**CS 213 Style Guide - Group 2**

**Introduction:**

This is the Group 2 style guide for CS 213. It pertains to HTML and CSS. Below are the team roles and the sections of the style guide. The guide is split into five main style sections, which have been bolded. They are:

* Readability
* Editor
* CSS vs. Direct Formatting
* Markup vs. Presentation
* Copyright

**Readability:**

Making your code readable is important because someone else will always see your code. We want to make it so that it can be easily understood what your solution to the task assigned. If there is problem later or something needs to be changed you may not be the one working with the code. There are a few ways that you can make your code more readable and easier to understand.

We need to have consistent indenting, if you change indenting styles in your code it will become difficult to follow. Pay attention to how you indent each element, nested indentations help to clear up the markup hierarchy and makes it easier to understand by quickly looking at it. Avoid deep nesting and keep it simple by grouping similar tasks together.

This is an example of good indentation:

<!DOCTYPE html>

<html>

<head>

<title>First page</title>

</head>

<body>

<p>Hello world!</p>

</body>

</html>

When we write our code, we should keep lines to about 80 characters long. If your code goes to edge of the text editor, the code can wrap around to the next line below or not wrap, and the reader is forced to scroll sideways to see what you had written. It is much easier to just see what was done in smaller lines.

<!DOCTYPE html>

<html>

<head>

<title>First page</title>

</head>

<body>

<!—We want to avoid long lines so that they don’t wrap around to the next line-->

<!—If our comment is long then continuing to the

next line and indenting is best-->

<p>Hello world!</p>

</body>

</html>

Comments in your code can help with explain your processes and how your code will work through tasks. Use only lowercase throughout your code in html it looks more consistent in the code to keep it all the same if possible. Be sure to remove trailing white spaces and unneeded blank lines in each element in your code, they just take up unneeded space in the file and bloat the file size in the memory storage.

<!DOCTYPE html>

<html>

<head>

<title>First page</title>

</head>

<body>

<!-- Write your comments here -->

<p>Hello world!</p>

</body>

</html>

References:

Guzel, Burak. "Top 15+ Best Practices for Writing Super Readable Code." Code Envato

Tuts+. Envato Market, 30 Mar. 2011. Web. 28 Apr. 2017. <https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118>.

"Google HTML/CSS Style Guide." Google HTML/CSS Style Guide. Google, n.d. Web. 28 Apr.

2017. <https://google.github.io/styleguide/htmlcssguide.html>.

["HTML5 Tutorial." HTML Tutorial. W3 Schools / Refsnes Data, 1999. Web. 28 Apr. 2017.](https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118)

<https://www.w3schools.com/html/default.asp>.

**Editors:**

There are many text editors out there that programmers can use to code their projects. We will be looking at *Emacs*, *Notepad++*, *Brackets*, and *Dreamweaver*. With almost any product there will always be Pros and Cons, so that is how we will look at these text editors.

Emacs

The first text editor we wanted to review is called Emacs. We wanted to start with Emacs since we have been using Emacs since we started programming at BYU-Idaho. Emacs is a free to use text editor that can debug, compile, and manage files. Since Emacs can provide those features it is classified as an IDE. An IDE is an “Integrated development environment that consists of a source code editor, build automation tools and a debugger.” Emacs is customizable by using Emacs lisp code or by using the graphical interface. The editor provides the ability to download and install extensions as well.

There are some issues using Emacs though. With any program or editor there will be a learning curve, and Emacs is no exception. One of the biggest issues we see with using Emacs is the lack of mouse support. All commands and movements in Emacs must be used with keyboard commands. For example, a user cannot use their mouse to move to a certain line. Another issue with using Emacs is the lack of line numbers. If there is an error in the program a user must count through the lines, or they need to know their code well enough to know where the error could be.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Free | Learning Curve |
| IDE (Debug, and compile) | No mouse support |
| Syntax Coloring |  |
| Customizable |  |
| Download & install extensions |  |

Notepad++

Notepad is a program most users have heard of since it is pre-installed on every windows computer, but there is a version for programmers. The program must be downloaded, but it is free. Notepad++ supports several different languages for several types of programmers. There is a new feature called PCRE (Perl Compatible Regular Expression). PCRE is a search and replace function, which means a user can search for a specific word and replace every instance of that word. Most programs already have that feature, so it’s nice to see Notepad++ adding it to their program. The graphical interface is entirely customizable to how the user likes to work. Notepad++ can give the user hints as well. These hints can be used to finish words, functions, and function parameters.

