How far should you break up stylesheets?

Asked 16 years, 2 months ago Modified 15 years, 4 months ago

Viewed 5k times

15

I'm building a new site for my company, and I'm at the stage where I've created the html mockup of the first page. I'm going to use this as a basis for the rest of the site. I'm thinking of organising my stylesheet better now I've got the design looking consistent cross-browser, but I'm wondering how far to go when I'm breaking it up.

One idea is to have the following:

- reset.css
- typography.css
- layout.css
- colors.css

but where do I draw the line? theoretically I could go on and break them down into classes, ids etc, but I think thats going overboard.

Does this seem a reasonable method?

css

Share

Improve this question

Follow

asked Oct 21, 2008 at 14:44

Chris J Allen

19.2k ● 21 ● 79 ● 114

12 Answers

Sorted by: Highest score (default)

9

✓

Coincidentally, A List Apart had an article covering this today. They recommend separating out into a few main categories, including some you listed (type, layout, color), but going further to include various tricks to keep older browsers happy.

On the other hand, keeping everything in one css file keeps the requests between browser & server down. A compromise might be to keep things separate for development, and merging for production (as a part of your build process, naturally :p).

Share Improve this answer

Follow

answered Oct 21, 2008 at 14:49

swilliams

48.8k ● 27 ● 102 ● 130

5

I don't tend to split typography into a seperate stylesheet, although it seems like a good idea. I'd keep colours with typography though. My typical way is to have the following structure:

`base.css`

- Global style used throughout the site
- Aims to be extendable, for example so that it can be reskinned (but using the exisiting layout) for a microsite.
- Implements/imports `reset.css`

`page.css`

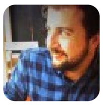
- Implements any page-specific changes.

`microsite_skin.css`

- See base.css point 2

Share Improve this answer Follow

answered Oct 21, 2008 at 14:49



Ross
47k ● 39 ● 123 ● 173

2

Pickledegg,

There's nothing wrong with breaking up style sheets. In fact, I find it very useful to organize css rules into different files based on their type, or what parts of the site they are applied to. If you lump rules into one large file, it can quickly become a mess and become very difficult to manage.

I would recommend coming up with your own scheme for separating your rules into files, and stick with it for all your projects.

Share Improve this answer

Follow

answered Oct 21, 2008 at 14:48

Rafe

9,235 ● 9 ● 53 ● 69

2

I would break layout down into 2+ more parts, Base layout, IE Hacks, Menus (one for each menu area. This could be something like a top menu and a side menu) If the site where to change color depending on the area I'd add one for each color area as well. You also could use Yaml or similar as a base framework for your layout. I'd keep the different stylesheets seperate while designing only merging them into 2-4 depending on the site shortly before uploading/releasing. always keeping the Hacks apart.

Share Improve this answer

Follow

answered Oct 21, 2008 at 14:53

Schalk Versteeg

1,057 ● 13 ● 23

2

Separate them based on what you estimate your needs are going to be later on. If you think the typography, layout, or colours (globally) are going to change, then it's probably wise to at least delineate styles in that way so it's easier to replace one stylesheet with another later on.

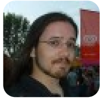
But if you go too far in this you'll end up with duplicate rules everywhere (eg. #content having a font-family rule in typography.css, a color rule in colors.css, etc). That's not a logical way to split things up unless you anticipate the key changes to be taking place there.

If, on the other hand, like most sites the graphic design is going to remain fairly static but the architecture is going to have some changes made (eg a new content type) then you want to be grouping your styles based on the context of the site. For instance, article.css, search.css, etc.

Essentially, try to look ahead at what changes are going to be needed later on and then try to anticipate those changes in your css file setup.

Share Improve this answer Follow

answered Oct 21, 2008 at 14:53



Rahul
12.2k ● 5 ● 45 ● 64

2

The major problem IMO, is the duplicate property definition issue. This makes style-sheets unmanageable. To bypass this issue the divide and conquer approach is used. If we can ensure manageable style sheets with non-conflicting rules, then merging sheets or modularising them would be easy.

During reviews all I do is check for rule conflicts, classitis and divitis. Using Firebug/CSSTidy combination, the problem areas are highlighted. Thinking in these lines, I don't even look into typography and font-separation.

The goal is to have a singel base CSS file and separate browser hacks files. Multiple base CSS files would be needed if an application has different themes.

