CSS Cheat Sheet

Rule

The building block of a CSS stylesheet. A rule consists of a selector and a declaration block (one or more declarations).

Declaration

A declaration is made up of a property and a value, separated by a colon and punctuated by a semi-colon.

Selector

The actual HTML object that the declaration(s) apply to.

Property

The characteristic of the selector that will be changed.

Value

The amount or type of change to be applied to the corresponding property of the matched selector.

Marking-Up

The process of assigning HTML tags to given text content in order to indicate its relation to the rest of the text or dictate how it should be displayed.

Serif Font

One of two general categories of fonts (typefaces) that uses marks (called "serifs") to embellish characters at the end of strokes. A common serif font is Times New Roman.

Sans-Serif Font

One of two general categories of fonts that do not embellish characters at the end of strokes ("sans serif" means "without serif"). A common sans-serif font is Helvetica.

Class

A class attribute is added to an HTML element in order to provide a "hook" to refer to that element in your CSS. CSS class selectors begin with a .. Classes can be used multiple times per page and are used for styling groups of elements.

ID

An ID attribute is added to an HTML element in order to provide a "hook" to refer to that element in your CSS. CSS ID selectors begin with a #. Unlike classes, IDs can only be used one time per page.

Classes vs. IDs

Classes and IDs are ways of targeting the style of specific HTML elements on your page.

So, what's the difference between them? In short, classes allow you to style many elements with a particular style, while IDs are only capable of styling a single element.

Using these selectors allows you flexibility and control of styling individual, as well as groups, of elements on your page.

Horizontal Rule

Add a horizontal line across your page using the `hr` tag.

Line Break

Break up a block of text using the `br` tag.

Image

Add images to your HTML using the `img` tag. Tell the browser the source of the image file with an `src` attribute.

Absolute File Path

A path to a website or file that includes a full web address (starting with "http") that the browser loads from the remote location directly. For example:

```
<img src="http://imgur.com/awesomedog.jpg">
```

Relative File Path

A path to a website or file that gives you the path to the resource you are looking for as it relates to your website's local file structure. For example, if you wanted to retrieve an image called "newlogo.png" that resides in a directory called `images`, you would enter the following relative address:

```
<img src="images/newlogo.png">
```

Why Separate HTML from CSS?

Separating HTML from CSS offers you scalability and versatility. If you separate how your site looks from what your site says, things become more flexible. You can make a change in one place and have it apply to your whole site, and apply any number of different styles to the same content.

CSS Color Treatment

While color names are fine when you're just beginning, there's a number of reasons you'll want to switch over to something more advanced. First, color names are rendered differently by different browsers. Second, there are only 147 color names accepted as standard, meaning your options are pretty limited.

Instead, you'll want to use either RGB or hexadecimal codes. Both of these are built on a system of entering values for the colors red, green, and blue.

By mixing different intensities of these three colors, you can create millions of different colors and shades. Intensity values range from 0 (no intensity) to 255 (full intensity) in the RGB system.

In hex, they range from 0-9, then continue from A-F, with two characters each for red, green, and blue. This is clearer with examples, so here are some common colors with their RGB and Hex equivalents.

The format for color names, RGB, and hexadecimal should look like the following, respectively:

```
p {
    color: red;
}
```

```
p {
    color: (255,0,0);
}
```

```
p {
    color: #FF0000;
}
```

CSS Text Treatment

font-family

To adjust the font of your selected text element, use the font-family property. For the value, enter the name of the font to which you'd like to alter your text. To be safe, put a comma after your selected font and enter a generic family as a fallback. If the web browser doesn't support the font you selected, it will choose the fallback.

```
h1 {
    font-family: Arial, sans-serif;
}
```

If your selected font is more than one word, capitalize both words and put them in quotation marks.

```
h1 {
    font-family: "Courier New", monospace;
}
```

font-size

To increase or decrease the spacial dimensions of your chosen text, use the font-size property. As a beginner, you should enter pixel values for your font-size values.

```
h3 {
    font-size: 24px;
}
```

As you become more advanced, try using percentages or ems instead of pixels.

font-weight

To adjust the thickness of your selected text, use the font-weight property.

As a beginner, you can enter values like "normal" to make your text thin and "bold" to make your text thick. As these values aren't very specific, different browsers may interpret their display with slightly different outputs.

```
h1 {
    font-weight: normal;
}
```

```
h1 {
    font-weight: bold;
}
```

As you become more advanced, try using the numbers 100, 200, 300, 400, 500, 600, 700, 800, and 900 as values to gain more granular control. With this system, 400 is roughly equivalent to "normal" and 700 roughly equals "bold."

```
h1 {
    font-weight: 400;
}
```

```
h1 {
    font-weight: 700;
}
```

font-style

To make normal text italic, use the property font-style and the value "italic." To reverse this effect, use the value "normal."

```css
a {
    font-style: italic;
}

a {
    font-style: normal;
}
```

text-align

To adjust the positioning of a text element, use the CSS property "text-align" and one of the following values: left, right, center, or justify.

```css
body {
    text-align: center;
}
```

text-decoration

To add an underline to normal text, use the CSS property "text-decoration" and the value "underline."

```css
h1 {
    text-decoration: underline;
}
```

To remove underlines, use the value "none." This declaration is often applied to anchor tags.

```css
a {
    text-decoration: none;
}
```

Less commonly used values include "overline" to add a line above text and "line-through" to strike a line through your text.

text-transform

To adjust capitalization in a selected text element, use the "text-transform" property.

Values for this property include "uppercase" to make every letter capitalized, "lowercase" to make every letter uncapitalized, and "capitalize" to make the first letter of every word in the selected text capitalized.

```css
h1 {
    text-transform: uppercase;
}
```

Adding Color

1. Change the value of the `#name div` from `white` to `red`.

   ```
   #name {
       background: red;
   }
   ```

2. Turn the `#hex div` red by leaving the first two characters at their maximums of `ff`, and changing the last four characters to their minimums of `0`.

   ```
   #hex {
       background: #ff0000;
   }
   ```

3. Turn the `#rgb div` red by leaving the red value at its maximum of `255`, and changing both the green and blue values to their minimums of `0`.

   ```
   #rgb {
       background: rgb(255,0,0);
   }
   ```

4. What color do you think you'd get if you set all the RGB values to their minimum of `0`? Can you set all three boxes to that color?

   ```
   #name {
       background: black;
   }

   #hex {
       background: #000000;
   }

   #rgb {
       background: rgb(0,0,0);
   }
   ```

5. Bonus: Now, let's work with the alpha value in the RGB model.

   ```
   #rgb {
       background: rgba(0,0,0,.5);

   }
   ```

```css
#rgb {
    background: rgba(0,0,0,.8);
}



#rgb {
    background: rgba(0,0,0,.2);
}
```

Working with Images

1. Take a look at the starter code below. Yep, it's your HTML boilerplate!
2. Write in an `img` tag between the body tags.

```html
<img src="">
```

3. Add `src="http://i.imgur.com/z9gGd0t.jpg"` to your image tag. (Grumpy Cat has strong opinions about visual design!)

```html
<img src="http://i.imgur.com/z9gGd0t.jpg">
```

4. Add `alt="grumpy cat hates bad design"` to your image tag.

```html
<img src="http://i.imgur.com/z9gGd0t.jpg" alt="Grumpy Cat hates bad design">
```

5. Add `title="grumpy cat"` to your image tag.

```html
<img src="http://i.imgur.com/z9gGd0t.jpg" alt="Grumpy Cat hates bad design" title="grumpy cat">
```