The main issue that we have with Notepad++ is that the program is only supported in Windows, so MAC users will have to look somewhere else for their text editor. Also, Notepad++ makes the user learn how to code. The program does provide hints to the user, but the user must be the one to know how the code works. We wanted to bring this up here, but we will talk more about that feature when we review Dreamweaver.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Free | Windows support only |
| Supports several languages |  |
| PCRE (Search/Replace) |  |
| GUI customizable |  |
| Tab interface |  |
| Word & function hints |  |

Brackets

Brackets is a free to use text editor that is lightweight and easy to use. The editor has in inline editor feature, so the user does not have to open a new tab to code. An excellent feature Brackets implemented is Live Previews. The user can see their website in real time, which can help save time. Brackets also has a “Replace All” feature that Notepad++ is starting to use. This feature makes it easy for a user to find one word and change every instance of it in the code. Brackets has extension support as well.

We noticed that there is limited language support for Brackets. This editor is only for users that program using a certain language. Brackets is an open-source software, so there is a chance more languages will be provided soon.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Free | Only supports HTML, CSS & JavaScript |
| Inline Editor |  |
| Live Preview |  |
| “Replace All” |  |
| Extension support |  |

Dreamweaver

Dreamweaver is a feature rich editor, but is it practical? A feature we like in a text editor is code hints, and Dreamweaver does that for us. Another feature of Dreamweaver is real-time error checking. While a user is coding Dreamweaver will alert the user of a possible error in their code. Dreamweaver shares a feature that Brackets has, and that feature is real-time previews. A user can see what they are coding as they are coding it.

We believe a good editor is one that can provide hints while we code. As students, we are learning new languages that can be difficult at times, so a text editor that helps us code is a plus. However, Dreamweaver may not be the best for a student to use. Dreamweaver basically codes for you. A user that is learning to code may want to stay away from Dreamweaver because the program does not help you learn a language. A student wanting to learn a language may want to find a different editor. Also, Dreamweaver is a paid editor. A student can get a discount on the product, but there are free editors out there that may even be better for a student.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Code hints | Paid |
| Real-time error checking | Codes for you |
| Preview in real-time |  |

References:

“A modern, open source code editor that understands web design.” *Brackets.* N.p., n.d. Web. 27 Apr 2017 (<http://brackets.io/>)

“Adobe Dreamweaver CC.” *Adobe*. N.p., n.d. Web 27 Apr 2017. (<http://www.adobe.com/products/dreamweaver.html?sdid=KKQTJ&mv=search&s_kwcid=AL!3>)

“Emacs vs Notepad detailed comparison as of 2017.” *Slant*. N.p., n.d. Web. 27 Apr 2017 (<https://www.slant.co/versus/43/44/~emacs_vs_notepad>)

“Features.” *Notepad Features.* N.p., n.d. Web. 27 Apr. 2017. (<https://notepad-plus-plus.org/features/>)

“Integrated development environment.” *Wikipedia.* Wikimedia Foundation, 25 Apr. 2017. Web 27 Apr. 2017 (https://en.wikipedia.org/wiki/Integrated\_development\_environment)

