HTML5: It's A Whole New Ballgame

In this article, Nancy is going to introduce you to **HTML5**. This article is the first in a series that will start with an overview of the HTML5 structure and continue with a look at new tags and uses for the tags. We will also cover what browsers are supported, which are not and how to get the support-nots to play nicely.

Background

A few months ago, HTML5 was something to me that was going to happen in about a decade, because browser support wasn't there and wasn't going to be there in anything that served more than ten percent of the browsing community. I have been doing this sort of work since 1996 and if there is one thing I have learned, it's that progress on getting anything new is slow because there are so many parties that have to agree; so many browsers that have to adopt; and then there is the matter of getting designers to use it and use it right. Development progress generally takes years! So why should I worry about HTML5?

Then there was a barrage of "Steve Jobs Hates Adobe" and "Flash is Going to Die" blog posts and articles and suddenly the climate had turned to HTML5 **versus** Flash. My first thought was, "but this isn't even going to be an issue for YEARS! Fast forward about six weeks and I am writing my first article on HTML 5 after living and breathing the stuff on a daily basis for the last few weeks and studying it intently for several before that.

NOTE: A word about HTML5 versus Flash. That is a misconception produced in the media that is just as exaggerated as the "Apple versus Adobe" claims. HTML5, in my opinion, is not going to replace Flash, but I can surely see uses for both technologies. Unless browser support for the Flash player is virtually abandoned, having universal browser support for both HTML5 **and** the Flash Player is going to give the developer a multitude of options, making all of us the real winners here.

Recently, Adobe released the **HTML5 pack for Dreamweaver** (read http://www.dmxzone.com/go?17604) to help the newly released **Dreamweaver CS5** to take better into account some of the amazing attributes of HTML5**.** It was virtually Adobe's statement of "we HEAR you and we like HTML5, really we do!" This is a very good thing. HTML5 is now heavily supported in Dreamweaver CS5, right down to the code hinting.

I found myself thinking that I needed to get my rear in gear and find out about HTML5 and I was pleased to find a simple structure that makes sense. I can remember when web work was mostly straight pages of text with images here and there. After awhile, someone discovered you could put anything inside a table cell even though they were supposed to be only for data and finally you could layout a page (in a table) and get it to look like something. This went along smoothly for awhile, then The Powers That Be (all hail the W3C) said, no, no, you can't lay out pages that way, you have to have divs all over the place and use **CSS** to shove them around to get what you want.

Of course, there are divs and there are divs. This is not the Netscape 4 AP Div that you could just lay on top of other junk, although I am continually shocked as to how many web designer-wanna-be's think that. I actually inherited a website a few months ago where the guy before me did the whole site with layers upon layers upon layers. You couldn't edit anything and I had to remake the entire site. But this is not those DIVs. This is the other div where we put them around everything, sometimes twice or three times to accommodate the CSS that will tell them where they sit and how they can look. We give them identification names (id) and we name them something meaningful and then we tell them how to look and how to act using CSS. Then we tear out our hair because this div won't float that way or the margin is doubling in this browser while being non-existent in another browser.

What is HTML5?

The entrance of HTML5 almost looks like a return to a simpler life. We have tags that make sense. A header is not THE header, but the heading of a section, of an article or whatever. The same holds true for other elements like the footer, section, article and so on. We can have more than one header, for example, in a page and they truly act as a header over the content they govern. We have a nav tag for .. you guessed it .. navigation. We have aside for secondary content of the page and we have tags called article and section that can subdivide the content into something that defines where it goes. Simpler.

This is probably easier to see with an example, however, so let's take a look at the structure by constructing a simple HTML5 document. Our discussion should begin at the very beginning, the doctype. If you look at the list of doctypes supported by Dreamweaver CS5, it's enough to make your head spin. Starting with HTML 4.01, we have transitional or strict. Then we move to XHTML 1.0 where again we have transitional or strict. Moving along, we can proceed to XHTML 1.1 or if you want to design for Mobile, there's a doctype for that as well. Each one gets a little more complicated and a little longer than the earlier ones.

Then we move to HTML5. Have you seen the doctype for HTML5? It's this:

```
<!DOCTYPE HTML>
```

No kidding. That's it. Simpler.

