

# Cascading & Inheritance





By Tristan Schmurr from Luxembourg, Luxembourg (Waterfall) [CC BY 2.0 (http://creativecommons.org/licenses/ by/2.0)], via Wikimedia Commons

# CSS *flows* like water.

It starts at the top and rushes over a series of plateaus. What matters most to us, though, is usually the pool of water at the bottom. The closer the plateau is to that pool, the more effect it has on the water that we can actually interact with.



#### We see this in two main ways

**Inheritance Cascading** 





### Inheritance.

Inheritance is the way in which properties flow from parental elements to child elements. So, body { font-family: Arial } will mean that Arial is also the font-family of body p, unless you define a different font-family for body p.





# Not everyone has kids.

But not every property inherits. It'd be crazy if backgroundcolor or margin were to inherit. If every single child element inherited background-color, you'd end up with a massive area of overlapping background colouring that would almost certainly not be desirable.



### Why inheritance rocks.

```
Imagine you want to have an unordered list of things
```

in HTML. So you define
#things as we see at the right.

By using inheritance, you only have to change the color in one place, at #things.

```
#things {
    background: #013c57;
    color: #fff
    }
#things li a {
    color: inherit;
    }
```

Of course, this is a simple example, so you're not saving that much time, but as your code complexity grows, inheritance, like The Force, becomes a powerful ally.





Cascading is related to inheritance. But its main function is to determine what to do when there is a conflict. "Winners" are decided by these three concepts.

Importance
 Specificity
 Source order

Solution Powered By WIKIA

#### The importance of style sheet order.

Imagine Fandom had a class called . yoda in its base CSS, like this: .yoda { color : red } but you created .yoda { color : green } in your local Wikia.css. What color is Yoda gonna be?

Importance (that is, the order in which style sheets are considered) agrees with the Star Wars Galaxy.

He's gonna be green.





#### Your wiki's stylesheets win over Fandom's.

Like our waterfall, the water that's most important to our experience is that which is closest to the pool at the bottom. Because Wikia.css is loaded after Fandom's base CSS, it's closest to the pool, so it "wins".

http://www.ForestWander.com [CC BY-SA 3.0 us (http:// creativecommons.org/licenses/by-sa/3.0/us/deed.en)], via Wikimedia Commons



#### Wikia.css

MediaWiki page



🖀 View source 🔻 Talk 15

margin-bottom: -5px:

Note: After saving, you may have to bypass your browser's cache to see the changes.

- Chrome Windows: Hold the *Ctrl* key and press the *F*5 key. OS X: Hold both the *# Cmd* and *☆* Shift keys and press the *R* key.
- Firefox Windows: Hold the *Ctrl* key, and press *F5*. OS X: Hold the *# Cmd* and *⊕ Shift* keys, and then press *R*.
- Internet Explorer Hold the Ctrl key, and press F5 (or click the Refresh button).

```
/*** button styling ***/
 .cc2-wikia-button a {
    -moz-border-radius:4px;
    -webkit-border-radius:4px;
    border-radius:4px;
    background-color:#002E54;
    background-image: -moz-linear-gradient(top, #004A87 35%, #002E54 65%);
    background-image: -webkit-gradient(linear, 0% 0%, 0% 100%, color-stop(35%, #00
    border:1px solid #002E54;
    color:white:
    cursor:pointer;
    display:inline-block;
    font: 12px/20px Helvetica, Arial, sans-serif;
    height: 19px;
    margin:0:
    padding:0 7px;
    text-decoration:none;
    white-space:nowrap;
.cc2-wikia-button a:hover {
    background-color:#002E54;
    background-image: -moz-linear-gradient(top, #004A87 35%, #002E54 65%);
    background-image: -webkit-gradient(linear, 0% 0%, 0% 100%, color-stop(35%, #00
    text-decoration:none
.blogsubhead {
    font-size: 14px;
    padding: 8px 0 0;
    font-weight: bold;
```

# Local style sheet pro-tip.

A good way to organise your stylesheets is to blank Common.css completely, and put it all in Wikia.css.

It used to make sense to put most of your code in Common, since that code would be used in Monobook. But since only a few thousand page views a day are now in Monobook, Common.css is less important these days than it once was.



#### Another local CSS pro-tip. Read Help:Including additional CSS and JS

Importing a local CSS file through importArticles:

```
importArticles({
   type: "style",
   article: "MediaWiki:Local.css"
});
```

@import method /Edit

Using the @import, styles can be imported from any CSS file hosted on a wiki or on other websites (ending with .css on the url and contains changes for the default skin elements on the file). The syntax is as follows:

#### @import "path\_to\_file\_to\_import.css"

Per the above example, to import Local.css into Wikia.css, place this at the top of your Wikia.css file (given load order and differences with Common.css loading, imports may need to be added to your Common.css file in order to work properly):

@import "/load.php?articles=MediaWiki:Local.css&only=styles&mode=articles";

An alternative way, which will make it easier to spot files loaded in the "Network" tab of Firefox's "Web Developer" is to use:

@import url("/MediaWiki:Local.css?ctype=text/css&action=raw");

It's possible to load a style sheet via Fandom's importArticle feature in JavaScript.

But this method is subject to JavaScript review — and it happens *after* your articles are initially loaded. That means your page will slightly "flicker" as they adjust to CSS imported via this method.

@import avoids these issues.





By Tomruen - Own work, CC BY-SA 4.0, https:// commons.wikimedia.org/w/index.php?curid=40673932

### **Specificity is next.**

More specific selectors "win" conflicts. So if you have

.yoda { color : red }

and

.yoda p { color : green }

in the same style sheet, the color of the p text within yoda class will be green.



### **Specificity is a calculation.**

#### **Specificity Calculator**

A visual way to understand <u>CSS specificity</u>. Change the selectors or paste in your own.



Different things are given numerical weights. And the highest number wins. We won't bore you today with the actual math, but there are tons of specificty calculators around.

#### A good one is at https:// specificity.keegan.st.







#### Source order.

Think of this as the default way in which most conflicts are solved through cascading. If you had . yoda defined in two places in Wikia.css, the one that's closest to the bottom — nearest the pool at the bottom of the waterfall - is the one that "wins".

By David.Monniaux (Own work) [GFDL (http://www.gnu.org/copyleft/ fdl.html) or CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/ 3.0/)], via Wikimedia Commons



#### Just say no to ! important

It can be tempting to use !important, as with
 .yoda { color : red!important }

This trumps everything we've just talked about. But you should avoid using this, because it will override any cascading and make errors a lot harder to catch.

If you want to ensure a property shows up (like color : green), the best way to do it is to make your selectors more specific.





### **Questions?**

