {
28         right: 0;
29         bottom: 0;
30     }
```

```

31
32     button {
33         margin-top: 10px;
34     }
35 </style>
36 </head>
37 <body>
38 <div id="container">
39     <div id="square1" class="square"></div>
40     <div id="square2" class="square"></div>
41 </div>
42 <button id="startBtn">Start Animation</button>
43 <button id="stopBtn">Stop Animation</button>
44
45 <script src="Task 6.js"></script>
46 </body>
47 </html>
48

```

The js code :

```

JS Task 6.js > document.addEventListener("DOMContentLoaded") callback > stopAnimation
1  document.addEventListener("DOMContentLoaded", () => {
2      const container = document.getElementById("container");
3      const square1 = document.getElementById("square1");
4      const square2 = document.getElementById("square2");
5      const startBtn = document.getElementById("startBtn");
6      const stopBtn = document.getElementById("stopBtn");
7      let animationInterval;
8
9      function getRandomPosition() {
10         const maxWidth = container.offsetWidth - square1.offsetWidth;
11         const maxHeight = container.offsetHeight - square1.offsetHeight;
12         const x = Math.floor(Math.random() * maxWidth);
13         const y = Math.floor(Math.random() * maxHeight);
14         return { x, y };
15     }
16
17     function moveSquares() {
18         const pos1 = getRandomPosition();
19         const pos2 = getRandomPosition();
20         square1.style.left = pos1.x + "px";
21         square1.style.top = pos1.y + "px";
22         square2.style.left = pos2.x + "px";
23         square2.style.top = pos2.y + "px";
24     }
25
26     function startAnimation() {
27         moveSquares();
28         animationInterval = setInterval(moveSquares, 1000);
29     }
30

```

```

25
26     function startAnimation() {
27         moveSquares();
28         animationInterval = setInterval(moveSquares, 1000);
29     }
30
31     function stopAnimation() {
32         clearInterval(animationInterval);
33         console.log("Animation stopped");
34     }
35
36     startBtn.addEventListener("click", startAnimation);
37     stopBtn.addEventListener("click", stopAnimation);
38 });
39

```

The output :

