COMP 1017

Day 14
Fixed-Width Page Layouts

First, let's start off with a quick review!

the box model

padding

border

margin

Today, we're going to create a shiny new framework.

But, first, let's talk about what we'll

be putting inside of it.

Open Source CSS

Making our lives easier by stealing — er, I mean, *sharing*.

What is reset code?

Reset code is meant to reduce inconsistencies between default styles in different browsers.

This is a fantastic (and simple!) reset code by Eric Meyer.

https://meyerweb.com/eric/tools/css/reset/

We can keep our reset.css as a separate stylesheet.

However, we must link to it before all our other styles.

so it loads first!

Universal Selector

It's nice to be inclusive.

What's a universal selector?

The universal selector (*) selects everything.

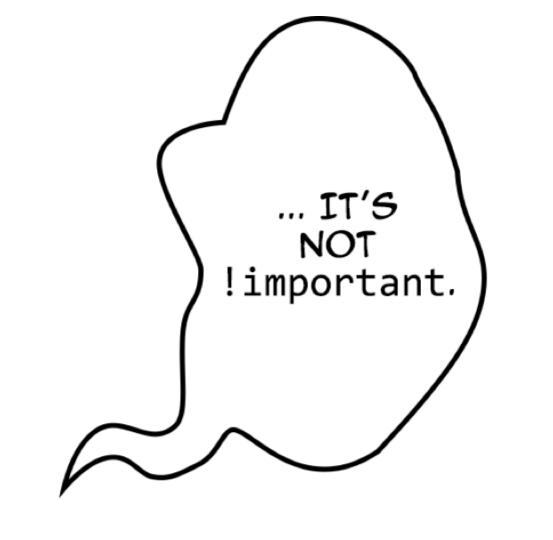
It is very weak* and can easily be overridden.

* its specificity value is calculated as (0) Ziltch, Nada!

battle against each other, the * would be like Hawkeye versus ... uh, literally any other Avenger.

If we were to pit our selectors in a





Fixed Width

Fixin' to do something with my layout.

Finally, in this course, we're going to create fixed width layouts.

add up to a certain width.

This means that everything will

In our case, this width will always be 960 pixels.

great practice on the modern web.

This is easier to code than

responsive websites, but not a

The reason is that it isn't very

mobile friendly.

Either way, the way we'll create our layout is by wrapping our

content with a generic container.

The container will hold the page's

entire contents.

```
.container {
 margin: 0 auto;
 width: 960px;
 font-size: 100%;
 line-height: 1.5;
```

width: 960px;

This defines the width of the

container.

margin: 0 auto;

Takes away the margin (top & bottom)

auto; (left & right) margins are

automatically set by the browser based on

the container, to make the element centered.

font-size: 100%;

Makes your text 100% of the base font size set in the browser. Most browser defaults use 16px base font size.

line-height: 1.5;

Line height sets the difference between two

lines of your content. (leading)

Non-Semantic Wrappers

Everything from here on out is meaningless.

Sometimes, you'll come across a scenario where there isn't really an ideal semantic element to wrap some content.

•

We didn't really need this in

Module 1, but it'll become

important in Module 2.

<div>

<div> is a generic container for
any sort of content that you may
want to group together.

no semantic meaning and doesn't require a heading.

However, unlike <section>, it has

It doesn't really represent or mean anything.

(In fact, we'll only really be using it as a way of targeting specific things with CSS. Think of it as a styling hook)

This makes it a perfect 'wrapper'.

Keep in mind that a <div> will

create its own 'row' by default.

The element works exactly the same as <div>.

 will not create a new row.

In fact, the only difference is that

a heading (inline).

This is great if you want to change,

say, the colour of a single word in

```
<h2>
The <span>Great</span> Escape
</h2>
```

```
h2 span {
    font-family: 'Avenir';
    color: #ed5565;
}
```

The Great Escape