Thinking Craftsman

CSS Basic

Nitin Bhide
(http://thinkingcraftsman.in)
May 2011

What is CSS ?

CSS => Cascading Style Sheet

- What is a '*Style Sheet*' ?
- Why '*Cascading*' ?

Structured Document

- The content of Structured documents can be reused in many contexts and presented in various ways
- A **structured document** is an electronic document where some method of embedded coding (i.e. markup) is used to give the whole, and parts, of the document various *structural meanings* according to a schema.
- SGML pioneered the concept of structured documents

Structured Documents

- SGML was based on two novel concepts
  1. Markup should describe a document's structure and other attributes. It should not specify the processing to be performed on it.
     This way descriptive markup needs to be done only once, and will suffice for future processing.
  2. Markup should be rigorous so that the techniques available for processing rigorously-defined objects like programs and databases can be used for processing documents as well.
     For example, generating documents of various quality, automatically generating titles and indices, etc.

- DTD (Document Type Definition) defines the 'schema' (or meaning of each structural element).

Structured Documents

- Most widely used structured document formats
  - HTML, XML, SVG, XUL etc
- XML
  - The W3C XML (Extensible Markup Language) is a profile (subset) of SGML
  - It is designed to ease the implementation of the parser compared to a full SGML parser,
  - XML is primarily designed for use on the World Wide Web
  - Applications of XML include *XHTML, XQuery, XSLT, XForms, XPointer, JSP, SVG, RSS, Atom, XML-RPC, Semantic Web, and SOAP*.

What is a Style Sheet ?

- For content in structured documents to be presented, a set of stylistic rules must be applied.
- Typical Stylistic rules describe
  - Colors,
  - Fonts,
  - Layout
- A collection of stylistic rules is called a **style sheet.**

- Different style sheets can be attached to the logical structure to produce different presentations.
  - *For example, how content will be when printed, when viewed on computer, in-voice, for Braile etc.*

Why 'Cascading' ?

- CSS specifies a *priority scheme* to determine which style rules apply if more than one rule matches against a particular element.

- In this so-called *cascade*, priorities or *weights* are calculated and assigned to rules, so that the results are predictable.

CSS – Some history

- CSS Level 1 :
  - Initial Release.
  - 17 December 1996; 15 years ago
- CSS Level 2 :
  - Published in May 1998
  - Superset of Level 1
- CSS Level 3:
  - CSS3 is divided into several separate documents called "modules".
  - The earliest CSS3 drafts published in June 1999.

CSS Structure - Rule

A style sheet consists of a list of *rules*.

Each rule or rule-set consists of one or more ***selectors*** and a ***declaration block***.

CSS Structure – Declaration Block

consists of a list of *declarations* in braces.

Each declaration itself consists of a *property*, a colon (:), a *value*.

If there are multiple declarations in a block, a semi-colon (;) must be inserted to separate each declaration.

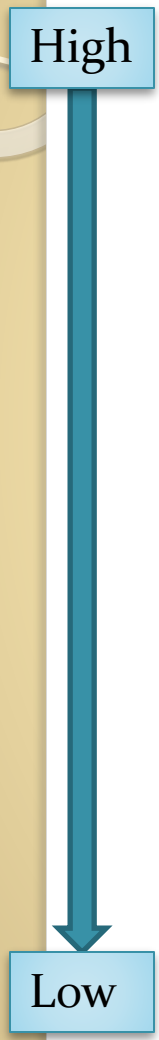
CSS Structure : Selectors

are used to declare which of the markup elements a style applies to, a kind of match expression.

Selectors may apply to all elements of a specific type (e.g. class), or only those elements that match a certain attribute (e.g. id);

Elements may be matched depending on how they are placed relative to each other in the markup code, or on how they are nested within the DOM.

Cascade scheme (priority scheme)



- Author styles (provided by the web page author):
  - Inline styles, inside the HTML document, style information on a single element, specified using the "style" attribute
  - Embedded style, blocks of CSS information inside the HTML itself
  - External style sheets, i.e., a separate CSS file referenced from the document
- User style:
  - A local CSS file the user specifies with a browser option, which acts as an override applied to all documents
- User agent (browser) style:
  - Default styles applied by the user agent, i.e., the browser's default settings for element presentation
  - Each browser has slightly different defaults for various elements

CSS Advantages

Flexibility

Separation of content from presentation

- CSS facilitates publication of content in multiple presentation formats based on nominal parameters. Nominal parameters include
  - explicit user preferences,
  - different web browsers,
  - the type of device being used to view the content (a desktop computer or mobile Internet device),
  - the geographic location of the user and many other variables.

Site-wide consistency

- a global style sheet can be used to affect and style elements site-wide.
- If the situation arises that the styling of the elements should need to be changed or adjusted, these changes can be made by editing rules in the global style sheet.
- Before CSS, this sort of maintenance was more difficult, expensive and time-consuming.

