

## **Practical – 3**

**AIM:**

Perform the Following tasks using the JavaScript.

- Print prime Numbers up to given value in the input box
- Write Script to reverse the given input string
- Create Dynamic Multiplication Table using inputs
- Find the Age from input date.(Ex. 17 Yrs, 3 Monts,13 Days)
- Find the No. of Days between two given dates

**Source Code:**

1. Print prime Numbers up to given value in the input box

**Source Code:**

```
<!DOCTYPE html>

<html>
<body>

<input type="text" id="inputValue" placeholder="Enter a value">

<button onclick="printPrimes()">Submit</button>

<p id="output"></p>

<script>

function printPrimes() {

    var inputValue = document.getElementById("inputValue").value;
    var output = document.getElementById("output");
    var primes = "";
    for (var i = 2; i <= inputValue; i++) {
        var isPrime = true;
        for (var j = 2; j < i; j++) {
            if (i % j === 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            primes += i + " ";
        }
    }
    output.innerHTML = primes;
}

</script>
```

```
}

if (isPrime) {

    primes += i + " ";

}

}

output.innerHTML = primes;

}

</script>

</body>

</html>
```

### Output:

The screenshot shows a web-based IDE interface for OneCompiler. At the top, there's a navigation bar with links like Gmail, YouTube, Maps, Translate, Course Joiner, AWS Builders, Download - CHARU..., Docfly | Dashboard, AWS, QuizHub, Home - OpenFermi..., and L&T Financial Services. Below the navigation bar, there are search, refresh, Q&A, Posts, Challenges, and More buttons.

The main workspace contains three tabs: index.html, styles.css, and script.js. The script.js tab contains the following code:

```
index.html          styles.css          script.js          3yv32eabn
8 <p id="output"></p>
9
10 <script>
11 function printPrimes() {
12     var inputValue = document.getElementById("inputValue").value;
13     var output = document.getElementById("output");
14     var primes = "";
15
16     for (var i = 2; i <= inputValue; i++) {
17         var isPrime = true;
18         for (var j = 2; j < i; j++) {
19             if (i % j === 0) {
20                 isPrime = false;
21                 break;
22             }
23         }
24         if (isPrime) {
25             primes += i + " ";
26         }
27     }
28
29     output.innerHTML = primes;
30 }
31 </script>
32
33 </body>
34 </html>
35
```

To the right of the code editor, there's a preview area showing the output of the script. The input value is set to 10. The output window displays the prime numbers 2, 3, 5, and 7.

**2. Write Script to reverse the given input string****Source code:**

```
<!DOCTYPE html>

<html>
<body>

<input type="text" id="inputString" placeholder="Enter a string">

<button onclick="reverseAndDisplay()">Reverse</button>

<p id="output"></p>

<script>

function reverseAndDisplay() {

    var inputString = document.getElementById("inputString").value;
    var output = document.getElementById("output");
    output.innerHTML = reverseString(inputString);
}

function reverseString(string) {

    return string.split("").reverse().join("");
}

</script>

</body>
</html>
```

**OutPut:**

The screenshot shows a web-based IDE interface. On the left, there's a code editor with a tab for 'index.html' containing the following code:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <input type="text" id="inputString" placeholder="Enter a string">
6 <button onclick="reverseAndDisplay()">Reverse</button>
7
8 <p id="output"></p>
9
10<script>
11<function reverseAndDisplay() {
12   var inputString = document.getElementById("inputString").value;
13   var output = document.getElementById("output");
14
15   output.innerHTML = reverseString(inputString);
16 }
17
18<function reverseString(string) {
19   return string.split('').reverse().join('');
20 }
21</script>
22
23</body>
24</html>
25

```

On the right, there's a terminal window with the command 'vaidik' and a button labeled 'Reverse'. Below it, the output 'kidav' is displayed.

### 3. Create Dynamic Multiplication Table using inputs

**Source code:**

```

<!DOCTYPE html>

<html>

<body>

<label for="rows">Rows:</label>

<input type="number" id="rows" min="1" value="10">

<label for="cols">Columns:</label>

<input type="number" id="cols" min="1" value="10">

<button onclick="createTable()">Create Table</button>

<div id="tableContainer"></div>

<script>

function createTable() {

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

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

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

```

```
tableContainer.innerHTML = "";  
  
var table = document.createElement("table");  
  
table.setAttribute("border", "1");  
  
for (var r = 1; r <= rows; r++) {  
  
    var row = document.createElement("tr");  
  
    for (var c = 1; c <= cols; c++) {  
  
        var cell = document.createElement("td");  
  
        cell.innerHTML = r * c;  
  
        row.appendChild(cell);  
  
    }  
  
    table.appendChild(row);  
  
}  
  
tableContainer.appendChild(table);  
  
