

CSS: using comments the right way

The sad truth about CSS files is that they tend to become huge in size, despite all your efforts to keep them as small as possible. For that reason, comments play an important role when it comes to search and find something within a CSS file. However, there are no common conventions for writing consistent CSS comments. In this article I'll try to address this problem.

Comment blocks

Comment blocks generally should be written just before a group of rules that share common characteristics. For example, if you're writing the CSS rules for the header section of your site, then you could write something like that:

```
/* Header
 * - description -
 */
```

This comment block is not very useful written as is. How can you find it with your editor? It's simply a *generic comment block*. To make it more useful, you should add a marker to it.

Adding a marker to a comment block makes it more searchable. This is called *flagged comment block*. For example:

```
/*= Header
 * - description -
 */
```

In this case, the flag used is the equal sign (=). By using it as a search string in your editor, you can actually navigate through all your sections of your CSS file.

However, a simple marker like an equal sign may be inappropriate when you want to be more specific. In this case, you can add an identifier to the flag:

```
/*=head Header
 * - description -
 */
```

Now you can search for =head in your editor. This is called a *labeled flagged comment block*.

Another feature that you can add through comment blocks is *metadata*. A typical example of CSS metadata is the initial block put at the very beginning of every WordPress main CSS file (style.css):

```
/*
Theme Name: Twenty Eleven
Theme URI:
http://wordpress.org/extend/themes/twentyeleven
Author: the WordPress team
Author URI: http://wordpress.org/
Description: ...
Version: 1.3
License: GNU General Public License
License URI: license.txt
Tags: ...
*/
```

Metadata can also be described by using PHP/JSDoc-like keywords:

```
/*=head Header
 * @author Gabriele Romanato
```

```
* @since 1.0
*/
```

Since only a very small subset of these keywords fits the CSS standard, this kind of metadata inclusion is somewhat limited.

Inline comments

CSS supports only the multiline C-like comments, but some CSS frameworks also allow for C-like single-line comments. We should use inline comments when we have to explain the meaning of a rule or declaration. For example:

```
#site {
    width: 100%; /* Responsive width */
    max-width: 50em;
}
```

If more than one developers modifies the file, then it would be useful to specify the author as well:

```
#site {
    width: 100%; /* Responsive width @author John
Doe */
    max-width: 50em;
}
```

Coding style

Nicolas Gallagher writes:

Well commented code is extremely important. Take time to describe components, how they work, their limitations, and the way they are constructed. Don't leave others in the team guessing as to the purpose of uncommon or non-obvious code.

Comment style should be simple and consistent within a single code base.

- Place comments on a new line above their subject.
- Avoid end of line comments.
- Keep line-length to a sensible maximum, e.g., 80 columns.
- Make liberal use of comments to break CSS code into discrete sections.
- Use "sentence case" comments and consistent text indentation.

Tip: configure your editor to provide you with shortcuts to output agreed-upon comment patterns.

Whatever coding style or format you choose, the golden rule is to keep it consistent throughout your code.