SASS

Syntactically Awesome Stylesheets

What is SASS?

SASS is a CSS preprocessor

Runs on Node.js

Extends the functionality of CSS

Eliminates (or, at least mitigates) many of the traditional CSS pain points

Source files are created by the user (more, later) and 'compiled' by SASS into the output CSS file

Why Use SASS?

It is a pre-processing language which provides indented syntax (its own syntax) (LESS) or CSS-like structure (SCSS) that are processed into the site's CSS

It provides some features, which are used for creating stylesheets that allows writing code more efficiently and is easier to maintain than vanilla CSS

Why Use SASS?

It is a super set of CSS, which means it contains all the features of CSS and is an open source pre-processor, coded in JavaScript

It provides the document style in a good, structured format, better than vanilla CSS

It uses re-usable methods, logic statements and some of the built-in functions such as color manipulation, mathematics and parameter lists

Features of SASS

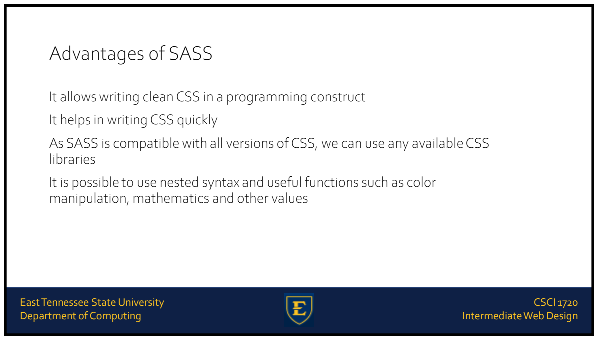
It is more stable, powerful, and compatible with versions of CSS

It is known as syntactic sugar for CSS, which means it makes easier way for user to read or express the things more clearly

It uses its own syntax and compiles to readable CSS

You can easily write CSS in less code within less time

It is an open source [awesome!] pre-processor, which is interpreted into CSS



Advantages of SASS

It allows writing clean CSS in a programming construct

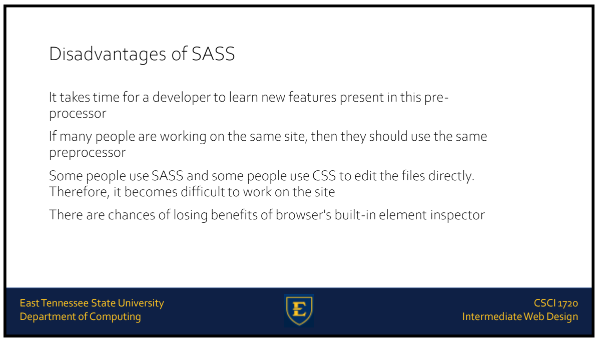
It helps in writing CSS quickly

As SASS is compatible with all versions of CSS, we can use any available CSS libraries

It is possible to use nested syntax and useful functions such as color manipulation, mathematics and other values

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



Disadvantages of SASS

It takes time for a developer to learn new features present in this pre-processor

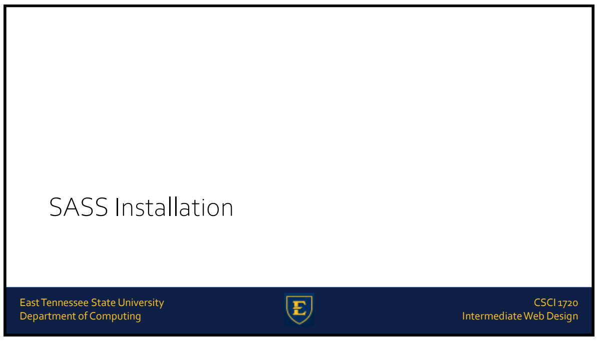
If many people are working on the same site, then they should use the same preprocessor

Some people use SASS and some people use CSS to edit the files directly. Therefore, it becomes difficult to work on the site

There are chances of losing benefits of browser's built-in element inspector

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

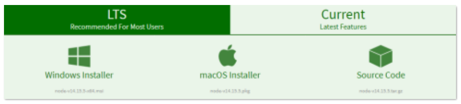


SASS Installation

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

Installing SASS



SASS is now a Nodejs package

If you don't have Nodejs installed on your PC, you'll need to do that first

The download is at https://nodejs.org/en/download/

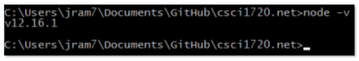
East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

Installing SASS

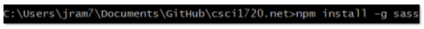
Once Nodejs is installed, open a command prompt