CSS Advantages

Reduced Bandwidth

- A stylesheet specifies the style *once* for a range of HTML elements selected by class, type or relationship to others.
- This is much more efficient than repeating style information inline for each occurrence of the element.
- An external stylesheet is usually stored in the browser cache, and can therefore be used on multiple pages without being reloaded, further reducing data transfer over a network.

Page reformatting

- With a simple change of one line, a different style sheet can be used for the same page.
- This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices. Furthermore, devices not able to understand the styling still display the content.

CSS BROWSER SUPPORT

Browser Support

- Features supported
  - Feature support of various browser and their version is documented on wikipedia
  - Comparison of CSS Layout Engines
- Different User Agent styles:
  - Default values of various elements like fonts, colors, layouts etc.
  - Yahoo 'reset.css'

ACID Tests

- Testpages Developed by 'Web Standard Project'.
- ACID 2 focuses mainly on CSS.
- ACID3 tests DOM and Javascript as well.
- Mainly developed by Ian Hickson (Google employee)
- Test pages are available at http://acidtests.org

CSS Frameworks

- Pre-prepared libraries
- That allow for easier, more standards-compliant styling of web pages
- Minimize the browser specific variations
- usually incorporated as external .css sheets referenced in the HTML <head>
- Included before 'custom css'.

- For Example
  - Yahoo CSS Grid, reset.css
  - BluePrint (by Googlers)
  - 960 Grid

Thinking Craftsman

CSS BOX MODEL & POSITIONING

CSS Box Model

Margin	Border	Padding	Content
• Clears an area around the border. • The margin does not have a background color, • it is completely transparent	• A border that goes around the padding and content. • The border is affected by the background color of the box	• Clears an area around the content. • The padding is affected by the background color of the box	• The content of the box, where text and images appear

Positioning Schemes

CSS 2.1 defines three positioning schemes:

Normal flow

- *Inline* items are laid out in the same way as the letters in words in text, one after the other across the available space until there is no more room, then starting a new line below.
- *Block* items stack vertically, like paragraphs and like the items in a bulleted list.
- Normal flow also includes relative positioning of block or inline items, and run-in boxes.

Floats

- A floated item is taken out of the normal flow and shifted to the left or right as far as possible in the space available.
- Other content then flows alongside the floated item.

Absolute positioning

- An absolutely positioned item has no place in, and no effect on, the normal flow of other items.
- It occupies its assigned position in its container independently of other items

Position

Static

- The default value places the item in the *normal flow*

Relative

- The item is placed in the *normal flow*, and then shifted or offset from that position.
- Subsequent flow items are laid out as if the item had not been moved.

Absolute

- Specifies *absolute positioning*

Fixed

- The item is *absolutely positioned* in a fixed position on the screen even as the rest of the document is scrolled

If an item is positioned in any way other than static, then the further properties top, bottom, left, and right are used to specify offsets and positions.

Float & Clear

Notes

- *Absolutely* positioned or *fixed* items cannot be floated.
- Other elements normally flow around floated items, unless they are prevented from doing so by their clear property.

Float:left

- *Floats* to the left of the line that it would have appeared in; other items may flow around its right side

Float:right

- *Floats* to the right of the line that it would have appeared in; other items may flow around its left side

Float:none

- Removes the *float* property from an item

Thinking Craftsman

CSS LAYOUTS

Table layouts Vs CSS Layouts

Table Layouts	CSS Layouts
• mixes presentational data in with content. • Needs to create separate pages for different 'representations' like print, on screen, mobile devices, voice etc • Hard to maintain visual consistency across pages • Larger page download sizes.	• Smaller page sizes, CSS file is shared across • Separates presentation data and content • Different css defines how page will be displayed on various 'representations' like print, on screen, mobile devices etc. • Easy to maintain visual consistency across pages. • **Redesigns are easier and less expensive**

Table layouts Vs CSS Layouts

Table Layout	CSS Layout
• Looks nearly same in all browsers • Works well in HTML emails • Hence Minimize the table use. • Easy to get 'correct' fluid layout	• Needs some care to ensure that results are same in all browsers. • Because of 'casacading' nature, difficult to find offending css rule. • Small change can break multiple pages. • Don't work well in HTML emails. • Take correct to get the correct results in 'fluid' layouts

Thinking Craftsman

DESIGNING WITH CSS

Steps to Design with CSS

Identify the types of content/information in your site as a whole

Identify the sections and pages of your site.

- Product information
- Pricing information
- Company information
- Investor information
- Shopping cart
- User forums and so on

Steps to Design with CSS

Breaking down your pages with logical divisions of content.

