# Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7AB**

# Lab 2: Javascript

**Date: 12 September, 2019**

**Time: 10:00-01:00pm & 02:00-05:00pm**

# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

# 

# Lab 2: Javascript

**Introduction**

JavaScript is a well-known scripting language which is widely used to handle behavior of a web site. Most of modern websites use JavaScript and JavaScript based frameworks to control dynamic behavior at the client side. Students have learned basic and advanced concepts of javascript during their previous semesters. This lab will help them to further revise and understand these concepts by practically using javascript in given situations.

**Objectives**

The objective of this lab is helping students to familiarize themselves with basic concepts of JavaScript by practically implementing them in a given situation. The knowledge, students already gained in their previous semesters will help students to develop and control dynamic behavior of a web page using JavaScript.

**Tools/Software Requirement**

Notepad, browser.

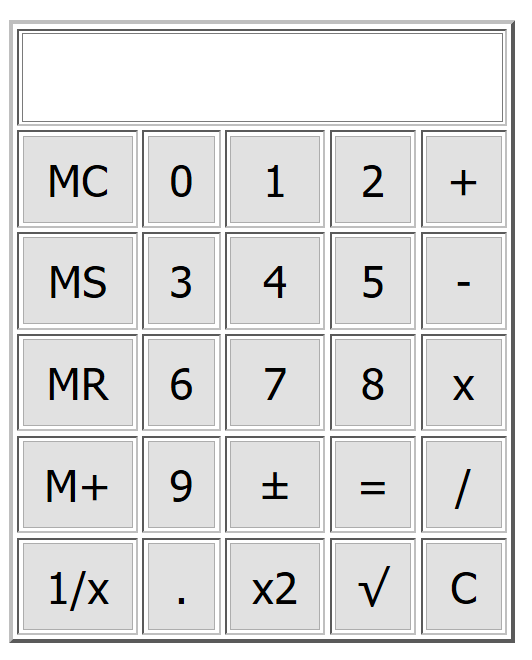
**Helping Material:**

W3Schools: <https://www.w3schools.com/js/default.asp>

**Lab Tasks**

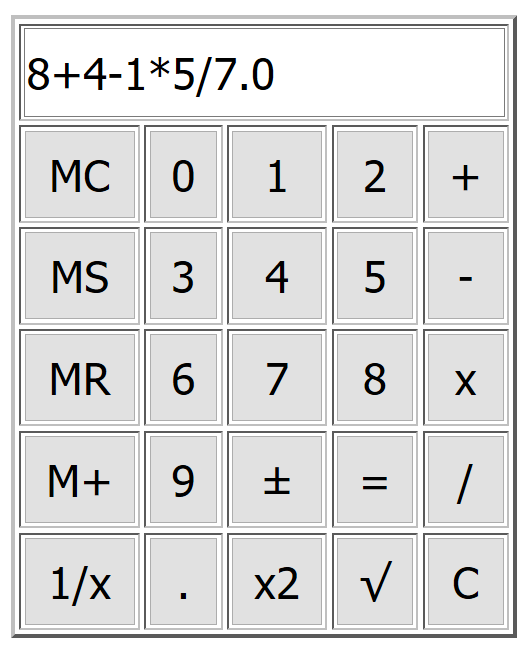
**Task**

Create a simple web based calculator as shown in the image below.



Following are the important functionalities of the calculator:

1. Whenever a user presses any number (0-9) or operations (+,-,x,/,±, .), it must be shown in the input text field (see example image below).



1. In case of ‘.’ button pressed, a ‘0’ must be added at the end of the string as shown in above image.
2. When ‘=’ button is pressed result is shown in the text field.
3. ‘C’ button must clear and reset everything (text field, any storage variables).
4. For a given input N, ‘1/x’ should give the results of 1/N (e.g. input =5, result = 0.20) of the given input.
5. For a given input N, ‘x2’ should square the input (e.g. input =5, result = 25).
6. The square root button, ,should calculate the square root of the input.
7. The ‘±’ button should add/remove a ‘–’ sign to the input value.
8. MS button must store the numeric input value written in the text field in a variable. In case of an equation as shown in the image above, it should not store anything.
9. MC button should clear stored numeric value.
10. MR button should recall the stored value and display it in the input text field.
11. M+ button should add the input value given in the text field in to the stored value in memory and save it as stored value.

