Web System & Technology

Lecture # 26

Ehtisham Rasheed

Department of Computer Science
University of Gurjat, Gujrat

UNIVERSITY OF GUJRAT

UOG

What is JavaScript?

- JavaScript is a client-side programming language of the web

- Used to create dynamic web pages

- It can update and change both HTML and CSS

- It can calculate, manipulate and validate data

JavaScript can change HTML Content

- One of many JavaScript HTML methods is getElementById()

- Example

```html
<p id="demo">JavaScript can change HTML content.</p>

<button type="button" onclick='document.getElementById("demo").innerHTML = "Hello JavaScript!"'>Click Me!</button>
```

- JavaScript accepts both single and double quotes

```javascript
document.getElementById('demo').innerHTML = 'Hello JavaScript';
```

3

JavaScript Can Change HTML Attribute Values

```
<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn on the
light</button>

<img id="myImage" src="pic_bulboff.gif" style="width:100px">

<button onclick="document.getElementById('myImage').src='pic_bulboff.gif'">Turn off the
light</button>
```

More on JavaScript

```html
<p id="demo">JavaScript can change the style of an HTML element.</p>

<button type="button"
onclick="document.getElementById('demo').style.fontSize='35px'">Click Me!</button>
```

```html
<p id="demo">JavaScript can hide HTML elements.</p>

<button type="button"
onclick="document.getElementById('demo').style.display='none'">Click Me!</button>
```

```html
<p id="demo" style="display:none">Hello JavaScript!</p>

<button type="button"
onclick="document.getElementById('demo').style.display='block'">Click Me!</button>
```

The <script> Tag

```
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

JavaScript Functions and Events

- A JavaScript function is a block of JavaScript code, that can be executed when called

- For example, a function can be called when an **event** occurs, like when the user clicks a button

- Script can be placed in the <body>, or in the <head> section of an HTML page, or in both

JavaScript Functions and Events

```html
<head>
<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>

<h2>Demo JavaScript in Head</h2>

<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
```

External JavaScript

- Scripts can be placed in external files

- External scripts are practical when the same code is used in many different web pages

- JavaScript files have the file extension **.js**

- To use an external script, put the name of the script file in the src (source) attribute of a <script> tag

```
<script src="myScript.js"></script>
```

JavaScript Output

- JavaScript can display data in different ways:
  - Writing into an HTML element, using innerHTML or innerText
  - Writing into the HTML output using document.write()
  - Writing into an alert box, using window.alert()
  - Writing into the browser console, using console.log()

Using InnerHTML

```html
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "<h2>Hello World</h2>";
</script>
```

Using InnerText

```
<p id="demo"></p>

<script>
document.getElementById("demo").innerText = "Hello World";
</script>
```

Use innerHTML when you want to change an HTML element.

Use innerText when you only want to change the plain text.

Using document.write()

- For testing purpose, it is convenient to use document.write()

- Using document.write() after an HTML document is loaded, will delete all existing HTML

Using document.write()

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```

Using document.write()

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p>My first paragraph.</p>


<button type="button" onclick="document.write(5 + 6)">Try it</button>


</body>
</html>
```

Using window.alert()

```
<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>
```

Using console.log()

- For debugging purpose, you can call the console.log() method in the browser to display data

- F12 on keyboard will activate debugging

- Then select "Console" in the debugging menu

```
<body>

<script>
console.log(5 + 6);
</script>

</body>
```

Hands on Lab

- Complete Hands on lab 1 and complete three assignments