Week 1 Reflection & Notes

Semantic HTML Markup The term “HTML Markup” refers to all of the tags (<h1>,<ul>, etc) that one uses to format their web page. “Semantic” refers to the meaning of information on one’s web page; so, the concept of “Semantic HTML Markup” is the process of using the correct HTML tags to identify information.

For example, if I use the header 1 tag (<h1></h1>) for the name of my website, any information following it would be a subsection; so if you were to add another heading to the information, you would use header 2 (<h2></h2>).

Tags We Used This Week

*note: during this course I will go over the most common and useful tags for marking up our website; however, we may not cover them all. I encourage you guys to follow this link to w3schools.com. This website is a good reference for all of the HTML elements, as well as CSS rules for styling.

<!DOCTYPE html> this is the Doctype tag for html5 and should be the first element in your html document.

<html> The <html> tag needs to enclose your head and body tags; so, your basic html page should look like this:

```
<!DOCTYPE html>
<html>
        <head>
        </head>
        <body>
        </body>
</html>
```

These tags are to inform the browser that everything occurring between them is html. Your pages will typically work fine without them (obviously, because ours did :); however, you won’t pass validation (which we’ll get into later). Simply put, it’s the right way to do it.

<head> Between the head tags is where we place meta information such as the title of the website (<title></title>). It is also good practice to link to any external css and javascript files here.

<body> the body occurs below the head. Within the opening and closing body tags are where one places all of their content for the web page.

<h1>,<h2>,... These are header tags. We use them when we need to title a new section of information.

  The paragraph tag is used to enclose sections of text. It is important that all information in our website is enclosed in a tag that appropriately identifies the type of information it is. If you have a block of text that you don't consider to be a paragraph, but don't know what else to mark it up as, just go with the paragraph tag.

  The div tag does not have semantic meaning. It exists only to further categorize and divide information. This is most often used for purposes of creating section that can be targeted with styles. Div tags are very useful to for creating a layout. Div tags can take class and id properties just like any other tag.

Attributes
Attributes are used to modify html tags. The two common attributes we learned about this week are "classes", and "ids".

class: A class is a name we give to one or more html objects. We can then target the class via CSS and change its visual appearance. Here is an example on a paragraph tag.
This paragraph has been given a class. We can make up whatever name we like for the class; however, it's a good idea to give it a name that makes sense.

CSS for the rule would look like this:
```
.myBoldClass{
        font-weight: bold;
}
```

In our css, the syntax to address a class must be preceded by a period ".".

id: An Id can be used in the same way as a class; however, an id is not reusable.
Here is an example on a div that is used as a container for the rest of our html.
```
<div id="Container">
</div>
```

In this example, I would not reuse the id Container on anything else.

I could target the id from CSS with the following code:

```
#Container{
        width: 1000px;
        min-height: 600px;
}
```

When targeting an Id via CSS the Id name is preceded by a hashtag “#”.

For further reading and interactive examples with IDs and Classes, check out the following link.
Id: http://www.w3schools.com/tags/att_global_id.asp
class: http://www.w3schools.com/tags/att_global_class.asp

CSS Recap
CSS is what we use to style the visual presentation of our HTML.

If you are writing your styles in your html file then they must be between the <style></style> tags, and it is best practice to have this in the <head> of your html file.

A good practice is to have your CSS in its own .css file that is simply linked to from the html file. This way you only have to update your styles in one place, rather than for each page of the website.

Here is the markup for linking to a css file. This example assumes that you’ve named your file “styles.css” and placed it in a folder called “css” that is in the root of your website.

```
<link type=”text/css” rel=”stylesheet” href=”css/styles.css”>
```

float: The float is one of the harder rules to master, and it is one that is used often. Do not worry if you didn’t understand them in our first week. We’ll be spending some quality time with it during week two, and beyond.

padding: the space between the bounding box of an html element, and the content it contains.
usage:

```
        padding: 0 0 0 0;
        (top, right, bottom, left)
```

If only one value is used for all four positions, you only need to write that one value, not all four. The above example is shorthand. You can also specify individual sides with:

padding-top:

padding-right:
padding-bottom:
padding-left:

margin: the space between the bounding box of an html element and the other html elements surrounding it.
usage:
margin: 0 0 0 0;
(top, right, bottom, left)If only one value is used for all four positions, you only need to write that one value, not all four.

The above example is shorthand. You can also specify an individual side with:
margin-top:
margin-right:
margin-bottom:
margin-left:

Concepts

Box Model: Each html element can be considered a box. This is a "bounding box". The box's dimensions are the width and height of the element it contains. The "Box Model" refers to the process of laying out your html elements.

Here is a link to w3shools.org. They have further description and illustrations.
http://www.w3schools.com/css/css_boxmodel.asp

Wireframe the wireframe is what you guys began working on. It is a diagram, or blueprint of your site. It often only contains boxes (the bounding boxes) of your elements, and where you want them to be placed visually.

Further definition can be found at wikipedia, they're accurate enough on this one.
http://en.wikipedia.org/wiki/Website_wireframe

If you would like to investigate floats on your own, check out the following link:
http://www.w3schools.com/cssref/pr_class_float.asp

Again, our next class is going to focus heavily on understanding the use of floats, and from there we'll transition to our next CSS topic.

See you guys in class!