# Week 1 – Second Meeting: Online

Topics covered:

1. Introduction to HTML
2. Introduction to CSS
3. Introduction to JavaScript

## 1. Introduction to HTML

HTML, or the Hyper Text Markup Language, is a *programming language* that describes a Web page. HTML is made up of a series of *tags* which describe the content that is enclosed within them. For example, the tag set <html> </html> indicates that everything included between them is to be considered as HTML content. HTML documents contain tags, plain text, and can also include images.

There are three basic parts in an HTML document:

- The HTML declaration statement
- The header
- The body

The HTML declaration section should include two statements. The first one, which states that the document is of HTML type, is <!DOCTYPE html>. This tag is used when version 5 of HTML is used. When using version 4.01 the required tag is: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">.

The second tells the calling application where to *start looking for* HTML content, <html> </html>. Notice that the DOCTYPE declaration does not require a closing tag, but most other tags do necessitate a closing argument.

Let's examine the HTML 4.01 DOCTYPE statement a bit further. W3C is the World Wide Web consortium. You can visit their Web at www.w3.org. This organization is an international community that develops standards for the Web. The DTD part is the Document Type Definition, which is used to define an XML document. XML is another markup language, similar to HTML, which also describes a document. XML stands for Extensible Markup Language, and is used in a plethora of documents, from your typical Microsoft Word to PDF. "Transitional" indicates that the document can include all HTML elements and attributes, including deprecated and presentational ones. The EN part states that the content including should be interpreted using the English language. The URL, or Web Address, at the end states where the document definition can be found.

The header section includes other information about the document. The section is delimited by the <head> and </head> tags. In this area instructions such as

title, meta-tags, styles, links, and scripts. This section, when used, must always be included before the body section. Visit W3Schools
http://www.w3schools.com/tags/tag_head.asp
for a detailed description of the head tag and the elements that can included within it.

The third section includes the elements used in the visible area of a Web page. This section is enclosed by the <body> </body> tags. The body tag can also include more specific information, such as background and background color for example. In order to set the background color for a Web page (*please note that this element is deprecated, which means that it should not be used in this tag, but will be rendered by a browser*) would be <body bgcolor="#0000FF">. The set of letters and numbers is the hexadecimal code for the color, in this case for the color blue.

So, what is the fuzz about *deprecated*? The word implies that an element should not be used because it is being phased out. This means that the standard has changed and another method should be used to obtain the desired result. For example, the *bgcolor* element mentioned above should instead be indicated inside a CSS definition. CSS stands for Cascading Style Sheets, and it refers to the "new" manner in which the look and feel of the displayed content should be described. Hence, deprecated means that "it can be used, but you should not". Furthermore, in HTML version 5 deprecated elements are not allowed.

Visit the HTML introduction at the W3Schools Web, using this link.
http://www.w3schools.com/html/html_intro.asp
You can even try using it in their Web, by clicking on the *Try It Yourself* buttons.

## 2. Introduction to CSS

CSS, short for Cascading Style Sheets, define how elements inside the body tags should be displayed. When using CSS, some standard HTML tags are “re-defined” in order to *behave* just the way you want them to. For example, the <p> tag tells the browser that a paragraph is next. By defining how paragraphs are displayed using CSS, the same tag can be rendered in different ways. Instead of using additional tags in the document’s body (like center), the <p> is defined inside a CSS file. CSS separates the styles from the content, bringing a degree of modularity to HTML. Using CSS files also allows the designer to make changes to a single file that will affect all Web pages that reference it. CSS can also be defined in the header section, but doing so will implicate that every document must be changed when a different style is to be used.

After opening the <head> tag, a CSS style can be defined in the following manner:

```
<style type="text/css">
h1 {color:red;}
p {color:blue;}
</style>
```

This definition states that the text enclosed within <h1> and </h1> tags will be rendered in the color red, and that all text in a paragraph inside the <p> </p> tags will be in blue. Instead of defining the style in each HTML document, it is a good idea to create a CSS file and just reference it from within the header section. For example, the CSS file called *myStyles.css* can be referenced using the <link> tags by writing <link rel="stylesheet" type="text/css" href=" myStyles.css" >. The rel attribute specifies the relationship between the current and linked documents; in this case it states that the link is to a *style sheet*. The type attribute specifies the MIME type of the referenced document; which in our case is text/css. MIME, or Multipurpose Internet Mail Extensions, is a set of formats developed originally to enhance e-mail communications. MIME has been extended to be used in HTML documents. The last part of the statement, *href*, indicates where to find the CSS file. In the example, the name of the file is *myStyles.css* and it is located in the same directory as the HTML document. It is very important to include the fully qualified location in case the referenced style sheet is located in a different directory or server. For example, if the CSS file is located in a directory *lower* than the current one, named CSS, then the correct statement should have been /CSS/myStyles.css.

Visit the W3Schools Web site for more information on CSS, and to try it, by clicking on this link.
http://w3schools.com/css/css_intro.asp

## 3. Introduction to JavaScript

JavaScript is a scripting language. A scripting language is one that is interpreted rather than compiled. Compiled languages convert the source code into machine readable object code. Scripting languages are interpreted at runtime, when called into use. HTML is a pseudo language because it does not *make decisions* as it only describes a document and its contents. With JavaScript you can include conditional statements, such as if something is true, the do some other thing. JavaScript was designed to provide some interactivity, or dynamism, to Web pages. JavaScript set of instructions, also known as *snippets*, can be inserted anywhere inside the <html> </html> tags. One important thing to remember: JavaScript is in no way related to the Java programming language.

JavaScript is officially known as ECMAScript-262, and was adopted by the ECMA (European Computer Manufacturers Association) standards association in 1997. JavaScript can be used to validate form information, to manipulate HTML elements, or even to decide which content is rendered depending on the type of browser asking for it. JavaScript snippets can be included in the header section or inside the body section of the HTML document. JavaScript files can also be referenced, just as CSS files, plus the reference can be stated either in the header or body sections of the HTML document. For example, to use an external JavaScript file named *myScripts.js*, you would write:

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

The *src* attribute indicates where the file is located; in this case it can be found in the same directory as the HTML document. You can try a simple snippet by visiting the W3Schools at this link.
http://www.w3schools.com/js/js_intro.asp
You can also open notepad in your computer and write the following statements:

```
<!DOCTYPE html>
<html>
      <head>
            <title>JavaScript Example</title>
      </head>
      <body>
            <h1>My First Web Page</h1>

            <script>alert("My First JavaScript");</script>
      </body>
</html>
```

Save the file as **JavaScriptExample.html** to your Desktop directory. Next, go to your Desktop, locate the file, and open it using a Web browser. NOTE: Most operating systems will automatically open .html files with your default browser; hence you can just click (or double-click) on the file to open it.

So, what just happened? When opening the HTML document the snippet *executed* and rendered an Alert Box in front of the Web page. The box contains the message included inside the double quotes ( “ ” ) of the snippet’s instructions and an OK button. Clicking on the “OK” button closes the box and displays the Web page. As you have just experimented, JavaScript can be very easy to use.

## 4. Conclusion

You are ready to create your first Web page!

By now, you should understand the basic behavior of HTML, CSS, and JavaScript. Use the W3Schools’ Web site as a reference to learn even more about HTML, CSS, and JavaScript. If you have any questions, do not hesitate to contact your instructor by bringing the question to our next meeting.

Answer the questions in the next section of this learning module to test your knowledge of the concepts described in this lesson.