

The Cascading Order

In this lesson we will learn the details of the cascading order.
Let's begin!

WE'LL COVER THE FOLLOWING ^

- The three steps of cascading

Now, you have a basic understanding of the syntax and semantics of style sheets.

When your page is displayed, all style rules that match an HTML tag are utilized to determine the visual properties of that specific tag.

While these rules set separate properties, it's easy to define the result: **all property settings are applied to the HTML tag**. Of course, the ones that are not applicable are simply omitted.

It may happen that multiple rules define values for the same property. While these rules define exactly the same property value, it does not matter which rule wins, because they all set the same value. However, most often these are *different* values and you must know which value is used.

CSS was designed with the intention to allow assigning multiple rules to the same element.

As you already know, **each browser has its own default style sheet** that sets a number of properties for each type of HTML tag.

If you add a style sheet to the page, regardless if it's an internal or external

style sheet, it is very likely that there are **competing styles**, because your style assignment may compete with the browser's default one.

Browsers use the cascading order to determine which rule in a set of competing rules gets applied.

The three steps of cascading

If there are competing rules, there are three main steps to determine which rule wins. These are the following ones:

1. The more specific selector (the selector with the highest priority) wins. The fundamental principle behind the cascade order is that there are **general selectors** that set overall styles for a document, and more **specific selectors** that override the general selectors to apply more specific styles. If the same property is set by two or more selectors, the more specific selector wins. If two or more selectors are equally specific, the competition goes on with the next step.

Earlier, Listing 9-2 demonstrated this situation with the following rules:

```
h1 { color: red; }  
.blue { color: blue; }  
#greenLabel { color: green; }
```



Let's assume you have an `<h1>` tag in your page like this:

```
<h1 class="blue" id="greenLabel">  
  This should be GREEN  
</h1>
```



Here, the most specific selector is the `#greenLabel` ID selector, so the heading's color is set to green.

2. The selector with a **higher specificity** wins. A specificity value can be calculated for each selector. If the same property is set by two or more selectors with the same specificity, the competition goes on with the last (third) step

(third) step.

Take a look at these rules:

 style.css

```
h1.spec { color: yellow; }  
.spec { color: blue; }
```



Let's assume, you have this `<h1>` tag in your page:

 index.html

```
<h1 class="spec">Heading</h1>
```



Here, the first selector is applied, and the heading is displayed with a yellow color. Both selectors are equally specific (both are **class** selectors), so the second step is used to determine the one with the highest selectivity. By this calculation `.spec` has lower specificity value than `h1.spec`, so `h1.spec` is the **winner**.

3. The selector with a higher priority location wins. If the first two steps do not announce a winner, the final decision is made by the location of the rule. According to the location where the selector was defined, the one with the highest priority can be found (there is exactly one of them), and the rule at this location wins the right to set up the particular property.

Now that we have covered the basic concept regarding the steps involved in cascading, let's look behind these steps in more detail from the *next lesson*.