latex input: mmd-article-header Title: CSS Notes Author: Ethan C. Petuchowski Base Header Level: 1 latex mode: memoir Keywords: HTML, CSS, programming language, syntax, fundamentals, web programming, web design CSS: http://fletcherpenney.net/css/document.css xhtml header: copyright: 2014 Ethan Petuchowski latex input: mmd-natbib-plain latex input: mmd-article-begin-doc latex footer: mmd-memoir-footer

Useful properties

Align the text, like in a Word doc

```
text-align: center;
```

Set the background color of the whole page

```
background: black;
```

Set the background to an image

```
background: url("url");
```

Center the background image on the page

```
backgound-position: center;
```

Prevent the background image from being "tiled"

```
background-size: cover;
```

Set the font color

```
color: white;
color: #f00; // the order is red-green-blue ("RGB")
color: rgba(255, 0, 15, 0.5); // if you want to use "alpha"
```

Set the font

```
font-family: helvetica;
```

Set the font size

```
font-size: 22px;
```

Change the default border on things like `<input>` and `<table>`

Styles: `solid, dashed, dotted, double`

```
border: width [style] [color];

border: 0;
```

Adjust the border for each side of the element individually

```
border-left: width style color;
border-top: ...
etc...
```

Make the corners *rounded*

```
border-radius: 20px;

border-radius: top_left top_right bottom_right bottom_left;
```

Create a perfectly circular curve

Make the radius at least *half* the length of the shape's edge.

Set the width of the element

```
width: 500px;
```

Allow the width to shrink on smaller screens but not grow

As mentioned below, this makes the site more *responsive*

```
max-width: 500px;
```

Break an element out of the document's regular flow

```
position: absolute;
```

Make height changes appear in a 1/2 second *animation*

```
[-webkit|moz-]transition: height .5s;
```

Add a radial gradient

Shapes: `circle, ellipse`

(It looks like the *vendor prefix* isn't necessary for my Chrome, but it *is* a CSS3 thing)

```
background: [-webkit|moz-]radial-gradient(shape, center color, outer

background: radial-gradient(circle, white 15%, transparent 40%), #cc
```

Media Queries

These allow us to set styles that only activate when the browser is a certain width

This one will shrink the header and make the navbar vertical if the browser width is *smaller than 500px*

```
@media (max-width: 500px) {
  h1 {
    font-size: 36px;
  }
  li {
    display: block;
    padding: 5px;
  }
}
```

Rough around the edges

Padding

Intro

- The amount of space between the edge of the element and the stuff *inside* it
- This is super useful for making `<form>` elements look better

E.g.

```
padding: 10px;
```

Customizing

- When you just enter one number, it adds that spacing on *all sides of the element*
- Sometimes you want to dictate what you want for each side

So you can do

```
padding: top right bottom left;
```

Which has the **mnemonic**:
Never **E**at **S**hredded **W**heat

```
padding: Never Eat Shredded Wheat;
padding: North East South West;
```

For example:

```
padding: 0px 10px 0px 10px;
```

Or alternatively you can specify **2 sizes**

```
padding: top-and-bottom left-and-right;

padding: 0px 10px;
```

Margin

Intro

- The amount of space between the edge of the element and the space *outside* it (compare with padding, which creates space *inside*)

More

Center the content *within its element*

The first piece is `top-and-bottom` the second is `left-and-right`

```
margin: 0 auto;
```

Padding vs Margin

After playing around a bit, I *believe* this is the difference:

- **Padding** is the space between the stuff in the element and the borders of the element.
- **Margin** is the space between the element and its container

Selecting elements

Select a tag type with the tag name

```
h1, p {
  ...
}
```

Select only those with a certain *attribute* value

E.g. to change the color of the submit button on a form

```
input[type="submit"] {
  ...
}
```

Useful snippets

Make the list items each *next* to the other in stead of vertical

Good for use in a **navbar**

```
li {
  display: inline;
}
```

The more you know

`block` vs. `inline`

- **Block elements** stretch the *whole width of the page* and have *line breaks before and after*
- **Inline elements** exist within the normal flow of the text they're contained within (no line breaks etc.)

Gotchas

1. `<a>` tags won't inherit properties from their parents
2. `<p>` tags have a non-zero `margin` by default

Linking in Google Web Fonts (nice, free fonts)

1. Go to google.com/fonts
2. Find the font you want
3. Hit the right-arrow-box
4. Go to section (3) of the web-page
5. Copy the `<link href='http://fonts.googleapis.com/css?family=Londrina+Shadow' rel='stylesheet' type='text/css'>` part to the top of your `<head>`
6. Go to section (4) of the web-page
7. Copy the CSS `font-family: 'Londrina Shadow', cursive;` into the CSS block selecting where the font should be applied
8. Change the `http` in the `<link>` to `https`

Making things responsive

1. Link to `<link href="/normalize.css" rel="stylesheet">`
    - This is something you have to download or install or something
    - It does a bunch of stuff you want, but I haven't looked into exactly what

2. Wrap the header of your content in a `<header>` tag
3. Wrap each element of your content in an `<article>` tag
4. `<header>` and `<article>` make the macro-elements of your page CSS-able
    - Alternatively you'd use a `<div>` for this, but using named elements is better for the *semantic-ness* of your HTML

        - In other words you want your intentions to be conveyed in the tag-names you use for SEO and ease-of-CSS-use
5. Use `max-width` with the **media query** demonstrated above instead of `width`

Sources

1. The tutorial at dash.ga.co