LAB # 07

Statement Purpose:

To familiarize the students with

- ⊡ Regular Expression
- ⊡ Document Object Model
  - Accessing a Node
  - Changing Node structure
  - DOM Coordinate

Form validation?

Form validation is the process of making sure that data supplied by the user using a form, meets the criteria set for collecting data from the user. For example, if you are using a registration form, and you want your user to submit name, email id and address, you must use a code (in JavaScript or in any other language) to check whether the user entered a name containing alphabets only, a valid email address and a proper address.

Checking non-empty field

Javascript function to check whether a field is empty or not

```
// If the length of the element's string is 0 then display helper message
  function required(inputtx)
  {
    if (inputtx.value.length == 0)
     {
       alert("message");
       return false;
     }
     return true;
   }
```

At first the function required() will accept the HTML input value through inputtx parameter. After that length property of string, object is used to get the length of the said parameter. If the length of value.inputtx is 0 then it returns false otherwise true.

HTML Code

```
<!Doctype html>
```

```
<html lang="en">
<head>
	<meta charset="utf-8">
	<title>JavaScript form validation - checking non-empty</title>
	<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body>
	<div class="mail">
		<h2>Input your Name and Submit</h2>
		<form name="form1" action="#" onsubmit="required()">
		<ul>
			<li><input type='text' name ='text1'/></li>
			<li class="rq">*Required Field</li>
			<li><input type="submit" name="submit" value="Submit" /></li>
		</ul>
		</form>
	</div>
	<script src="non-empty.js"></script>
</body>
</html>
```

JavaScript Code

```
function required()
{
	var empt = document.forms["form1"]["text1"].value;
	if (empt == "")
	{
		alert("Please input a Value");
		return false;
	}
	else
	{
		alert('Code has accepted : you can try another');
		return true;
	}
}
```

Checking non-empty field

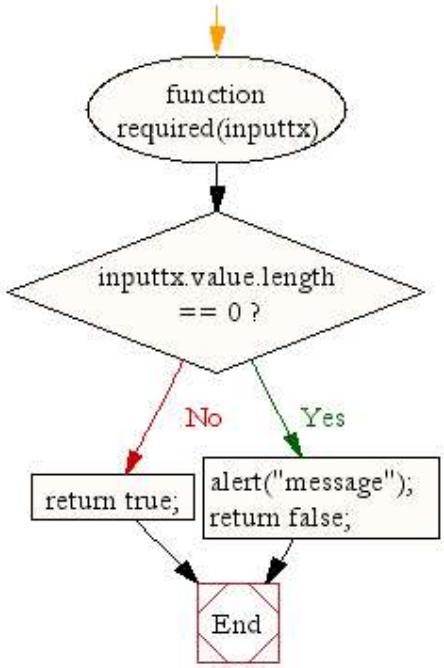
Often, situations arise when a user should fill a single or more than one field in an HTML form before they submit it. You can write a JavaScript form validation script to check whether the required field(s) in the HTML form is blank or not. The following function can be used to check whether the user has entered anything in a given field. Blank fields indicate two kinds of values. A zero-length string or a NULL value.

Javascript function to check whether a field is empty or not

```
// If the length of the element's string is 0 then display helper message
  function required(inputtx)
  {
    if (inputtx.value.length == 0)
     {
        alert("message");
        return false;
     }
     return true;
   }
```

At first the function required() will accept the HTML input value through inputtx parameter. After that length property of string, object is used to get the length of the said parameter. If the length of value.inputtx is 0 then it returns false otherwise true. Here is the complete web document.

Flowchart:



HTML Code

```
<!Doctype html>
<html lang="en">
<head>
	<meta charset="utf-8">
	<title>JavaScript form validation - checking non-empty</title>
	<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body>
	<div class="mail">
	<h2>Input your Name and Submit</h2>
		<form name="form1" action="#" onsubmit="required()">
		<ul>
			<li><input type='text' name ='text1'/></li>
			<li class="rq">*Required Field</li>
```

```
                    <li><input type="submit" name="submit" value="Submit" /></li>
                </ul>
                </form>
        </div>
        <script src="non-empty.js"></script>
        </body>
        </html>
```

JavaScript Code

```
function required()
{
        var empt = document.forms["form1"]["text1"].value;
        if (empt == "")
        {
                alert("Please input a Value");
                return false;
        }
        else
        {
                alert('Code has accepted : you can try another');
                return true;
        }
}
```

CSS Code

```
li {list-style-type: none;
font-size: 16pt;
}
.mail {
margin: auto;
```

```
padding-top: 10px;
padding-bottom: 10px;
width: 400px;
background : #D8F1F8;
border: 1px soild silver;
}
.mail h2 {
margin-left: 38px;
}
input {
font-size: 20pt;
}
input:focus, textarea:focus{
background-color: lightyellow;
}
input submit {
font-size: 12pt;
}
.rq {
color: #FF0000;
font-size: 10pt;
}
```

Document Object Model

Accessing Node

setAttribute()

The **setAttribute()** method is used to set or add an attribute to a particular element and provides a value to it. If the attribute already exists, it only set or changes the value of the attribute.