- Main navigation
- Sub navigation
- Headers and footers
- Content
- Related information
- Other

Analyze existing markup for presentational HTML that can be replaced with structural markup

- <font> tags
- the <b> and
 markup.
- presentational markup for tables (bgcolor, background, and the like).

Steps to Design With CSS

Replace presentational tags with structural markup

- Think about the *structure* of your document!
- What is the most important header? Mark it up with an <h1> tag.
- Mark your subheads with <h2> tags and so on.
- Mark up paragraphs with <p> tags.
- Mark up your navigation as unordered lists.
- Choose a DOCTYPE and use it. (Use XHTML transitional, if you really want to be 'strict' use XHTML strict.)

Divide your page into logical divs

- Put your main navigation into a div with an id of mainnav;
- put your subnav inside a div with an id or class of subnav,
- put your footer in a ,
- wrap your content inside a .
- Page doesn't look like much now, but once you start adding rules to your style sheets, things will get better quickly.

Steps in Designing with CSS

It's time to start writing your CSS

- At the begining, give each div a border.
  For example, div {border: 1px dotted gray; padding: .5em}
  This will help you see where they begin and end, and also whether or not you have any nesting going on.
- Write your CSS for element selectors first
  (<html>, <body>, <p>, <h1>, <h2>, <ul>, <li>, etc.)
- Use contextual or descendant selectors as much as possible.
  This will keep your markup much cleaner. For example, #subnav li {border: 1px solid black; padding: .5em; display: inline} will only affect list items that occur within your subnav div.
- Test in as many browsers as you can and get your friends to test it in their browsers.

Thinking Craftsman

CSS TIPS

CSS Tips

- Minimize use of tables.
- Make sure that ids are unique.
  - CSS just assumes Ids are unique.
  - And incase of multiple ids, apply the rule on first id found
- Donot use external or internal style sheets in HTML emails. Use ‘style’ attribute on individual element.

CSS Limitations

Poor controls for flexible layouts

- CSS is still at heart a styling language (for fonts, colours, borders and other decoration), not a layout language (for blocks with positions, sizes, margins, and so on).
- While new additions to CSS3 provide a stronger, more robust feature-set for layout
- Still These limitations mean that creating fluid layouts generally requires hand-coding of CSS.
- Hence key feature of Many CSS Framework, is 'Grid Layouts'.
- It has held back the development of a standards-based WYSIWYG editor.

Selectors are unable to ascend

- CSS offers no way to select a parent or ancestor of an element that satisfies certain criteria.
- An advanced selector scheme (like XPath) would enable more sophisticated style sheets.
- The main reasons for the CSS Working Group rejecting proposals for parent selectors are related to browser performance and incremental rendering issues.

CSS Limitations

Vertical control limitations

- Vertical placement is frequently unintuitive, convoluted, or impossible.
- Simple tasks *(such as centering an element vertically or getting a footer to be placed no higher than bottom of viewport)* either require complicated and unintuitive style rules, or simple but widely unsupported rules.

Absence of expressions

- There is currently no ability to specify property values as simple expressions (such as margin-left: 10% - 3em + 4px;).
- This would be useful in a variety of cases, such as calculating the size of columns subject to a constraint on the sum of all columns.
- However, a working draft with a calc() value to address this limitation has been published by the CSS WG.
- Internet Explorer versions 5 to 7 support a proprietary expression() statement, It statement is no longer supported from Internet Explorer 8 onwards,

CSS Limitations

Lack of column declaration

- While possible in current CSS 3 (using the column-count module)
- Layouts with multiple columns can be complex to implement in CSS2.1.
- With CSS 2.1, the process is often done using floating elements, which are often rendered differently by different browsers, different computer screen shapes, and different screen ratios set on standard monitors.

Cannot explicitly declare new scope independently of position

- Scoping rules for properties such as z-index look for the closest parent element with a position:absolute or position:relative attribute.
- This odd coupling has undesired effects such as it is impossible to avoid declaring a new scope when one is forced to adjust an element's position, preventing one from using the desired scope of a parent element.

Pseudo-class dynamic behavior not controllable

- CSS implements pseudo-classes that allow a degree of user feedback by conditional application of alternate styles.
- One CSS pseudo-class, ":hover", is dynamic (equivalent of javascript "onmouseover") and has potential for abuse (e.g., implementing cursor-proximity popups),
- but CSS has no ability for a client to disable it (no "disable"-like property) or limit its effects (no "nochange"-like values for each property).

REFERENCES

References

1. CSS Filters (or CSS hacks)
2. CSS Limitations
3. CSS Feature Support comparison table
4. CSS contents and browser compatibility
5. CSS Zen Garden.
   Lots CSS design examples
6. Why Tables for Layout is Stupid ?

References

7. W3Schools CSS Tutorial
8. Web Standards Project
9. ACID Tests and ACID3 Tests