Design Reading List

Design

In small teams, the designer is also responsible for the CSS. Therefore, this reading list contains some resources on design.

If you are a designer, here are some good news: CSS is **not** a programming language, so you do **not** need to be a programmer to understand and use CSS.

The fundamentals of web design are: **Typography, Layout,** and **Color**.

Typography

To get started with typography, read through http://www.threestyles.com/tutorials/css-tips-for-better-typography/, which has some general rules to help with type layout, as well as visual examples of different font properties.

It is **very easy** for beginners to go overboard and decorate their websites with a bunch of different crazy fonts. Don't fall into this trap! Three very simple rules you can follow are:

- No more than **three fonts** per page. This includes variations on one font, such as changing size or weight.
- Use **sans serif** fonts (Arial, Helvetica, etc.) for headers. Sans serif fonts tend to be bolder, more eye-catching, and more legible from a distance.
- Use **serif** fonts (Times New Roman, Georgia, etc.) for paragraphs. Serifs make it easier to transition between words, leading to a smoother reading experience. (However, at smaller sizes and lower resolutions, this benefit is erased.)

http://www.webdesignfromscratch.com/basics/readability/ offers an interesting read on contrast and readability of websites.

There are some standard, "web safe" fonts that most computers can display. However, if you do wish to look for more exciting fonts that better match your website's theme, https://typekit.com/ and http://www.google.com/webfonts are good places to start.

Layout

Positioning elements on a page: http://www.alistapart.com/articles/css-positioning-101/

Floats are often called "viral", because adding *float* to a layout usually solves a problem at the cost of introducing a new problem, which is also solved by adding another *float*, and so on, until (almost) everything ends up floating. Nevertheless, floats are very helpful when used sparingly.
http://www.alistapart.com/articles/css-floats-101/

Guidelines for figuring out layout: http://www.alistapart.com/articles/the-web-aesthetic/

Color

The Example section in MDN's color reference shows what colors look like in CSS.
https://developer.mozilla.org/en-US/docs/CSS/color

Designers just magically know what colors they should use, and they can play with Adobe Kuler to figure out what color they want, and get the CSS notation for it. Spend 5 minutes creating colors with it, and get a feel for how it works.
https://kuler.adobe.com/ (the HEX color is used in CSS)

We, programmers, generally have really bad color intuition. When left to our own devices, we tend to build sites that look like http://jimvarney.org . Fortunately, we can use math and color theory to keep our intuition in check.

Color Schemes

Throwing all the colors you like onto a webpage is a surefire way to make a poor-looking website. Color schemes teach us which colors work well together, and help us restrain ourselves when choosing colors. (skim the links)
http://en.wikipedia.org/wiki/Color_scheme
http://en.wikipedia.org/wiki/Color_wheel

Go to Adobe Kuler, *Create* > *From a Color*, and try all the options under *Select a Rule*. These are tried-and-true methods for generating color schemes.

For best results, you should use a Monochromatic or Complementary scheme for your first site. (skim the links to understand some details)
http://en.wikipedia.org/wiki/Monochromatic_color
http://en.wikipedia.org/wiki/Complementary_color

Color Models

Color models help you mix colors, and are most often used for generating colors within the constraints of a color scheme. Color models turn a color such as blue into numbers such as (0, 0, 255). Two colors are mixed by doing some arithmetic on their numbers.

The most popular color model in CSS is RGB. (skim the links, look at the pictures)
http://en.wikipedia.org/wiki/RGB_color_space
http://en.wikipedia.org/wiki/RGB_color_model

RGB is very close to how monitors build colors. Unfortunately, it is not very useful for humans, because RGB colors rarely combine in meaningful ways. Try using Kuler's color picker in RGB mode, and see how unitunitive it is.

CSS3 also supports the HSL color model. Spend 5 minutes playing with an HSL color picker, and notice that the three sliders (Hue, Saturation, Lightness) work in a more intuitive way than RGB sliders.
HSL picker: http://hslpicker.com/
RGB picker: http://developer.yahoo.com/yui/examples/slider/slider-rgb.html

To understand a bit about HSL, skim the links and look at the pictures.
http://en.wikipedia.org/wiki/HSL_and_HSV

HSL is a good aid to human color pickers. If you plan to mix colors algorithmically, you should know that HSL has some issues. The link below will take you deeper down the rabbit hole.
http://vis4.net/blog/posts/avoid-equidistant-hsv-colors/