Lecture # 7

JavaScript

Advance Web Designing and Development

Ms. Hafiza Alia
(alia@theprotec.com)

Get value from input elements

- The value property sets or returns the value of the value attribute of a text field.

- The following code will change the value to Abc
  - document.getElementById("myText").value="Abc";
- The following code will get the value to
  - document.getElementById("myText").value;

An example that shows the difference between the defaultValue and value property:

- Name: <input type="text" id="myText" value="Mickey">

```
var x = document.getElementById("myText");
var defaultVal = x.defaultValue;
var currentVal = x.value;
```

JavaScript Events

- HTML events are "things" that happen to HTML elements.

- When JavaScript is used in HTML pages, JavaScript can "react" on these events.

HTML Events

- An HTML event can be something the browser does, or something a user does.

  - ❑ An HTML web page has finished loading
  - ❑ An HTML input field was changed
  - ❑ An HTML button was clicked

JavaScript Events

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Onclick

```
<button onclick="myFunction()">Click me</button>
<p id="demo"></p>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Hello World";
}
</script>
```

Onclick

```
<button onclick=“func()">The time is?</button>

<p id=“demo”>  </p>
<script>
Function func(){
}
</scripy>
```

Example
Get the value of a text field:

```
First Name: <input type="text" id="myText" value="Mickey">
<button onclick="myFunction()">Try it</button>
function myFunction() {
    var x = document.getElementById("myText").value;
    document.getElementById("demo").innerHTML = x;
}
```

Syntax

- In HTML

```
p>This example demonstrates how to assign an "onclick" event to a p element.</p>
<p id="demo" onclick="myFunction()">Click me.</p>

<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "YOU CLICKED ME!";
}
</script>
```

Examples

-
  ```
  <h1 onclick="this.innerHTML = 'Ooops!'">Click on this text!</h1>
  ```

-
  ```
  <h1 onclick="changeText()">Click on this text!</h1>

  <script>
  function changeText(x) {
  ```

-
  ```
  Document.get
      x.innerHTML = "Ooops!";
  }
  </script>
  ```

Assign Events Using the HTML DOM

```
<button id="myBtn">Try it</button>
<p id="demo"></p>
document.getElementById("myBtn").onclick = displayDate();

function displayDate() {
    document.getElementById("demo").innerHTML = Date();
}
```

onchange Event

- Execute a JavaScript when a user changes the selected option of a <select> element:

```
<select id="mySelect" onchange="myFunction()">
</select>

function myFunction() {
    var x = document.getElementById("mySelect").value;
    document.getElementById("demo").innerHTML = "You selected: " + x;
}
```

JavaScript Strings

- A JavaScript string simply stores a series of characters like "John Doe".
- A string can be any text inside quotes. You can use single or double quotes:

String Length

- By using .Lenght keyword we can get the length of a string.
- **Example:**
- <script>
- var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
- document.getElementById("demo").innerHTML = txt.length;
- </script>

Special Characters

- var y = ”I said “ ssfsafD”."

Code	Outputs
\'	single quote
\"	double quote
\\	backslash

Strings Can be Objects

- <script>
- var x = "John";   // x is a string
- var y = new String(“sjahfjsahf”); // y is an object
- Y=“fasdnf”;
- document.getElementById("demo").innerHTML =
- typeof x + "<br>" + typeof y;
- </script>

Cont..

- When using the == operator, equal strings are equal:
- When using the === operator, equal strings are not equal, because the === operator expects equality in both type and value.
- <script>
- var x = "John";   // x is a string
- var y = new String("John");   // y is an object
- document.getElementById("demo").innerHTML = (x==y);
- document.getElementById("demo").innerHTML = (x===y);
- </script>

JavaScript String Methods

- **String Length (.length)**
- The **length** property returns the length of a string:
- **Finding a String in a String(indexof)**
  - The **indexOf()** method returns the index of (the position of) the **first** occurrence of a specified text in a string:
  - The **lastIndexOf()** method returns the index of the **last** occurrence of a specified text in a string.
  - Both the indexOf(), and the lastIndexOf() methods return -1 if the text is not found.