The entire document from that point on is also simpler, or it should be. The following basic HTML5 document is all you need to have and it can grow from there.

```
<!DOCTYPE html>
<html>
<head>
<meta charset=“utf-8” />
<title>My HTML5 Document</title>
<!--[if IE]>
<script src=“http://html5shiv.googlecode.com/svn/trunk/html5.js”></script>
<![endif]-->
<style>
  article, aside, dialog, figure, footer, header,
  hgroup, menu, nav, section { display: block; }
</style>
</head>
<body>
<p>This is the first HTML5 document.</p>
</body>
</html>
```

This simple document tells you a lot about HTML5 just in its few lines. In the head of the document, you will notice that we have a link to a script on Google Code. That script will get some of the HTML5 elements to be recognized in IE8 and below, which is everything available at the present time. IE9 promises to be HTML5 compliant so we'll watch that with interest. The style declaration which comes next displays the new HTML5 elements as blocks so they have some substance to them in current browsers that don't support the new tags. It's important to put this chunk of CSS in every HTML5 document that you create at the present time. As browser support continues to grow, there will come a time when this style chunk is no longer necessary.

The rest of the document is the body but it's not quite the same body we have been used to seeing. Unless you're still designing in tables, you're used to using a lot of divs with id's like header, content, footer, wrapper and sidebar but those id's are for identification in the application of the pertinent CSS. The structural tags of

HTML5 have nothing to do with CSS. The HTML5 tags I would define as "structural" would be **header, section, aside, article, nav** and **footer**. In part one of this article, I would like to discuss just these structural tags in detail.

Header

By definition, the header is the beginning of whatever content it is introducing. This can be a page, a section, an article or even a table of contents. Therefore, there is often more than one header on a page so it's easy to see that one would not use the header tag when constructing the CSS for a page because the CSS for one header probably has nothing to do with the CSS for another. The trick here is that we have to get beyond the thought that "header" is an ID for a div and we center it with auto margins, etc. If we did that, we would have a jumbled mess because while one header may look like the top of a page, another header on the same page may have smaller headings contained within it and therefore, a whole different set of CSS rules.
Header is used like this as the header of the page:

```
<body>
<header>
<h1>This is My Page</h1>
</header>
<section><p>The first section of my page is going to be about anything I want it to be
about.</p></section>
<section><p>The second section of my page is going to be about the schools I have
attended.</p></section>
<footer>The End</footer>
</body>
```

However, header can also be used like this:

```
<body>
<header>
<h1>This is My Page</h1>
</header>
<section><p>The first section of my page is going to be about anything I want it to be
about.</p>
<article>
<header><h2>Article 1</h2></header>
<p>This is my first article and I hope you like it.</p>
</article>
<article>
<header><h2>Article 2</h2></header>
<p>This is my second article and it absolutely rocks!</p>
</article>
<footer>The End</footer>
</section>
</body>
```

You will notice that the second example has a header for the page heading, but also two smaller headers, one as the heading of each of the two articles contained in the document. When we get to styling the document above, we could either declare id's on the header elements, use classes for styling, or in some cases, enclose the section in a div whose id we could than use for the declaration of the CSS. This reasoning is probably why Adobe switched their CSS page samples in Dreamweaver CS5 to use classes for styling rather than div id's.

Section

The definition of "section" in the HTML5 spec is as follows:

The section element represents a generic document or application section…The section element is not a generic container element. When an element is needed for styling purposes or as a convenience for scripting, authors are encouraged to use the div element instead.

Please read the definition above. Now read it again. It is very easy to see "section" and think .. aha .. there is my container and go to town creating CSS for it. However, please note that the "section" tag is NOT a container for styling. A section is a portion of a document and the tag has nothing to do with the way it I s styled. A section could contain one or more articles and each of those could have headers and footers within them but a section will basically stop when the end of the document portion or application portion is reached.

Article

An article is a section of content that reads like a blog post or a newspaper article. An Article can contain sections, it can have its own header or footer, and will even contain comments sometimes. When you have content and you can't decide if you should use Section or Article, ask yourself if the content reads like a standalone piece that you might read on a blog and get the complete thought so that it could easily be its own short page. If that is the case, then you want Article. Section is more generic in nature.

Aside

The "aside" tag is used to set apart secondary content, similar to what we are used to seeing in a sidebar. "Aside" does not contain your articles or other main content. Sometimes the navigation will be enclosed in an aside tag, but it is much more than that. You might have contact information that you want set off by itself away from the content. You would use the "aside" tag for that sort of content.