Enter **node -v** to ensure that Nodejs has been successfully installed



```
C:\Users\jram7\Documents\GitHub\csci1720.net>node -v
v12.16.1

C:\Users\jram7\Documents\GitHub\csci1720.net>_
```

Enter **npm install -g sass**



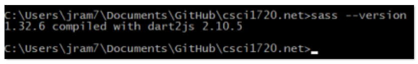
```
C:\Users\jram7\Documents\GitHub\csci1720.net>npm install -g sass
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

Installing SASS

Enter **sass --version** to confirm successful installation



```
C:\Users\jram7\Documents\GitHub\csci1720.net>sass --version
1.32.6 compiled with dart2js 2.10.5

C:\Users\jram7\Documents\GitHub\csci1720.net>_
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

NOTE: The CLI

One thing I've learned over the years is this:

You

CAN NOT!

make it as an IT Pro, if you're not comfortable working from a Command Line Interface. Period.

Installing SASS

It isn't required that you install SASS on your own machine

I've confirmed that it does work on the lab computers, but is a little more convoluted

If you install to your laptop, the **`sass`** command will be installed on the system's PATH and can be run from anywhere

Installing SASS

So, if your project files are on your E:\ drive, for example, let's say

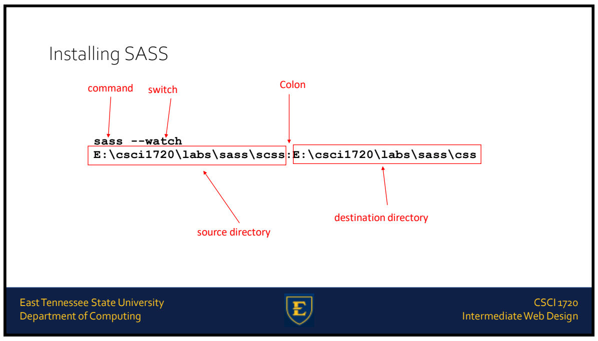
```
E:\csci1720\labs\sass
```

On a lab machine, you'd have to navigate to

```
C:\ruby24-x64\bin
```

and run

```
sass --watch
E:\csci1720\labs\sass\scss:E:\csci1720\labs\sass\css
```

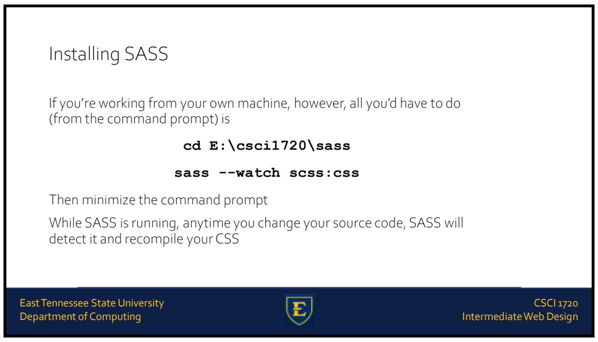


Installing SASS

command switch Colon

```
sass --watch
E:\csci1720\labs\sass\scss:E:\csci1720\labs\sass\css
```

source directory destination directory



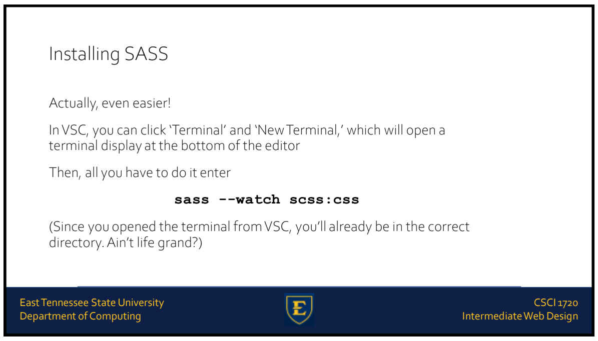
Installing SASS

If you're working from your own machine, however, all you'd have to do (from the command prompt) is

```
cd E:\csci1720\sass
sass --watch scss:css
```

Then minimize the command prompt

While SASS is running, anytime you change your source code, SASS will detect it and recompile your CSS



Installing SASS

Actually, even easier!

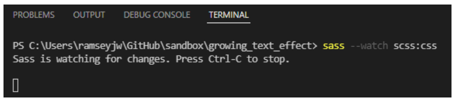
In VSC, you can click 'Terminal' and 'New Terminal,' which will open a terminal display at the bottom of the editor

Then, all you have to do it enter

```
sass --watch scss:css
```

(Since you opened the terminal from VSC, you'll already be in the correct directory. Ain't life grand?)