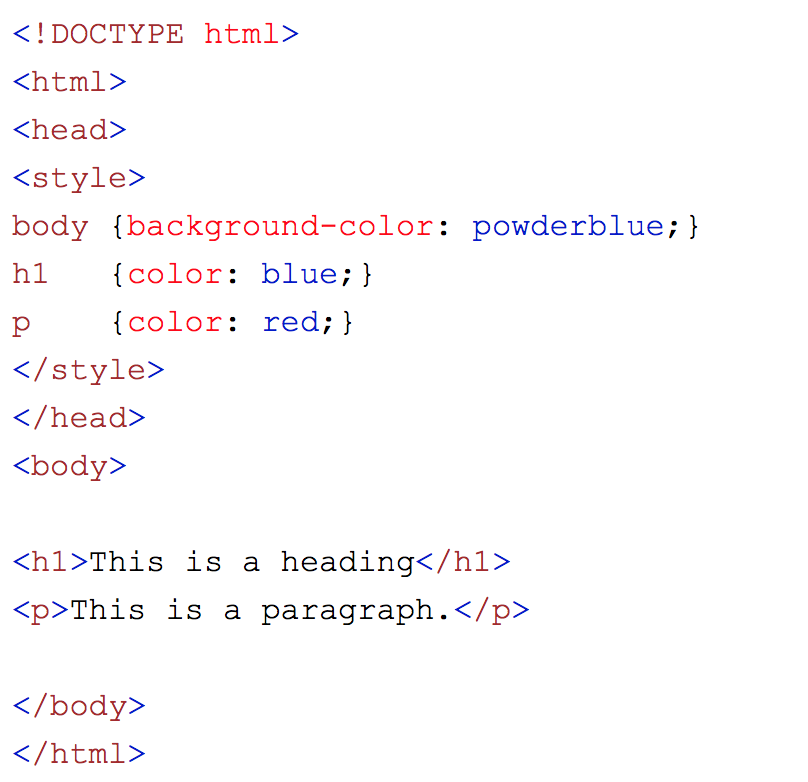
**CSS vs. Direct Formatting:**

CSS and direct formatting both have strengths and weaknesses. Our research has led us to a few conclusions about what they each are and how/when to use them.

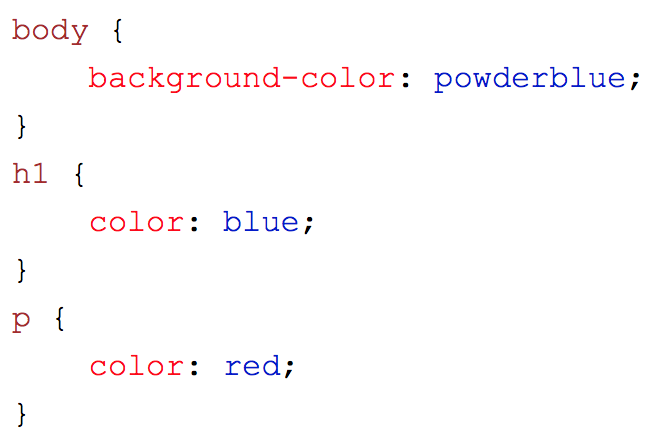
What is CSS?

CSS stands for Cascading Style Sheets. W3Schools describes the role of CSS as a “[description of] how HTML elements are to be displayed on screen, paper, or in other media.” It can be written inline using a style attribute, internally by using the <style> element in the <head> section of a HTML document or externally by using an external CSS style document. It is contained within the HTML document itself. Examples of inline of CSS are:

Screenshot 2017-04-27 16.30.09.png



An example of an external CSS style sheet is:



The difference from inline flavour: it is not in the HTML document like the internal CSS. Rather, it is referenced by the HTML



Below, we have included a small table of pros and cons of CSS

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| Speed of Design: CSS helps web engineers create multiple pages with the same style very quickly | Speed: Downloading HTML documents with CSS embedded can slow the load speed of a web page. |
| Consistency: CSS helps build a framework that can cross all pages of a website. | Weirdness: CSS has a syntax that is very different from the HTML it supports. It is not the most user-friendly and can be tricky to learn how to integrate it and HTML together. |
| Ease of Use: CSS is easy to learn with a large number of tutorials and help forums online. | Complications: CSS gets messy and difficult when using third-party software, such as Dreamweaver, to create webpages. |
| Multiple Browser Support: CSS supports Safari, Internet Explorer, Chrome, Firefox, etc. |  |

Overall, despite CSS being very different from HTML and being somewhat tricky to use on third-party software, it is the preferred and recommended way to style HTML documents. As a general rule, we will be using CSS as a group when doing group and individual projects.

