



UNIVERSITI MALAYSIA TERENGGANU

FACULTY OF OCEAN ENGINEERING TECHNOLOGY & INFORMATICS

SEMESTER 2 2022/2023

FRONT-END PROGRAMMING

CSM 3103

LAB 3 REPORT

PREPARED FOR:

DR. RABIEI BIN MAMAT

PREPARED BY:

MOHAMAD HAZIM BIN MOHD SHAKRI (S61770)

Lab 3

Task 1 – JavaScript Function

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

```
1  function square(num) {  
2      return Math.pow(num, 2);  
3  }  
4
```

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

```
1  function sumOfCubes(num1, num2) {  
2      return Math.pow(num1, 3) + Math.pow(num2, 3);  
3  }  
4
```

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

```
1  function reverseNumber(num) {  
2      let reversed = 0;  
3      while (num !== 0) {  
4          let digit = num % 10;  
5          reversed = reversed * 10 + digit;  
6          num = Math.floor(num / 10);  
7      }  
8      return reversed;  
9  }  
10
```

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

```
1  function printDivisibleNumbers(z) {  
2      for (let i = 1; i <= 100; i++) {  
3          if (i % z === 0) {  
4              console.log(i);  
5          }  
6      }  
7  }  
8
```

Task 2 - JavaScript Recursion Function

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

```
1  function sumOfDigits(num) {  
2      let sum = 0;  
3      while (num !== 0) {  
4          let digit = num % 10;  
5          sum += digit;  
6          num = Math.floor(num / 10);  
7      }  
8      return sum;  
9  }
```

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

```
1  function power(x, y) {  
2      if (y === 0) {  
3          return 1;  
4      } else if (y % 2 === 0) {  
5          let temp = power(x, y/2);  
6          return temp * temp;  
7      } else {  
8          let temp = power(x, Math.floor(y/2));  
9          return x * temp * temp;  
10     }  
11 }  
12
```

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 let product = {  
2   productName: "Apple",  
3   quantity: 10,  
4   price: 1.99  
5 };
```

- b. Access all the properties and display them.

```
1 console.log(product.productName); // Output: "Apple"  
2 console.log(product.quantity);    // Output: 10  
3 console.log(product.price);       // Output: 1.99  
4  
5 product.quantity = 20;  
6 console.log(product.quantity);    // Output: 20
```

2. Write a JavaScript program to create object book

- a. Add the property book name, author name

```
1 let book = {  
2   bookName: "The Great Gatsby",  
3   authorName: "F. Scott Fitzgerald"  
4 };
```

- b. Add the prototype property price.

```
1 let book = {  
2   bookName: "The Alchemist",  
3   authorName: "Paulo Coelho",  
4   price: 9.99  
5 };
```

- c. Display all the properties.

```
1 console.log(book.bookName);    // Output: "The Alchemist"  
2 console.log(book.authorName);  // Output: "Paulo Coelho"  
3 console.log(book.price);       // Output: 9.99  
4  
5 book.price = 12.99;  
6 console.log(book.price);      // Output: 12.99
```

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 allthe properties.

```
1  // Parent object
2  let Employee = function(employeeName, employeeId, salary) {
3      this.employeeName = employeeName;
4      this.employeeId = employeeId;
5      this.salary = salary;
6  };
7
8  // Child object
9  let Manager = function(managerName, branch) {
10     this.managerName = managerName;
11     this.branch = branch;
12 };
13 Manager.prototype = new Employee();
14
15 // Create an instance of Manager
16 let manager = new Manager("John Doe", "New York");
17
18 // Access properties of Manager object
19 console.log(manager.employeeName); // Output: undefined
20 console.log(manager.employeeId);   // Output: undefined
21 console.log(manager.salary);        // Output: undefined
22 console.log(manager.managerName);   // Output: "John Doe"
23 console.log(manager.branch);        // Output: "New York"
```

4.

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

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Change paragraph color on click</title>
5      <style>
6          p {
7              color: black;
8              font-size: 24px;
9              cursor: pointer;
10         }
11         p.clicked {
12             background-color: yellow;
13         }
14     </style>
15 </head>
16 <body>
17     <p onclick="changeColor()">Click me to change color</p>
18
19     <script>
20         function changeColor() {
21             let p = document.querySelector("p");
22             p.classList.add("clicked");
23         }
24     </script>
25 </body>
26 </html>
```

