**School of Digital Media & Infocomm Technology (DMIT)**

**ST2111 Mobile Application Development I**

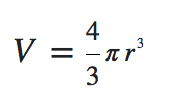
**Practical 10**

**DOM**

|  |
| --- |
| Objectives:  After completing this lab, you should be able to:   * Create programs which utilize the DOM |

**Exercise 1: Write Programs Using DOM**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical10-1** folder
3. Write a JavaScript program to calculate the volume of a sphere. Hint: Use document.getElementById(‘radius’) to get the value from the text field.



Formula for volume of a sphere

|  |  |
| --- | --- |
| <html>  <head>  <meta charset="utf-8">  <title>Volume of a Sphere</title>  <style>  body{padding-top:30px;}  label,input{display:block;}  </style>  </head>  <body>  <p>Input radius value and get the volume of a sphere.</p>  <form action="" method="post" id="MyForm">  <label for="radius">Radius</label><input type="text" name="radius" id="radius" required>  <label for="volume">Volume</label><input type="text" name="volume" id="volume">  <input type="submit" value="Calculate" id="submit"> </form>  </body>  </html>  container.html | Sample Output of the form |

**Exercise 2:**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical10-2** folder
3. Write a JavaScript program to do the following print First Name and Last Name contained in the form below.

|  |  |
| --- | --- |
| <html><head>  <meta charset=utf-8 />  <title>Return first and last name from a form - w3resource</title>  <style type="text/css">  body {margin: 30px;}  </style>  </head><body>  <form id="form1" onsubmit="getFormvalue()">  First name: <input type="text" name="fname" value="David"><br>  Last name: <input type="text" name="lname" value="Beckham"><br>  <input type="submit" value="Submit">  </form>  </body></html>  container.html | Sample Form Output  Output:  “Bill Clinton” |

**Sample Solution:**

function getFormvalue()

{

var x=document.getElementById("form1");

for (var i=0;i<x.length;i++)

{

if (x.elements[i].value!='Submit')

{

document.write(x.elements[i].value);

}

}

}

**Exercise 3:**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical10-3** folder
3. Write a JavaScript program to add a row to the table below. Set

|  |  |
| --- | --- |
| <html><head>  <meta charset=utf-8 />  <title>Insert row in a table - w3resource</title>  </head><body>  <table id="sampleTable" border="1">  <tr><td>Row1 cell1</td>  <td>Row1 cell2</td></tr>  <tr><td>Row2 cell1</td>  <td>Row2 cell2</td></tr>  </table><br>  <input type="button" onclick="addRow()" value="Insert row">  </body></html>  container.html | Sample Form Output |

**Sample Solution:**

function addRow()

{

var x=document.getElementById('sampleTable').insertRow(0);

var y = x.insertCell(0);

var z = x.insertCell(1);

y.innerHTML="New Cell1";

z.innerHTML="New Cell2";

}

**Explain the difference between insertRow and insertCell:**

|  |
| --- |
|  |

**Exercise 4:**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical10-4** folder
3. Write a JavaScript program to:
   1. Prompt the user for the number of rows
   2. Prompt the user for the number of columns
   3. Use a nested loop to create rows and columns for table **myTable**

|  |  |
| --- | --- |
| <html>  <head>  <meta charset=utf-8 />  <title>Create a table</title>  <style type="text/css">  body {margin: 30px;}  </style>  </head><body>  <table id="myTable" border="1">  </table><form>  <input type="button" onclick="createTable()" value="Create the table">  </form></body></html>  container.html | Prompt the user for number of rows    Prompt the user for number of columns    Sample Output |

**Exercise 5:**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical10-5** folder
3. Write a JavaScript program modify the style of the paragraph with id **text. Set** the following parameters:
   1. Font: **comic style MS**
   2. Font color: green
   3. Font size of 14**.**

|  |  |
| --- | --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset=utf-8 />  <title>JS DOM paragraph style</title>  </head>  <body>  <p id ='text'>JavaScript Exercises - w3resource</p>  <div>  <button id="jsstyle"  onclick="js\_style()">Style</button>  </div>  </body>  </html> | JavaScript Exercises - w3resource  Sample Output |

**Sample Solution:**

function js\_style()

{

text.style.fontSize = "14pt";

text.style.fontFamily = "Comic Sans MS";

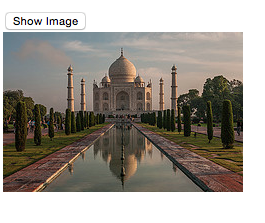
text.style.color = "green";

}

**Advanced Exercise 6 (Optional): Manipulating Images**

1. Create a copy of the **practical1-1** in the same **MAD1** folder
2. Rename the copied folder as **practical10-6** folder
3. The program below is done for you. It displays a random image when the user selects the button.

|  |  |
| --- | --- |
| <html>  <head>  <meta charset=utf-8 />  <title>Display a random image.</title>  <style type="text/css">  body {margin-top: 30px;}  </style>  </head>  <body>  <div>  <button id="jsstyle"  onclick="display\_random\_image();">Show Image</button>  </div>    </body>  </html>  container.html | function display\_random\_image()  {  var theImages = [{  src: "http://farm4.staticflickr.com/3691/11268502654\_f28f05966c\_m.jpg",  width: "240",  height: "160"  }, {  src: "http://farm1.staticflickr.com/33/45336904\_1aef569b30\_n.jpg",  width: "320",  height: "195"  }, {  src: "http://farm6.staticflickr.com/5211/5384592886\_80a512e2c9.jpg",  width: "500",  height: "343"  }];    var preBuffer = [];  for (var i = 0, j = theImages.length; i < j; i++) {  preBuffer[i] = new Image();  preBuffer[i].src = theImages[i].src;  preBuffer[i].width = theImages[i].width;  preBuffer[i].height = theImages[i].height;  }    // create random image number  function getRandomInt(min,max)  {  // return Math.floor(Math.random() \* (max - min + 1)) + min;    imn = Math.floor(Math.random() \* (max - min + 1)) + min;  return preBuffer[imn];  }  // 0 is first image, preBuffer.length - 1) is last image    var newImage = getRandomInt(0, preBuffer.length - 1);    // remove the previous images  var images = document.getElementsByTagName('img');  var l = images.length;  for (var p = 0; p < l; p++) {  images[0].parentNode.removeChild(images[0]);  }  // display the image  document.body.appendChild(newImage);  }  script.js |



**Sample Output**

**Explain the difference between theImages Array and preBuffer Array:**

|  |
| --- |
|  |

**Explain the purpose of the statement:** images[0].parentNode.removeChild(images[0])

|  |
| --- |
|  |

**Exercise 7: Run Exercises in the Android Simulator**

You may run your exercises using the Phonegap Android Simulator to view the output. For example, if you wish to test out **practical9-1**, do the following:

1. From the [Start screen,](http://www.computerhope.com/jargon/w/windows8.htm) click **Command Prompt** to open the Windows console.
2. Navigate to **MAD1** folder. (Note: If your MAD1 folder is in **c: drive**, type **cd\** to bring you to the root directory first, then type **cd mad1** to go to MAD1 folder. If your MAD1 folder is in **d: drive**, type **d:** to change to d: drive first, then type **cd mad1** to go to MAD1 folder.**)**
3. Create the new app by typing **phonegap create practical10-1app**and press enter.
4. Open a text editor and open the file located in **mad1\practical10-1app\www\index.html**.
5. Replace the text with the code in **Listing D** below.
6. Copy **script.js** from **practical9-1** to the **www** directory.
7. Type **phonegap run android**.
8. Your JavaScript program is run in the Android Simulator.

**Listing A. index.html**