Share Improve this answer

Follow

answered Oct 21, 2008 at 15:16

questzen

3,287 ● 20 ● 21

2

I put all styles, including IE6-and-7-specific styles, in one sheet. The IE6 and 7 styles are targeted using conditionally-commented divs that only appear if one of those browsers come to the site, e.g.:

```
<body>
<!--[if IE 7]><div class="IE IE7"><![endif]-->
<!--[if IE 6]><div class="IE IE6"><![endif]-->

... rest of markup ...

<!--[if IE]></div><![endif]-->
</body>
```

Not sure what people think of this approach, but the extra markup is negligible, and being able to include IE6/7 styles next to main styles without the use of hacks... just prepending the selector with ".IE" or ".IE6" etc, is a big convenience.

As for multiple stylesheets... save your HTTP requests. Use image sprites, one stylesheet, one "application.js" javascript, etc.

I'd still include separate sheets for the print and handheld styles, but that's about it...

Share Improve this answer

Follow

answered Oct 21, 2008 at 18:10

Steve Paulo
18.1k ● 2 ● 25 ● 23

1

Don't go too much further than that. If you do have to, try and find a way to merge them before production. The biggest issue is that you begin stacking up HTTP requests. It's not so much an issue for the browser but the amount of requests that need to be made for each page. I would say you are at a good point, more than 4 would be going somewhat overboard. Remember you can always use good commenting and formatting to break up large CSS files.

Share Improve this answer

Follow

answered Oct 21, 2008 at 14:48

Jeremy B.
9,206 ● 3 ● 47 ● 57

Good answer, though file caching softens the impact of the HTTP request issue. – Kon Oct 21, 2008 at 14:57

1

Whenever I'm trying to decide how far to break apart some files (I usually do this with code modules, but I apply the same principals to my css/js) I break it down to the smallest reusable files that make sense together. This isn't the best for the flow of data across the wire, but it makes maintainability of my source a lot easier. Especially if you're going to be having a lot of css floating around.

If you feel comfortable taking your colors.css and using the whole thing in another location without modification then you're probably fine.

Share Improve this answer

Follow

answered Oct 21, 2008 at 14:52

Jeffrey Martinez

4,594 ● 3 ● 33 ● 36

1

Take a look at blueprint-css. Blueprint is a CSS framework which supplies a variety of default styles to you (such as browser reset code).

The style sheets in blueprint-css are organized as follows:

- **screen.css** - default styles for screens
- **print.css** - default styles for printing
- **ie.css** - Internet Explorer specific fixes
- **application.css** - contains the styles you write. Actually you shouldn't modify the previous three style sheets at all, because these are generated by blueprint-css. You can overwrite all blueprint-css styles in your application.css file.

For further details refer to the blueprint-css Git repository.

Share Improve this answer

Follow

answered Oct 27, 2008 at 15:24

Christoph Schiessl

6,858 ● 4 ● 35 ● 45

1

I break mine up almost exactly the same way:

- reset.ccs - The standard reset from MeyerWeb
- typography.css - Fonts, sizes, line-heights, etc
- layout.css - All positioning, floats, alignments, etc.
- colors.css (or more appropriately skin.css) - All colors, background images, etc.

These are then followed with IE-specific files included via conditional comments for the appropriate version.

I haven't had the need to separate them further, though I do organize each file in separate sections (base elements, named elements, classes, etc.)

UPDATE: I have changed my system a bit in the last few projects I have done.

- base.css - Contains the CSS from the YUI reset and YUI base CSS files (with attribution)
- main.css - Contains @import statements for the following:
    - typography.css - Fonts, sizes, line-heights, etc
    - layout.css - All positioning, floats, alignments, etc.
    - theme.css - All colors, background images, etc.
- print.css - Overrides for the other CSS files to make the pages print-friendly.

I also then use conditional comments to add any IE-specific CSS files if needed.

My next step in improving this would be to add a build-process step to combine all of the CSS files into one file to reduce the HTTP requests on the server.

Share Improve this answer Follow

edited Aug 10, 2009 at 13:16

answered Oct 21, 2008 at 18:04

Raleigh Buckner
8,393 ● 2 ● 32 ● 38

1

Everybody recommends to break. I'll be the devil's advocate.

How about NOT breaking style sheets. At least for production purposes is better to have a single request instead of many. The browser can process 2 simultaneous HTTP requests. This means that other 2 requests can be made after the first 2 are completed.

Combined files are a way to reduce the number of HTTP requests by combining all scripts into a single script, and similarly combining all CSS into a single stylesheet. Combining files is more challenging when the scripts and stylesheets vary from page to page, but making this part of your release process improves response times.

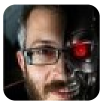
Yahoo and Google recommend this.

Google is recommending creating 2 files. One for everything necessary at the startup and 1 for everything else.

I think that even designers have to think in optimizing terms, not only hardcore developers.

Share Improve this answer Follow

answered Aug 10, 2009 at 13:25



Elzo Valugi

27.8k ● 16 ● 96 ● 115