b. ondblclick, blue background

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Change Paragraph Color on Double Click</title>
6      <style>
7        /* Default style */
8        p {
9          color: black;
10         background-color: white;
11       }
12       /* Style on double-click */
13       p:hover {
14         color: white;
15         background-color: blue;
16       }
17     </style>
18   </head>
19   <body>
20     <p ondblclick="this.style.color='white'; this.style.backgroundColor='blue'">
21       This is a paragraph. Double-click on it to change its color.
22     </p>
23   </body>
24 </html>
```


c. onmouseover , red background

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Change paragraph color on mouseover event</title>
5    <style>
6      /* Define the default styles for the paragraph */
7      p {
8        background-color: white;
9        color: black;
10       padding: 10px;
11     }
12
13     /* Define the styles for the paragraph when the mouse pointer is over it */
14     p:hover {
15       background-color: red;
16       color: white;
17     }
18   </style>
19 </head>
20 <body>
21   <p onmouseover="this.style.backgroundColor='red'">Move the mouse pointer over this para
22 </body>
23 </html>
```

d. onmouseout, green background

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Mouse Event Example</title>
5    <style>
6      /* Default paragraph color */
7      p {
8        color: black;
9      }
10
11     /* Paragraph color on mouse out */
12     p:hover {
13       background-color: green;
14     }
15   </style>
16 </head>
17 <body>
18   <p>This is a paragraph. Move the mouse out of the paragraph to change its color
19 </body>
```

2. Create a HTML page with textfield. Show some effects on the textfield when the following events occurred:

e. Onchange

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Text Field Effects on onchange event</title>
5      <style>
6        /* Add some styles to the text field */
7        input[type="text"] {
8          font-size: 16px;
9          padding: 10px;
10         border: 2px solid #ccc;
11         border-radius: 4px;
12         transition: border-color 0.2s ease-in-out;
13       }
14
15       /* Add a red border color to the text field when it's invalid */
16       input[type="text"]:invalid {
17         border-color: red;
18       }
19
20       /* Add a green border color to the text field when it's valid */
21       input[type="text"]:valid {
22         border-color: green;
23       }
24     </style>
25   </head>
26   <body>
27     <h1>Text Field Effects on onchange event</h1>
28     <label for="text-field">Enter some text:</label>
29     <input type="text" id="text-field" onchange="validateTextField()" required>
30
31     <script>
32       // Define a function to validate the text field
33       function validateTextField() {
34         let textField = document.getElementById("text-field");
35         if (textField.value === "") {
36           textField.setCustomValidity("Please enter some text");
37         } else {
38           textField.setCustomValidity("");
39         }
40       }
41     </script>
42   </body>
43 </html>
```

f. Onfocus

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Text Field Effects on onfocus event</title>
5      <style>
6        /* Add some styles to the text field */
7        input[type="text"] {
8          font-size: 16px;
9          padding: 10px;
10         border: 2px solid #ccc;
11         border-radius: 4px;
12         transition: border-color 0.2s ease-in-out;
13       }
14
15       /* Add a blue border color to the text field when it's focused */
16       input[type="text"]:focus {
17         border-color: blue;
18         box-shadow: 0 0 5px blue;
19       }
20     </style>
21   </head>
22   <body>
23     <h1>Text Field Effects on onfocus event</h1>
24     <label for="text-field">Enter some text:</label>
25     <input type="text" id="text-field" onfocus="onTextFieldFocus()">
26
27     <script>
28       // Define a function to handle the text field focus event
29       function onTextFieldFocus() {
30         let textField = document.getElementById("text-field");
31         textField.style.backgroundColor = "lightgray";
32       }
33     </script>
34   </body>
35 </html>
```

g. Onblur

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Text Field Effects on onblur event</title>
5      <style>
6        /* Add some styles to the text field */
7        input[type="text"] {
8          font-size: 16px;
9          padding: 10px;
10         border: 2px solid #ccc;
11         border-radius: 4px;
12         transition: border-color 0.2s ease-in-out;
13       }
14
15       /* Add a red border color to the text field when it's invalid */
16       input[type="text"]:invalid {
17         border-color: red;
18       }
19
20       /* Add a green border color to the text field when it's valid */
21       input[type="text"]:valid {
22         border-color: green;
23       }
24     </style>
25   </head>
26   <body>
27     <h1>Text Field Effects on onblur event</h1>
28     <label for="text-field">Enter some text:</label>
29     <input type="text" id="text-field" onblur="validateTextField()" required>
30
31     <script>
32       // Define a function to validate the text field
33       function validateTextField() {
34         let textField = document.getElementById("text-field");
35         if (textField.value === "") {
36           textField.setCustomValidity("Please enter some text");
37         } else {
38           textField.setCustomValidity("");
39         }
40       }
41     </script>
42   </body>
43 </html>
```