What is Direct Formatting?

Direct formatting is a way of formatting text and other items on a webpage directly in a HTML document. An example of this might be <strong> (important text), <i> (*italics*), etc. Direct formatting is very quick when formatting just one or two things (mainly text). The more formatting required, the harder direct formatting becomes because it is not a consistent method of style for a document. It is hard to replicate the formatting and if you want to make a simple change to multiple pages it will require a lot of effort and time. Overall, direct formatting is not an effective way to style HTML documents.

References for CSS, HTML and Direct Formatting:

“HTML Styles.” *HTML Styles.* N.p., n.d. Web. 28 Apr. 2017.

<https://www.w3schools.com/html/html_styles.asp>

"HTML Styles - CSS." *HTML CSS*. N.p., n.d. Web. 28 Apr. 2017.

<https://www.w3schools.com/html/html_css.asp>

"HTML Text Formatting." *HTML Text Formatting*. N.p., n.d. Web. 28 Apr. 2017.

<https://www.w3schools.com/html/html_formatting.asp>

"The Pros and Cons of CSS." *Sanford-Brown*. N.p., 24 Jan. 2015. Web. 28 Apr. 2017.

<http://www.sanfordbrown.edu/Student-Life/blog/January-2015/Pros-and-Cons-of-CSS>

**Markup vs. Presentation:**

Markup defines the HTML tag that specifies a Header in the page. The presentation portion of the tag is what changes the features of the text or section. As important as presentation can be in making the web page unique and have a greater value to the audience the more important portion of the tag and code is the markup. The markup will allow for an easier readability of the code and what each section is for editors and other programmers that will help with the HTML.

Once the markup is correct and clear others helping with the programming with can assist in the presentation side of things if needed. Both Markup and Presentation have equal importance with different audiences in mind. Markup adds effect for the client but the presentation is what really helps the audience receive a better viewing experience.

References:

Knierim, Thomas. "Semantic vs. Presentational HTML." *Thomasknierim.com*. N.p., 14

Feb. 2008. Web. 28 Apr. 2017. <http://www.thomasknierim.com/62/web-development/semantic-versus-presentational-html/>

**Copyright:**

Copyright is something that we always need to consider when coding. This especially pertains to web engineering because the code written is easily accessible to everyone on the internet. Based on our research we have determined a few things:

1. Commenting and documentation are extremely important. This allows you to recognize people, sources, etc. and give them credit. It also others to properly cite your work if they used pieces of it.
2. Making sure that the work being used or referenced to in projects and assignments is valid under the fair use law. For more information on that please see the reference to copyright.gov below.
3. The class policy is as follows,

“Students are encouraged to work with your classmates to learn

technologies and share ideas, but all submitted work must be original. Share ideas; DO NOT SHARE CODE! The penalty for copying or plagiarizing of assignments might be one or more of the following: a 0 on an assignment, being asked to withdraw from the class, a failing grade in the class, or disciplinary action by the University.”

These guidelines have lead us to the conclusion that looking at code from others, using developer forums and citing it is ok when permission is given. Directly copying and pasting code or even typing exactly what others have done, even if typed and not pasted, is in violation of copyright. Our group will work together, share ideas and solutions to problems but not simply share code.

References:

Beckstead, Ryan. “CS 213 Syllabus” N.p., n.d. 28 Apr. 2017

<https://byui.brightspace.com/d2l/le/content/242529/viewContent/3601265/View>

Guzel, Burak. "Top 15+ Best Practices for Writing Super Readable Code." Code Envato

Tuts+. Envato Market, 30 Mar. 2011. Web. 28 Apr. 2017. <https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118>

Office, U.S. Copyright. "More Information on Fair Use | U.S. Copyright Office." *Copyright*.

N.p., Apr. 2017. Web. 28 Apr. 2017.

<https://www.copyright.gov/fair-use/more-info.html>