}  
  
</script>  
  
</body>  
  
</html>
```

**Output:**

The screenshot shows a web-based IDE interface for OneCompiler. On the left, there are tabs for 'index.html', 'styles.css', and 'script.js'. The 'index.html' tab contains the following code:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <label for="rows">Rows:</label>
6 <input type="number" id="rows" min="1" value="10">
7 <label for="cols">Columns:</label>
8 <input type="number" id="cols" min="1" value="10">
9 <button onclick="createTable()">Create Table</button>
10
11 <div id="tableContainer"></div>
12
13 <script>
14 function createTable() {
15     var rows = document.getElementById("rows").value;
16     var cols = document.getElementById("cols").value;
17     var tableContainer = document.getElementById("tableContainer");
18
19     // Clear previous table
20     tableContainer.innerHTML = "";
21
22     var table = document.createElement("table");
23     table.setAttribute("border", "1");
24
25     for (var r = 1; r <= rows; r++) {
26         var row = document.createElement("tr");
27         for (var c = 1; c <= cols; c++) {
28             var cell = document.createElement("td");

```

On the right, there is a preview pane titled '3yv32eabn' with a 'RUN' button. Below the preview is a table creation tool with 'Rows: 10' and 'Columns: 10' selected. A preview of the 10x10 grid is shown.

#### 4. Find the Age from input date.(Ex. 17 Yrs, 3 Monts,13 Days)

##### Source Code:

```

<!DOCTYPE html>

<html>

<body>

<label for="birthdate">Birthdate:</label>

<input type="date" id="birthdate">

<button onclick="calculateAge()">Calculate Age</button>

<p id="output"></p>

<script>

function calculateAge() {

    var birthdate = new Date(document.getElementById("birthdate").value);

    var today = new Date();

    var ageInMilliseconds = today - birthdate;

```

```

var ageInSeconds = ageInMilliseconds / 1000;

var ageInMinutes = ageInSeconds / 60;

var ageInHours = ageInMinutes / 60;

var ageInDays = ageInHours / 24;

var ageInMonths = ageInDays / 30.44;

var ageInYears = ageInMonths / 12;

var years = Math.floor(ageInYears);

var months = Math.floor(ageInMonths % 12);

var days = Math.floor(ageInDays % 30.44);

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

output.innerHTML = years + " Yrs, " + months + " Monts, " + days + " Days";

}

</script>

</body>

</html>

```

## Output:

The screenshot shows an online compiler interface at [onecompiler.com/html/3yyv32eabn](https://onecompiler.com/html/3yyv32eabn). The code has been run, and the output window displays the result: "19 Yrs, 1 Mouts, 7 Days". The input field above the output shows the date "09-12-2003".

```

index.html      styles.css      script.js      3yyv32eabn
NEW   HTML ▾  RUN ▶  ⋮  ⌂
Birthdate: 09-12-2003 | Calculate Age
19 Yrs, 1 Mouts, 7 Days

```

## 5. Find the No. of Days between two given dates

### Source code:

```
<!DOCTYPE html>
```

```
<html>

<body>

<label for="startdate">Start Date:</label>

<input type="date" id="startdate">

<label for="enddate">End Date:</label>

<input type="date" id="enddate">

<button onclick="calculateDays()">Calculate Days</button>

<p id="output"></p>

<script>

function calculateDays() {

    var startdate = new Date(document.getElementById("startdate").value);

    var enddate = new Date(document.getElementById("enddate").value);

    var timeDiff = Math.abs(enddate.getTime() - startdate.getTime());

    var diffDays = Math.ceil(timeDiff / (1000 * 3600 * 24));

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

    output.innerHTML = diffDays + " days";

}

</script>

</body>

</html>
```

**Output:**

The screenshot shows a web-based code editor on onecompiler.com. The code consists of three files: index.html, styles.css, and script.js. The script.js file contains a function to calculate the difference between two dates. The output window shows the result: 2192 days.

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <label for="startdate">Start Date:</label>
6 <input type="date" id="startdate">
7 <label for="enddate">End Date:</label>
8 <input type="date" id="enddate">
9 <button onclick="calculateDays()">Calculate Days</button>
10
11 <p id="output"></p>
12
13 <script>
14 function calculateDays() {
15     var startdate = new Date(document.getElementById("startdate").value);
16     var enddate = new Date(document.getElementById("enddate").value);
17     var timeDiff = Math.abs(enddate.getTime() - startdate.getTime());
18     var diffDays = Math.ceil(timeDiff / (1000 * 3600 * 24));
19
20     var output = document.getElementById("output");
21     output.innerHTML = diffDays + " days";
22 }
23 </script>
24
25 </body>
```

## Conclusion:

Threw loops and strings this all are used for various type of logic and output.

## Course Outcome:

This all type of method is used for good and efficient output.