What are some of the functions and purposes of CSS inheritance? In this paper I will discuss some of the ways on how functions and purposes of CSS work hand in hand. I will then describe what functions are and what are some of the purposes. With CSS and functions there are some foundational elements that need to be discussed. I wanted to first start off by discussing how they work together, but then I understood that in order for the reader to really understand this principal that they first need to know and understand how what functions are what are they doing with CSS CSS functions are values used in conjunction with certain types of properties.

What this means is that a function such as a property are used in conjunction with other elements to produce a desired effect. Some examples of the CSS functions are the float property. The float property describes where an element will float on a page, either the element will float left or right, none or inherit, which means that the floated element should be inherited from the parent element.

Specificity in CSS can be broken down into a number of ways. One way I will discuss specificity in CSS is by breaking down the different rules that make it up. The first rule that I want to discuss is how specificity tells the browser what rules to apply to certain elements. Second rule is that every selector has its’ own rules in specificity hierarchy.

To grasp at what is really going on here. I would say that with specificity you are telling the browser exactly what element needs to be styled.

How to calculate specificity. Starting with 0 and adding 1000 for style attributes, and add 100 for each id and 10 for each attribute, and for class or pseudo-class add one for each name or pseudo-element can calculate it. A good example of this would look something like this: body #content. Data img: hover, which would end up representing this:  (**0,1,2,2** or 0122).

How can I override specificity? One-way to do this is with the! Important rule, which basically means that when this rule is present that it will be applied no matter what, and this will in then override all other specificity. Source order is also important because with source order the last declaration that is made will always be the one that gets applied to your elements, so if you had multiple declaration that were the same, the one that is called last will be the one that gets applied.

The origin of stylesheets and how they are involved in cascade and how they are applied can be explained in a number of ways. The one way that I understand it was, explain how the author, user, and user agent play a major role in this process. Author. The author specifies style sheets for a source document according to the conventions of the document language. The User: The user may be able to specify style information for a particular document. User agent: Conforming user agents must apply a default style sheet (or behave as if they did) prior to all other style sheets for a document.

**Work Cited**

"CSS : Origin of Cascading." *W3resource*. N.p., n.d. Web. 12 June 2013

"Smashing Magazine." *CSS Specificity: Things You Should Know*. N.p., n.d. Web. 12 June 2013.

"CSS Float Property." *CSS Float Property*. N.p., n.d. Web. 12 June 2013.