Syntax

element.setAttribute(attributeName, attributeValue)

Sample code 1:

Adding a href attribute with a value of "https://www.google.com/" to the <a> tag with id = "link".

```
<!DOCTYPE html>
```

```
<html>
<head>
   <title> JavaScript setAttribute() method </title>
   <script>
      function fun() {
         document.getElementById("link").setAttribute("href",
"https://www.google.com/");
      }
   </script>
</head>
<body style="text-align: center;">
   <h2> It is an example of adding an attribute using the setAttribute() method. </h2>
   <a id="link"> google.com </a>
   <p> Click the following button to see the effect. </p>
   <button onclick="fun()"> Add attribute </button>
</body>
</html>
```

Sample code 2:

Updating the value of an existing attribute using the setAttribute() method. Converting a textfield to a button by changing the value of type attribute from text to button.

```
<html>
<head>
        <title> JavaScript setAttribute() method </title>
        <script>
        function fun()
        {
        document.getElementById("change").setAttribute("type", "button");
        }
        </script>
</head>
```

```
<body style = "text-align: center;">
        <h2> It is an example to update an attribute's value using the setAttribute() method. </h2>
        <input id = "change"  type = "text" value = "Google"/>
        <p> Click the follwing button to see the effect. </p>
        <button onclick = "fun()"> Change </button>
</body>
</html>
```

getAttribute() method

The **getAttribute()** method is used to get the value of an attribute of the particular element. If the attribute exists, it returns the string representing the value of the corresponding attribute. If the corresponding attribute does not exist, it will return an empty string or null.

Syntax

element.getAttribute(attributename)

Sample Code 1:

Two div elements with id div1 and div2, each having style attribute. We are getting the value of style attribute by using the getAttribute() method. Click the given button to get the value of the style attribute of given div elements.

```
<!DOCTYPE html>
<html>

<head>
   <title>
      The getAttribute Method
   </title>
</head>

<body>
   <h1>
      Full Stack Web Development
   </h1>
```

```
    <h2>
        getAttribute() Method
    </h2>

    <div id="div1" style="background-color: yellow; font-size: 25px; color: red; border: 2px solid red;">
        This is first div element.
    </div>
    <br>
    <div id="div2" style="background-color: lightblue; font-size: 25px; color: blue; border: 2px solid blue;">
        This is second div element.
    </div>
    <br>
    <button onclick="fun()">
Click me!
</button>
    <p id="p"></p>
    <p id="p1"></p>
    <script>
        function fun() {
            var val = document.getElementById("div1").getAttribute("style");
            document.getElementById("p").innerHTML = val;
            var val1 = document.getElementById("div2").getAttribute("style");
            document.getElementById("p1").innerHTML = val1;
        }
    </script>
</body>
</html>
```

Sample code 2

Get the value of onclick **attribute of the button element. In this example, we are extracting the value of** onclick **attribute and the value of** href **attribute. There is an anchor element with the** href **attribute.**

```
<!DOCTYPE html>
<html>


<head>
   <title>
      The getAttribute Method
   </title>
</head>
<body>
   <h1>
      Get Attribute
   </h1>
   <h2>
      getAttribute() Method
   </h2>
   <div id="div1" style="background-color: yellow; font-size: 25px; color: red; border: 2px solid red;">
      This is the div element.
   </div>
   <br>
   <a href="http://www.google.com/" id="link"> goolge.com </a>
   <br><br>
   <button onclick="fun()" id="btn">
Click me!
</button>
   <p id="p"></p>
   <p id="p1"></p>
   <script>
      function fun() {
```

```
        var val = document.getElementById("btn").getAttribute("onclick");
        document.getElementById("p").innerHTML = val;
        var val1 = document.getElementById("link").getAttribute("href");
        document.getElementById("p1").innerHTML = val1;
    }
  </script>
</body>
</html>
```

DOM Mutating operations

Change an <img> tag attribute

Syntax

Return the src property:

imageObject.src

Set the src property:

imageObject.src = URL

```
<!DOCTYPE html>
<html>
<body>
  <img id="myImg" src="comp.gif" width="107" height="98">
  <p>Click the button to change the value of the src attribute of the image.</p>
  <button onclick="myFunction()">Try it</button>
  <script>
    function myFunction() {
      document.getElementById("myImg").src = "compman.gif";
    }
  </script>
</body>
</html>
```

Changing the node structure

```
!DOCTYPE html>
```

```
<html>
<body>
	<h1>The Document Object</h1>
	<h2>The createElement() Method</h2>
	<p>Create a p element with some text:</p>
	<script>
	// Create element:
		const para = document.createElement("p");
		para.innerText = "This is a paragraph.";
	// Append to body:
		document.body.appendChild(para);
	</script>
</body>
</html>
```

Sample code 2:

```
<!DOCTYPE html>
<html>
<body>
	<h1>The Element Object</h1>
	<h2>The insertBefore() Method</h2>
	<ul id="myList1">
		<li>Coffee</li>
		<li>Tea</li>
	</ul>
	<ul id="myList2">
		<li>Water</li>
		<li>Milk</li>
		<li>Juice</li>
	</ul>
```

```
<p>Click the button to move items from one list to another:</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
  const node = document.getElementById("myList2").lastElementChild;
  const list = document.getElementById("myList1");
  list.insertBefore(node, list.children[0]);
}
</script>
</body>
</html>
```