CSS Coding Conventions

Naming Conventions

Names

The names of the classes need to be clear and descriptive. Names should not contain style-information.

Case

First for the case of the names themselves. Start the class names with lowercase letters and start each embedded word in the name with a capital. HTML elements are in lower case in anticipation of XHTML.

Spacing, Punctuation and Brackets

Generally spacing promotes readability, so let's take advantage of it.

Empty lines between style definitions

Allow the separate style definitions to breathe, by adding one (and only one) empty line between them.

1 tab = 4 spaces

Indentation also helps with the readability of the code. So put every property-value pair on a new line and indent it with 4 spaces from the beginning of the line. You can use a tab for indentation, it doesn't matter, this is just a personal preference.

Space after colons

Visually emphasis on the separation between a property and value, by adding a space after the colon, like:

```
.class {
   prop: value;
}
```

… and not

```
.class {
   prop:value;
}
```

… or

```
.class {
   prop:      value;
}
```

Curly brackets

Curly brackets embrace all property-value pairs in a style definition. The opening bracket must be on the same line as the selector identifier and the closing bracket on a new line.

Example

```
.header {
  font-size: 2em;
}

.navigationBar,
.content,
.header i {
  font-size: 0.8em;
  border: solid black 1px;
  border-left: solid black 5px;
  border-right: solid black 5px;
}
```

Structure

When several style definitions share some styles and are of the same *family*, their common styles should be defined in one place.

Example

```
/* Common message styles */
.normal-message,
.warning-message,
.error-message {
        border-style: solid;
        border-width: 1px;
        padding: 5px;
        position: relative;
        width: 500px;
}

/* Specific message styles */
.normal-message {
        background-color: #E5EDF9;
        border-color: #4171B5;
}

.warning-message {
        background-color: #FFEFA7;
        border-color: #FFB30F;
}

.error-message {
        background-color: #FFD1D1;
        border-color: #FF0000;
}
```

Shorthands

Use shorthand properties to keep all parts of a style type on a single line.
For example:

```
        margin:5px 0 4px 10px;
```

Is much more efficient than:

```
        margin-top:5px;
        margin-right:0;
        margin-bottom:4px;
        margin-left:10px;
```

Combining properties onto a single line using shorthand properties means that your CSS will be easier to understand and edit.

Sections

Clearly divide your stylesheet into specific sections by using a seperator like this:

```
/*
========================================
            Global Styles
========================================
*/
```

Alphabetize Your Properties

Keep your properties organized by alphabetizing them.

Comments

Comments begin with the characters "/*" and end with the characters "*/". They may occur anywhere between tokens, and their contents have no influence on the rendering. Comments may not be nested.

Bronvermelding

[1] CSS Coding Style
http://radio.weblogs.com/0103451/stories/2002/02/16/cssCodingStyle.html

[2] Suggestions for Organizing Complex CSS
http://www.smileycat.com/miaow/archives/000153.html

[3] 5 Tips for Organizing Your CSS,
http://erraticwisdom.com/2006/01/18/5-tips-for-organizing-your-css

[4] CSS Coding Conventions
http://www.phpied.com/css-coding-conventions/