WEB ENGINEERING
JAVA SCRIPT

Lecture 5

JAVASCRIPT

JavaScript is the programming language of HTML and for the Web

JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997

ECMA-262 is the official name. ECMAScript 6 (released in June 2015) is the latest official version of JavaScript

JAVASCRIPT

JavaScript is a cross-platform, object-oriented scripting language. It is a small and lightweight language.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements.

JAVASCRIPT

Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects; for example:

Client-side JavaScript extends the core language by supplying objects to control a browser and its Document Object Model (DOM).

> For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.

Server-side JavaScript extends the core language by supplying objects relevant to running JavaScript on a server.

> For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

JAVASCRIPT

Why Study JavaScript?

JavaScript is one of the **3 languages** all web developers **must** learn:

1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript to program the behavior of web pages

JAVASCRIPT CAN CHANGE HTML CONTENT

One of many HTML methods is **getElementById()**.
This example uses the method to "find" an HTML element (with id="demo"), and changes the element content (**innerHTML**) to "Hello JavaScript":

```
<html>
<body>
<h1>What Can JavaScript Do?</h1>
<p id="demo">JavaScript can change HTML content.</p>
<button type="button"
onclick="document.getElementById('demo').innerHTML =
'Hello JavaScript!'">
Click Me!</button>
</body>
</html>
```

JAVASCRIPT CHANGES HTML ATTRIBUTES

```
<body>
<h1>JavaScript Can Change Images</h1>
<img id="myImage" onclick="changeImage()"
src="pic_bulboff.gif" width="100" height="180">
<p>Click the light bulb to turn on/off the light.</p>
<script>
function changeImage() {
    var image = document.getElementById('myImage');
    if (image.src.match("bulbon")) {
        image.src = "pic_bulboff.gif";
    } else {
        image.src = "pic_bulbon.gif";    }    }
</script>
</body>
```

JAVASCRIPT CHANGES HTML STYLES (CSS)

```
<body>
<h1>What Can JavaScript Do?</h1>
<p id="demo">JavaScript can change the style of an HTML element.</p>
<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
<button type="button" onclick="myFunction()">Click Me!</button>
</body>
```

JAVASCRIPT CAN VALIDATE DATA

```
<body>
<h1>JavaScript Can Validate Input</h1>
<p>Please input a number between 1 and
10:</p>
<input id="numb" type="number">
<button type="button"
onclick="myFunction()">Submit</button>
<p id="demo"></p>
<script>
function myFunction() {
    var x, text;
    // Get the value of the input field with
    //id="numb"
    x =

document.getElementById("numb").value;
    // If x is Not a Number or less than one or
    //greater than 10
    if (isNaN(x) || x < 1 || x > 10) {
        text = "Input not valid";
    } else {
        text = "Input OK";
    }
    document.getElementById("demo").innerHTML =
    text;
}
</script>
</body>
```

JAVASCRIPT <SCRIPT> TAG

Older examples may use a type attribute:

<script *type="text/javascript"*>

> The type attribute is not required. JavaScript is the default scripting language in HTML.

```
<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 the function is called or invoked.

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

A JavaScript function is placed in the <head> section or in <body> section of an HTML page.

- The function is invoked (called) when a button is clicked

JAVASCRIPT FUNCTIONS AND EVENTS

```
<head>
    <script>
    function myFunction() {
        document.getElementById("demo").innerHTML =
    "Paragraph changed.";
    }
    </script>
</head>
<body>
    <h1>JavaScript in Head</h1>
    <p id="demo">A Paragraph.</p>
    <button type="button" onclick="myFunction()">Try
    it</button>
</body>
```

JAVASCRIPT EXTERNAL

Scripts can also 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 the <script> tag:

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

JAVASCRIPT EXTERNAL ADVANTAGES

Placing JavaScript in external files has some advantages:

1. It separates HTML and code
2. It makes HTML and JavaScript easier to read and maintain
3. Cached JavaScript files can speed up page loads

JAVASCRIPT OUTPUT

JavaScript can "display" data in different ways:

1. Writing into an alert box,
   using **window.alert()**.
2. Writing into the HTML output
   using **document.write()**.
3. Writing into an HTML element,
   using **innerHTML**.
4. Writing into the browser console,
   using **console.log()**.

JAVASCRIPT WINDOW.ALERT()

```
<html>
<body>

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

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

</body>
</html>
```

JAVASCRIPT USING DOCUMENT.WRITE()

```
<html>
<body>

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

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

</body>
</html>
```

JAVASCRIPT USING INNERHTML

```
<html>
<body>

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

<script>
document.getElementById("demo").innerHTML =
5 + 6;
</script>

</body>
</html>
```

JAVASCRIPT CONSOLE.LOG()

```
<html>
<body>

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

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

</body>
</html>
```

JAVASCRIPT SYNTAX

JavaScript **syntax** is the set of rules, how JavaScript programs are constructed.

JavaScript Programs

1. A **computer program** is a list of "instructions" to be "executed" by the computer. JavaScript statements are composed of: Values, Operators, Expressions, Keywords, and Comments.
2. In a programming language, these program instructions are called **statements**.
3. JavaScript is a **programming language**.
4. JavaScript statements are separated by **semicolons**.

JAVASCRIPT SYNTAX

```
<body>
<h1>JavaScript Statements</h1>
<p>Statements are separated by semicolons.</p>
<p>The variables x, y, and z are assigned the values 5, 6, and 11:</p>
<p id="demo"></p>
<script>
    var x = 5;
    var y = 6;
    var z = x + y;
    document.getElementById("demo").innerHTML = z;
</script>
</body>
```

JAVASCRIPT VALUES

The JavaScript syntax defines two types of values: Fixed values and variable values.

1. Fixed values are called **literals**.
2. Variable values are called **variables**.

JAVASCRIPT KEYWORDS

JavaScript **keywords** are used to identify actions to be performed.

The **var** keyword tells the browser to create a new variable:

```
var x = 5 + 6;
var y = x * 10;
```

Other are

char, **String**, **Array**, **toString**, **NaN**, **valueOf**, and more…

JAVASCRIPT COMMENTS

Not all JavaScript statements are "executed".

Code after double slashes **//** or
between **/*** and ***/** is treated as a **comment**.

Comments are ignored, and will not be executed:

```
var x = 5;   // I will be executed
// var x = 6;   I will NOT be executed
```

JAVASCRIPT IDENTIFIERS

Identifiers are names.

In JavaScript, identifiers are used to name variables (and keywords, and functions, and labels).

In JavaScript, the first character must be a letter, an underscore (_), or a dollar sign ($).

- Subsequent characters may be letters, digits, underscores, or dollar signs.

JAVASCRIPT IS CASE SENSITIVE

All JavaScript identifiers are **case sensitive**

The variables **lastName** and **lastname**, are two different variables.

```
lastName = "Doe";
lastname = "Peterson";
```

JAVASCRIPT CHARACTER SET

JavaScript uses the **Unicode** character set.

Unicode covers (almost) all the characters, punctuations, and symbols in the world

End of Lecture

Q & A