

## Week 2 – Fourth Meeting: Online

Topics covered:

1. HTML 5
2. Using CSS3
3. Using JavaScript functions

### 1. HTML 5

HTML 5 is the latest version of the ever present Hyper Text Markup Language. The main differences between HTML 5 and the previous version are:

- No support for deprecated elements and attributes
- Includes elements for better form handling
- Includes new media handling elements, such as <audio>, <video>, etc.

HTML 4.01 became a standard in 1999. Since then some of the original elements and attributes have become obsolete, others were never used, and still others were used incorrectly. HTML 5 has re-written some elements, removed some, and included new ones in order to account for the dynamic use of the Web.

One of the new elements introduced in version 5 is the canvas. This element is used to draw graphics leveraging the power of scripting languages. It is a graphics container. The structure of the canvas tag is as follows:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

- Always use an id to uniquely identify the element
- <canvas> *declares* an empty container
- Always declare a width and height
- Use JavaScript to *draw*
- Visit this [W3Schools Web page](#) and try it yourself!

Another interesting HTML 5 addition is *Drag and Drop*. It lets Web surfers drag and drop object from one part of a page to another. Drag and Drop is also implemented using JavaScript. You can see an example [here](#).

The video tag provides a standard for including videos in any Web page. Like most other tags, it requires an opening and closing version, <video> </video>. Using this tag you can embed video in your page. You can also add controls to play/pause or change the size of the image. You can even add multiple sources in the same video declaration! [Visit this link](#) for an example.

HTML 5 includes another nifty element, the [application cache](#). This element enables the Web designer to allow user to view and use content while offline.

## 2. CSS3

CSS, short for Cascading Style Sheets, define how elements inside the body tags should be displayed. In order to keep up with the way users expect a Web page to *behave*, some changes were required. The new standard is CSS3. It supports previous versions of CSS, and its main advantage is the use of modules. Most of the parts of the *old* specification are now referred to as modules. Some important modules are: Selectors, Box Model, Backgrounds and Borders, and Animations. It is important to mention that CSS3 is a *work in progress*; hence not all browsers support it. Visit this [Web page](#) to learn more.

The Borders module allows you to create rounded borders for your *boxes*. For example, you can use an image as a border without the need to use a graphics design program (like Photoshop). Borders have three important properties: border-radius, box-shadow, and border-image. Each of them allows you to specify the size in pixels for each border. Visit the [W3Schools page on Borders](#) for more information and to try them out.

The Background module allows the designer to specify the size of the background image used. The earlier use of background in CSS determined the size based on the actual image's size. This new specification allows the designer to re-use background images in different contexts. Size can be specified in pixels or percentage. This [Web page](#) has an excellent example.

The Text Effects module allows the designer to decorate text in a Web page by adding a shadow. It also enables the arranging of text by providing a wrap method. The word-wrap property allows the designer to *fit* text inside a box, but splits words in the middle if necessary. Visit the [W3Schools Web site](#) to learn about other new text properties.

One of the most important modules implemented by CSS3 is the Animations module. This module allows the designer to create animations without the use of Flash, JavaScript, or animated images. In order to use the Animations module you need to learn about the @keyframes rule. This rule is responsible for creating the animation in your HTML 5 document. Please note that not all browsers support this feature, specifically Internet Explorer. Try it out [here](#).

The [User Interface](#) module introduces new features, such as:

- Resize
- Box-sizing
- Outline-offset

These features allow the designer to enable the user to resize page elements, create outlines for boxes or other objects, and define certain elements to fit in a specific area in a specific manner.

### 3. JavaScript Functions

Functions are source code modules that perform one duty. The use of functions stems from the need to provide modular components in order to make systems more easily maintainable. Functions have a name, and execute only when they are *called*. When a function is called, you use the following statement:

```
function_name(arguments)
```

Some functions require arguments, pieces of information provided to the function so it can use it to perform its job. Not all functions have arguments. Functions require two steps: the definition and the call. To define a JavaScript function, use the following nomenclature:

```
function function_name(arguments) {  
    The code that defines the function.  
}
```

Notice the opening and closing “curly brackets” ( { } ). All functions must include them. Within the curly brackets you write the JavaScript code that determines what the function does. Remember the Alert Box example in the previous online lesson? It can also be done using a function. We first define the function, inserting the code in the header section. Then we call this function from the body of the HTML document. In this case we use the click of a button to call the function.

```
<!DOCTYPE html>  
<html>  
<head>  
  <script>  
    function myAlert() {  
      alert("Welcome to JavaScript!");  
    }  
  </script>  
</head>  
<body>  
  <button onclick="myAlert()">Try it</button>  
</body>  
</html>
```

Open Notepad in your computer and copy the HTML code. Save the file as **FunctionExample.html** to your desktop directory. Then locate it and see what happens. For more information about JavaScript functions, visit this [Web page](#).

#### **4. Conclusion**

After learning about the new addition to HTML 5, the new services offered by CSS3 and JavaScript functions, you can now add interactivity to your Web page. Make sure your project includes these newer versions of the standards in order to be able to embed videos and create animations easily.

Your project should be almost completed by now! Remember to use HTML 5 and leverage the power of CSS3 and JavaScript to make your Web site dynamic and user friendly. Do not hesitate to contact your instructor with questions about the topics covered in this lesson.

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