Nav

The nav tag is used for the navigation of the page, navigation of a section or a group of pages. Like the other tags, the user is not limited to one "Nav" section on a page. In fact, it is fairly common to see main navigation across the top of a page, secondary navigation perhaps on the left side of a page and text navigation just before the page's footer. All these instances can use the "nav" tag to describe the navigational elements contained within them. By definition, use the "nav" element wherever you have links to other pages within the site or to other parts of the page.

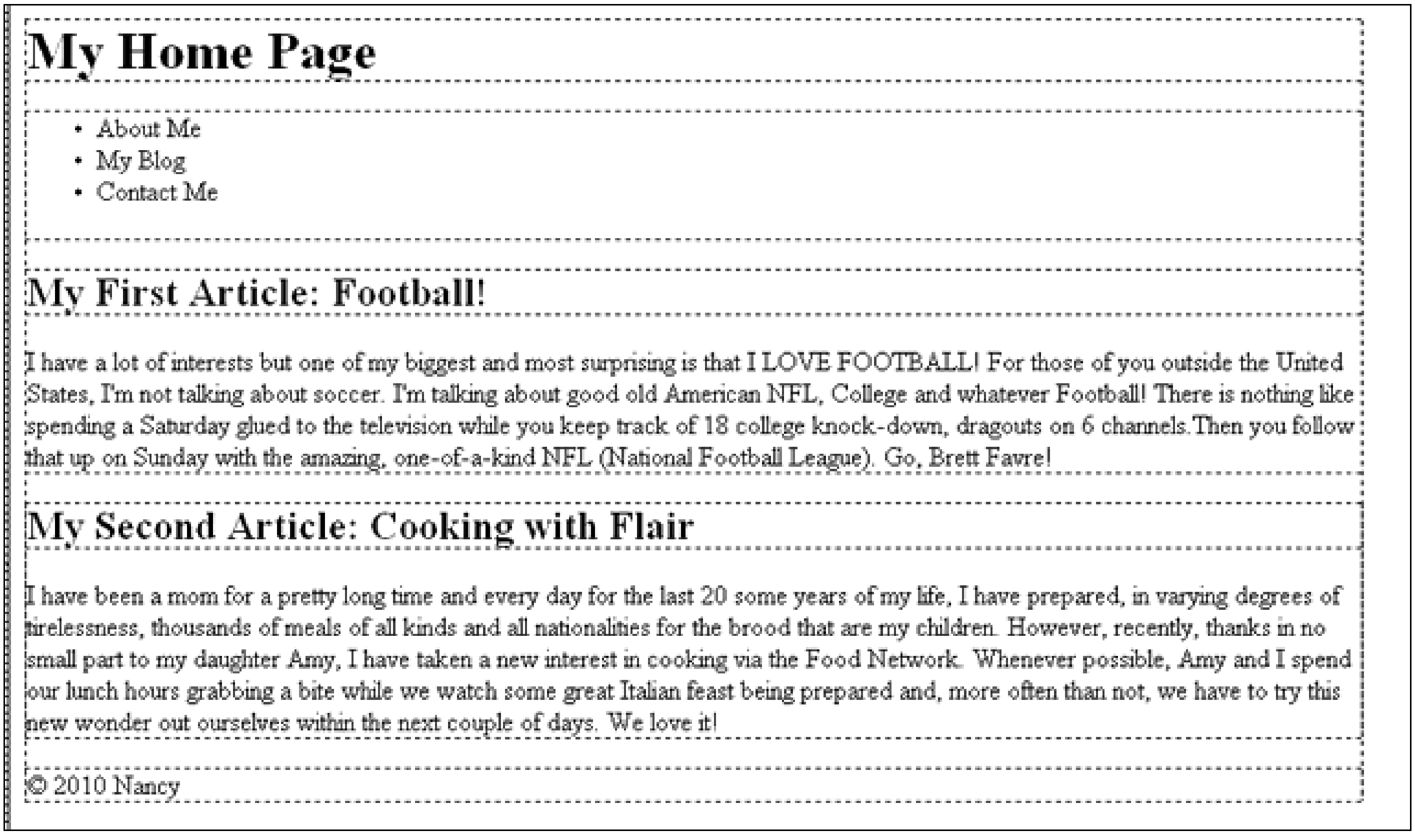
Footer

Footer is pretty much the opposite of "header". The "footer" tag can be used for the content at a bottom of a page, but you might also have a footer at the bottom of an article or even a section. I have also seen footer instances in an aside section. Like the header tag, the footer is not limited to one on the page and it has nothing to do with styling.