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

```
1 let tableData = JSON.parse(localStorage.getItem('tableData')) || [];  
2  
3 // Create a new record object  
4 let newRecord = {  
5   name: 'Mukhriz Jamil Asoka',  
6   email: 'mukriz@corp.jo',  
7   phone: '651181187223'  
8 };  
9  
10 // Add the new record to the table data array  
11 tableData.push(newRecord);  
12  
13 // Save the updated table data back to local storage  
14 localStorage.setItem('tableData', JSON.stringify(tableData));
```

2. Using javascript add the table header as follow:

a. #, Name, Email, Phone #

```
1
2  let table = document.getElementById('myTable');
3
4  let headerRow = document.createElement('tr');
5
6  let headerCell1 = document.createElement('th');
7  headerCell1.textContent = '#';
8  let headerCell2 = document.createElement('th');
9  headerCell2.textContent = 'Name';
10 let headerCell3 = document.createElement('th');
11 headerCell3.textContent = 'Email';
12 let headerCell4 = document.createElement('th');
13 headerCell4.textContent = 'Phone';
14
15 headerRow.appendChild(headerCell1);
16 headerRow.appendChild(headerCell2);
17 headerRow.appendChild(headerCell3);
18 headerRow.appendChild(headerCell4);
19
20 table.appendChild(headerRow);
```

3. Using javascript, delete any row from table when clicked on that row

```
1  let table = document.getElementById('myTable');
2
3  // Add a click event listener to the table
4  table.addEventListener('click', function(event) {
5      // Check if the clicked element is a table row
6      if (event.target.tagName.toLowerCase() === 'tr') {
7          // Delete the row from the table
8          event.target.remove();
9      }
10 });
```

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

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Animation with JavaScript</title>
5      <style>
6        /* Add some styles to the squares */
7        .square {
8          width: 50px;
9          height: 50px;
10         background-color: red;
11         position: absolute;
12       }
13
14       #big-square {
15         width: 300px;
16         height: 300px;
17         border: 2px solid black;
18         position: relative;
19       }
20     </style>
21   </head>
22   <body>
23     <h1>Animation with JavaScript</h1>
24     <button onclick="startAnimation()">Start Animation</button>
25     <button onclick="stopAnimation()">Stop Animation</button>
26     <div id="big-square">
27       <div id="square1" class="square"></div>
28       <div id="square2" class="square"></div>
29     </div>
```

```
31     <script>
32         let square1 = document.getElementById("square1");
33         let square2 = document.getElementById("square2");
34         let bigSquare = document.getElementById("big-square");
35         let animationInterval;
36
37         // Define a function to start the animation
38         function startAnimation() {
39             animationInterval = setInterval(moveSquares, 100);
40         }
41
42         // Define a function to stop the animation
43         function stopAnimation() {
44             clearInterval(animationInterval);
45         }
46
47         // Define a function to move the squares
48         function moveSquares() {
49             let x1 = Math.floor(Math.random() * (bigSquare.clientWidth - square1.clientWidth));
50             let y1 = Math.floor(Math.random() * (bigSquare.clientHeight - square1.clientHeight));
51             let x2 = Math.floor(Math.random() * (bigSquare.clientWidth - square2.clientWidth));
52             let y2 = Math.floor(Math.random() * (bigSquare.clientHeight - square2.clientHeight));
53
54             square1.style.left = x1 + "px";
55             square1.style.top = y1 + "px";
56             square2.style.left = x2 + "px";
57             square2.style.top = y2 + "px";
58         }
59     </script>
60 </body>
61 </html>
```