What is the current status of CSS?

Author: Mihai Rotaru

Date: 1 Dec 2011

finalized standard is CSS2.1 - which became a W3C Recommendation on the 7th of July 2011. CSS2.1 was developed after numerous flaws were found in CSS2, and some features were not implemented by browsers. CSS2 is currently not maintained (http://www.w3.org/TR/2008/REC-CSS2-20080411/), and W3C points authors and implementors to the CSS2.1 specification, effectively admitting that CSS2 is obsolete. CSS2.1 also includes some features initially scheduled for CSS3, but it is mainly a "snapshot" of CSS usage at the time it was published (http://www.w3.org/TR/CSS2/), encompassing features that were widely implemented by browsers at the time the recommendation was published.

CSS2.1, therefore, was a temporary, transitional solution, to easy the pain of web developers at the time and provide guidance for browser implementors. Work on CSS3 has begun around the time the CSS2 recommendation was published. But, unlike CSS2, the CSS level 3 is not a monolithic specification; instead, requirements are classified into modules, and each module can have a different priority for W3C, and level of implementation by browsers.

For example, Level 3 CSS Backgrounds and Borders is currently classified as having "High Priority", and is a Candidate Requirement - which means it was widely reviewed, and satisfies W3C's technical requirements (http://www.w3.org/2005/10/Process-20051014/tr#q73); if implementation experience is deemed satisfactory, it will gain a Proposed Recommendation status. Currently, there is no document describing the CSS3 standard, but separate documents are available for each module. For example, the document for Level 3 CSS Backgrounds and Borders can be found here: http://www.w3.org/TR/css3-background/.

Browser can have different levels of support for different CSS Level 3 modules; browser implementors are encouraged to define compatibility with Level 3 on a per-module basis, instead of individual features, or CSS 3 as a whole. Browser implementors are more likely to concentrate on implementing modules, than individual features.