String Length (.length) Example

- <script>
- var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
- document.getElementById("demo").innerHTML = txt.length;
- </script>

Index of method Example

```
<script>
function myFunction() {
    var str =“String Data”;
    var pos = str.indexOf("locate");
    document.getElementById("demo").innerHTML = pos;
}
</script>
```

Last index of method Example

```
 <script>
function myFunction() {
    var str =“String Data”;
    var pos = str.lastIndexOf(“Data");
    document.getElementById("demo").innerHTML = pos;
}
</script>
```

String Methods

- **Searching for a String in a String**
  - The **search()** method searches a string for a specified value and returns the position of the match:
- **Extracting String Parts**
  - **There are three methods for extracting.**
    - slice(start, end)
    - substring(start, end)
    - substr(start, length)

methods, indexOf() and search(),

- They accept the same arguments (parameters), and return the same value?
- The two methods are quite equal.
- These are the differences:
  - The search() method cannot take a second start position argument.

search() Method Example

- The two methods, indexOf() and search(), are equal.
- They accept the same arguments (parameters), and they return the same value

```
<script>
function myFunction() {
    var str = document.getElementById("p1").innerHTML;
    var pos = str.search("locate");
    document.getElementById("demo").innerHTML = pos;
}
</script>
```

Extracting String Parts Example

- **slice(start, end)**
  - <script>
  - var str = "Apple, Banana, Kiwi";
  - document.getElementById("demo").innerHTML = str.slice(7,13);
  - </script>
  - If a parameter is negative, the position is counted from the end of the string.
  - <script>
  - var str = "Apple, Banana, Kiwi";
  - document.getElementById("demo").innerHTML = str.slice(-12,-6);
  - </script>

Slice method

- If you omit the second parameter, the method will slice out the rest of the string:
- var res = str.slice(7);
- var res = str.slice(-12);

substring(start, end)

- **substring()** is similar to slice().
- The difference is that substring() cannot accept negative indexes.
- **Example**
- <script>
- var str = "Apple, Banana, Kiwi";
- document.getElementById("demo").innerHTML = str.substring(7,13);
- </script>

The substr(start,length) Method

- **substr()** is similar to slice().
- The difference is that the second parameter specifies the **length** of the extracted part.
- **Example**
- var str = "Apple, Banana, Kiwi";
  var res = str.substr(7, 6);

Replacing String Content

- The **replace()** method replaces a specified value with another value in a string:

```
<script>
function myFunction() {
    var str = document.getElementById("demo").innerHTML;
    var txt = str.replace("Microsoft","W3Schools");
    document.getElementById("demo").innerHTML = txt;
}
</script>
```

Replace() con..

- The replace() method can also take a regular expression as the search value.
- By default, the replace() function replaces only the first match. To replace all matches, use a regular expression with a g flag (for global match):

```
<button onclick="myFunction()">Try it</button>
<p id="demo">Please visit Microsoft and Microsoft!</p>
<script>
function myFunction() {
    var str = document.getElementById("demo").innerHTML;
    var txt = str.replace(/Microsoft/g,"W3Schools");
    document.getElementById("demo").innerHTML = txt;
}
</script>
```

Converting to Upper and Lower Case

- A string is converted to upper case with **toUpperCase()**:
  - <script>
  - function myFunction() {
  -     var text = document.getElementById("demo").innerHTML;
  -     document.getElementById("demo").innerHTML = text.toUpperCase();
  - }
  - </script>
- A string is converted to lower case with **toLowerCase()**:
  - var text1 = "Hello World!";   // String
    var text2 = text1.toLowerCase(); // text2 is text1 converted to lower

The concat() Method

- The **concat()** method can be used instead of the plus operator. These two lines do the same:
- var text = "Hello" + " " + "World!";
- var text = "Hello".concat(" ", "World!");

- **Note:**
- All string methods return a new string. They don't modify the original string. Formally said: Strings are immutable: Strings cannot be changed, only replaced.

Extracting String Characters

- There are 2 **safe** methods for extracting string characters:
  - charAt(position)
    - <script>
    - var str = "HELLO WORLD";
    - document.getElementById("demo").innerHTML = str.charAt(0);
    - </script>
  - charCodeAt(position)
    - <script>
    - var str = "AELLO WORLD";
    - document.getElementById("demo").innerHTML = str.charCodeAt(0);
    - </script>

Syntax (click,mouseover,mouseout)

In JavaScript, using the addEventListener() method:

```
<p id="demo">Click me.</p>
<script>
document.getElementById("demo").addEventListener("click", myFunction);
function myFunction() {
    document.getElementById("demo").innerHTML = "YOU CLICKED ME!";
}
</script>
```

JavaScript
End Of Lecture 7