**Hints:**

1. JS ‘OnClick’ event will be attached to all buttons in the calculator.
2. To handle value storage (MS, MC, MR, M+), create a global variable, M, and change its value according to the button pressed.
3. To display inputs as a sequence of numbers and arithmetic operation (e.g. 2+3-5+7/2.5), use string concatenation.
4. You can use JS eval() function to directly execute a string as a JS statement. See example below:

var input= “2+3-5+2.5/2.5”;

eval(input); // output: 1.0

1. Keep in mind point 4 on eval(), when “=” button is pressed just call var result = eval(document.getElementByID(“inputTextField”).value);

|  |
| --- |
| Solution |
| Task Code:  <!DOCTYPE html>  <html>  <head>  <title> 15 puzzle</title>  <style type="text/css">  table  {  border: 1px black solid;  width: 380px;  height: 480px;  table-layout: fixed;  }  td  {  border: 1px black solid;  background-color: #9CA1B5;  text-align: center;  font: 20px solid;  font-weight: bold;    }  #output  {  background-color:rgb(248, 248, 248);  text-align: right;  height: 60px;  }  </style>  </head>  <body>  <table >  <tr>  <td colspan="5" id = "output">  <div > </div>  </td>  </tr>  <tr id = "row1" >  <td id = "r1c1" class = "mem"> MC </td>  <td id = "r1c2" class = "digit">0</td>  <td id = "r1c3" class = "digit">1</td>  <td id = "r1c4" class = "digit">2</td>  <td id = "r1c5" class = "dOperator"> + </td>  </tr>  <tr id = "row2">  <td id = "r2c1" class = "mem"> MS</td>  <td id = "r2c2" class = "digit">3</td>  <td id = "r2c3" class = "digit">4</td>  <td id = "r2c4" class = "digit">5</td>  <td id = "r2c5" class = "dOperator"> - </td>  </tr>  <tr id = "row3">  <td id = "r3c1" class = "mem"> MR</td>  <td id = "r3c2" class = "digit">6</td>  <td id = "r3c3" class = "digit">7</td>  <td id = "r3c4" class = "digit">8</td>  <td id = "r3c5" class = "dOperator"> x </td>  </tr>  <tr id = "row4">  <td id = "r4c1" class = "mem">M+</td>  <td id = "r4c2" class = "digit">9</td>  <td id = "r4c3" class = "uOperator">±</td>  <td id = "r4c4" > =</td>  <td id = "r4c5" class = "dOperator"> / </td>  </tr>  <tr id = "row5">  <td id = "r5c1" class = "uOperator">1/x</td>  <td id = "r5c2" >.</td>  <td id = "r5c3" class = "uOperator">x2</td>  <td id = "r5c4" class = "uOperator">√</td>  <td id = "r5c5" >C</td>  </tr>  </table>  <script type="text/javascript">  document.addEventListener("click", press);  var output = document.getElementById("output");  var showText = "";  var memory = false;  var memoryVal = 0;  var operator = 0;  var end = true;  function print()  {  output.innerHTML = showText;  }  function press(e)  {    if(end && (e.target.className === "uOperator" || e.target.id === "r4c4" || e.target.className === "dOperator"))  {  alert("Enter some data first");  return;  }  else  {  end = false;  }    if(e.target.className == "uOperator" || e.target.className == "dOperator")  {  operator++;  }    if(e.target.className == "digit" || e.target.className == "dOperator")  {  showText += e.target.innerHTML;  print();    return;  }  switch(e.target.id)  {    case "r1c1":  {    memory = false;  memoryVal = 0;  reset();  print();  alert("Memory is cleared!");  break;  }    case "r2c1":  {  if(operator == 0)  {  memory = true;  memoryVal = parseInt(showText);  alert("Value saved!");  }  else{  alert("A number can be stored only!");  }  break;  }  case "r3c1":  {  if(memory)  {  showText += memoryVal;  print();    }  else{  alert("Memory empty!");  }  break;  }  case "r4c1":  {  if(operator == 0)  {  memory = true;  memoryVal += parseInt(showText);  alert("Number added in memory value");  }  else{  alert("Numbers can be added only.");  }  break;  }  case "r4c3":  {  if(operator == 1)  {  num = parseInt(showText);  num \*= -1;  showText = "" + num;  print();  operator = 0;  }  else{  alert("Numbers required only");  }  break;  }  case "r4c4":  {      result = eval(showText);    showText = "= " + result;  print();  operator = 0;  showText = "" + result;  break;  }  case "r5c1":  {  if(operator == 1)  {  result = 1 / parseInt(showText);  showText = "= " + result;  print();  showText = "" + result;  operator = 0;  }  else{  alert("Numbers required only");  }  break;  }  case "r5c2":  {  showText += ".0";  print();  operator = 0;  break;  }  case "r5c3":  {  if(operator == 1)  {  result = Math.pow(showText, 2);  showText = "= " + result;  print();  showText = "" + result;  operator = 0;  }  else{  alert("Numbers required only");  }  break;  }  case "r5c4":  {    if(operator == 1)  {  result = Math.sqrt(showText);  showText = "= " + result;  print();  showText = "" + result;  operator = 0;  }  else{  alert("Numbers required only");  }  break;  }  case "r5c5":  {  reset();  print();  break;  }  }    }  function reset()  {  end = true;  operator = 0;  showText = "";  }  </script>  </body>  </html>  Task Output Screenshot: |

### Deliverables

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to Ms. Ayesha Asif: [ayesha.asif@seecs.edu.pk](mailto:ayesha.asif@seecs.edu.pk).