There are, however, some guidelines of what might be appropriate content for a footer. If the footer is at the bottom of the page, one would probably put a copyright notice and/or author information in it. You are allowed to put both sections and nav elements in the footer, as one might do when creating a text navigation near the bottom of the page for accessibility reasons. Contact information might be another item to go into a footer, but the HTML5 spec people say you must include the **address** element around the contact information within the footer if you do so. Address is one of the elements we'll be discussing in the next segment(s).

Putting It All Together

I typically learn best by example, so I have put together a somewhat simple example of an HTML 5 document so you can get an idea of how the structural elements of HTML5 work together. You can copy the code posted below and paste it into a browser to study its structure.
The unstyled document looks like this:



My Home Page

- About Me
- My Blog
- Contact Me

My First Article: Football!

I have a lot of interests but one of my biggest and most surprising is that I LOVE FOOTBALL! For those of you outside the United States, I'm not talking about soccer. I'm talking about good old American NFL, College and whatever Football! There is nothing like spending a Saturday glued to the television while you keep track of 18 college knock-down, dragouts on 6 channels.Then you follow that up on Sunday with the amazing, one-of-a-kind NFL (National Football League). Go, Brett Favre!

My Second Article: Cooking with Flair

I have been a mom for a pretty long time and every day for the last 20 some years of my life, I have prepared, in varying degrees of tirelessness, thousands of meals of all kinds and all nationalities for the brood that are my children. However, recently, thanks in no small part to my daughter Amy, I have taken a new interest in cooking via the Food Network. Whenever possible, Amy and I spend our lunch hours grabbing a bite while we watch some great Italian feast being prepared and, more often than not, we have to try this new wonder out ourselves within the next couple of days. We love it!

© 2010 Nancy

Here is the html code for the same page:

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8" />
<title>My HTML5 Document</title>
<!--[if IE]>
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<style>
  article, aside, dialog, figure, footer, header,
  hgroup, menu, nav, section { display: block; }
</style>
</head>

<body>
```

```
<header>
<h1>My Home Page</h1>
</header>
<nav>
<ul>
<li>About Me</li>
<li>My Blog</li>
<li>Contact Me</li>
</ul>
</nav>
<section>
<article>
<header>
<h2>My First Article: Football!</h2>
</header>
<p>I have a lot of interests but one of my biggest and most surprising is that I LOVE
FOOTBALL! For those of you outside the United States, I’m not talking about soccer. I’m
talking about good old American NFL, College and whatever Football! There is nothing
like spending a Saturday glued to the television while you keep track of 18 college
knock-down, dragouts on 6 channels.Then you follow that up on Sunday with the amazing,
one-of-a-kind NFL (National Football League). Go, Brett Favre!  </p>
</article>
<article>
<header>
<h2>My Second Article: Cooking with Flair</h2>
</header>
<p>I have been a mom for a pretty long time and every day for the last 20 some years of
my life, I have prepared, in varying degrees of tirelessness, thousands of meals of all
kinds and all nationalities for the brood that are my children. However, recently,
thanks in no small part to my daughter Amy, I have taken a new interest in cooking via
the Food Network. Whenever possible, Amy and I spend our lunch hours grabbing a bite
while we watch some great Italian feast being prepared and, more often than not, we
have to try this new wonder out ourselves within the next couple of days. We love
it!</p>
</article>
</section>
<footer>
&copy; 2010 Nancy
</footer>
</body>
</html>
```

Conclusion

What we have seen in the code and explanation above is a lot to digest. HTML5 is a lot more than some new tags; it’s really a different way of thinking about content and the practice of applying meaningful sections to each type of content. It’s also a reinforcement of the idea that good page design separates HTML markup from style markup (CSS).

I realize that the code above doesn’t look like much yet, but it will with some styling. Next time, we’ll go through some do’s and don’ts of applying CSS with HTML5, including a few not-yet-well-supported CSS3 elements. Also, in the next installment, we’ll take a look at some of the more “sexy” parts of HTML5, including the highly-publicized “canvas” element and stuff like “drag and drop” in HTML5 and of course, video without the player. It’s all upcoming, so do stay tuned!