Jairo Jurado

Professor Kevin Imhoff

DWS

7 June 2013

CSS: Inheritance, the Cascade, and Specificity

When talking about inheritance in CSS, we are talking about the mechanism by which styles are applied not only to a specified element, but also to its descendants. For example, if I use the selector ul to change its color to red, the “children” of ul, the list items (li), would also change their text to red. (Meyer, p. 68-71) However, some properties are not inherited, “because it doesn’t make sense for them to be, which is similar to genetics” (Imhoff 2) There are also cases when the same selector has been used multiple times with multiple declarations. In this case, the one that will win will be determined by its specificity value; the one with the highest specificity value will win. A specificity value is expressed in four parts:“*style attribute*, *ID*, *class or pseudo-class attribute*, and *elements*”. For each of these encountered in the selector, we will add 1, and it will be written like: 0 (*style attribute)*, 0 (*ID*), 0 (*class or pseudo-class attribute*), 0 (*elements*). (Meyer, p. 62-63)

There are instances when two selectors have the same specificity value but they do different things. For example, if we have: h1 {color: blue;}, and h1 {color: green;}. How do we know which one will win? This is when Source Order is applied, “if two declarations affect the same element, have the same importance, and the same specificity, the final distinguishing mark is the source order” (Imhoff 10). What the Source Order says is whichever declaration that comes last in the style sheet will win. Source Order is important because there may be many lines of code and the developer made a decision later, perhaps forgetting that the same declaration was already applied, and automatically since the declaration was declared last, it will win over the other ones.

Before Source Order can be applied, rules have to go through a process called “The Cascade”, which comes from the name CSS (Cascading Style Sheets). “CSS is based on a method of causing styles to cascade together, which is made possible by combining inheritance and specificity” (Meyer 71). There three different types of style sheets: “*user agent/browser*, *author*, and *user*”. Every browser has a default style sheet; the web developer or designer implements the author style sheets; finally user style sheets are custom made that anyone can use. It’s possible to overwrite the specificity by increasing the specificity value or by using the cascade to make “!important” declarations. The order of !importance for these declarations is: “first, user agent declarations; second, author normal declarations; third, user normal declarations; fourth, author important declarations; and fifth, user important declarations” (Imhoff, p.16)

Works Cited

Imhoff, Kevin. “Lecture 2”. *Designing for Web Standards*. 2013. PDF file.

Meyer, Eric. *CSS: The Definitive Guide 3rd Edition*. O’Reilly Media, Inc., 2007. 62-72. Print.