Nkatz01

13128128

Advanced Web Authoring Web Development Log – Bootworld.

The semantic appropriateness of the HTML mark-up used in the page

In order for my site to be accessible for both human and computerized readers, such as assistive technology and search engines, I put an emphases on two principals. 1 is to make my html semantically appropriate by choosing HTML tags that reflect the meaning of the content placed in it rather than basing it on layout and presentation. That is keeping presentation and content separate. This I have done by choosing mark-up tags such as main, header, h1 h2 etc… nav, img, article, section and footer instead of just divs and spans and have a 'nav' and 'footer' as just ids in the divs. And when it came to formatting and layout, wherever possible, I've targeted the html tags by using ids and classes that are meaningful to the designer, rather than using the actual html tags so to preserve the separation of content and presentation.

2. A second principal was selecting the 'right' tags for the right content. i.e. an h1 for a header that is really important and an h3 for a header of less importance. Or a main for the content of the page, a p tag for a paragraph and put it after an h tag, since a paragraph or a couple of paragraphs usually needs a header on top. Similarly, I've used article denoting a block of content that is unrelated to its neighbour because it has its own new header and paragraph but yet related enough to be classed within a section as their content is supporting the same theme. Also when targeting them with css formatting, I was minded that those elements that are likely to be repeated in multiple pages and have semantic meaning, should have their classes, also named semantically, for example, pageheader, primary and footer so that I stick to the meaning and generality of the type of content that would conventionally be placed in them.

How the semantic appropriateness could have been improved?

It could have been improved by perhaps taking more time in researching on how I could include even more semantic tags and even fewer divs or match better those elements that I have used to the content which they wrap around. Also, I could have avoided using the ids I have used, despite its benefits, as later discussed, as they help those making the standards for semantic mark-up, such as WC3, to identify better practice and newer more refined tags, based on the attributes web sites often use for class and id values, pertaining to particular type of content.

Problems I have encountered and how I have solved or not solved them.

The following are some of the issues I encountered when working on the site:

Problem

Managing the margins and the paddings of the different containers and child containers. I found it particular tricky since one's padding is the other one's margin and also, when the browser is resized containers follow and may clash with each other if the extent of their container's edge has not been taken into account.

Solution

I've given striking background and borders to all containers throughout the development so that their contraction and expansion could be clearly monitored and only at the very end changed them to the colours specified in the requirements.

Problem

Choosing id names based on the heading or content of the paragraph turned out to be confusing as 1, as a designer, when doing format and positioning, I don't want to be concerned with the content every time I go back and forth from the text-editor to the browser and have to search through the html and CSS packed text-editor for a particular word. Moreover, I want to be able to look at an individual browser renderance of an instance in a css tweak of a particular container and know, immediately which css anchor this is and to whereabouts in the html text it points to, based on the visual position where it is on the page.

Solution

Give it id names that refer to coordinates and size rather than the content it that appears in it, especially since the content can very easily change.

Problem

Have less repeated css, that have only minute differences and are for the same anchors or are for anchors that would have had anyway inherited that particular setting already previously given to their parent. In other words, 1 have a more uniform, one stop shop, and clearly defined measure to use (i.e. em vs %, height vs width, padding/margin vs font and borders).

Solution

Although with this I was not doing so well, I have dealt with it better this time by first making a list of all the anchors I intend to use before using them and tried to add to the list as I went along and added a new anchor, searching first if they weren't already there. Another way of dealing with it was to put the anchors in css in the same order they appear in the html file. I also first concentrated on just the width of the containers/elements, in percentages, then the float positioning tactic, then margins right and left, then top and bottom, in ems, then the paddings, sides and top/bottom, and only then those that can take px, such as fonts, borders. As a final measure I went through some of the elements respectively, after they've been formatted, in the inspect console mode, to look for overrides on my own code and see if I can eliminate them.

Problems I recognise are there but could not manage to solve them.

The right spacing, in the centre between the right and left, bottom, smaller containers in their parent container.

The font size for the some of the paragraphs which seem too big.

Formatting of the form element, specifically the font size and in general font sizes and font weights in the page, I found hard.