Installing SASS



```
PS C:\Users\ramseyjw\GitHub\sandbox\growing_text_effect> sass --watch scss:css
Sass is watching for changes. Press Ctrl-C to stop.
```

Installing SASS

IMPORTANT POINT:

In a SASS environment, you never touch the CSS file!

SCSS

Sassy CSS

SASS Source Code

There are two ways (languages, if you will) to create SASS source code

SASS (e.g., main.sass)

- Uses indentation as delimiters
- Instead of braces {}
- I don't like it

SCSS (e.g., main.scss)

- Superset of CSS
- Uses CSS syntax we're familiar with

SASS Source Code

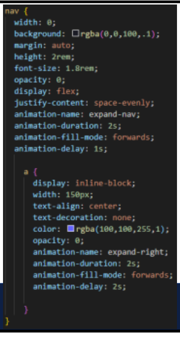
There are two ways (languages, if you will) to create SASS source code

SCSS seems to be more popular

Makes sense, since any valid CSS is also valid SCSS

Syntax is much the same

SASS Source Code

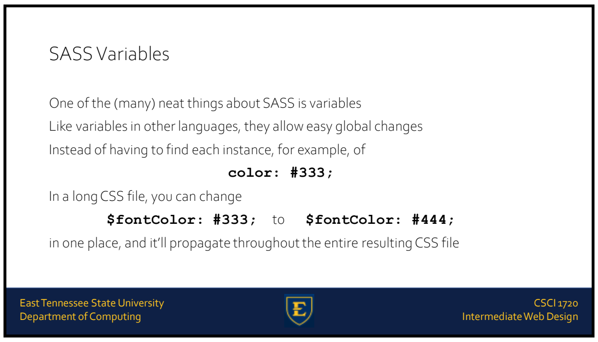


```
nav {
  width: 0;
  background: ▢rgba(0,0,100,.1);
  margin: auto;
  height: 2rem;
  font-size: 1.8rem;
  opacity: 0;
  display: flex;
  justify-content: space-evenly;
  animation-name: expand-nav;
  animation-duration: 2s;
  animation-fill-mode: forwards;
  animation-delay: 1s;

    a {
      display: inline-block;
      width: 150px;
      text-align: center;
      text-decoration: none;
      color: 🟦rgba(100,100,255,1);
      opacity: 0;
      animation-name: expand-right;
      animation-duration: 2s;
      animation-fill-mode: forwards;
      animation-delay: 2s;

    }
}
```

SASS Variables

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Variables

One of the (many) neat things about SASS is variables
Like variables in other languages, they allow easy global changes
Instead of having to find each instance, for example, of

```
color: #333;
```

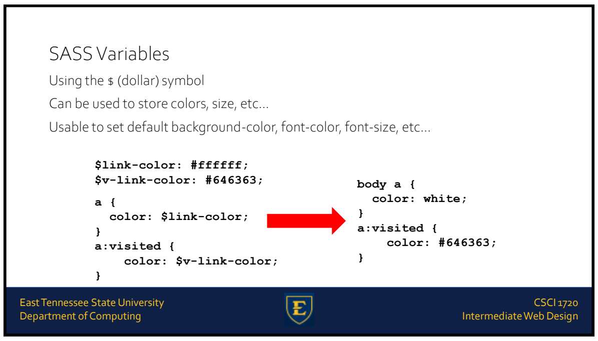
In a long CSS file, you can change

`$fontColor: #333;` to `$fontColor: #444;`

in one place, and it'll propagate throughout the entire resulting CSS file

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Variables

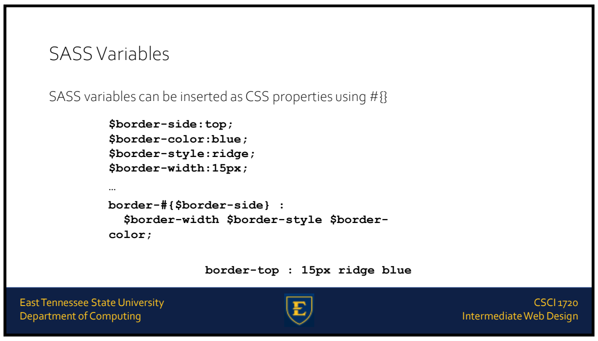
Using the $ (dollar) symbol
Can be used to store colors, size, etc…
Usable to set default background-color, font-color, font-size, etc…

```
$link-color: #ffffff;
$v-link-color: #646363;

a {
  color: $link-color;
}
a:visited {
    color: $v-link-color;
}
```

```
body a {
  color: white;
}
a:visited {
    color: #646363;
}
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Variables

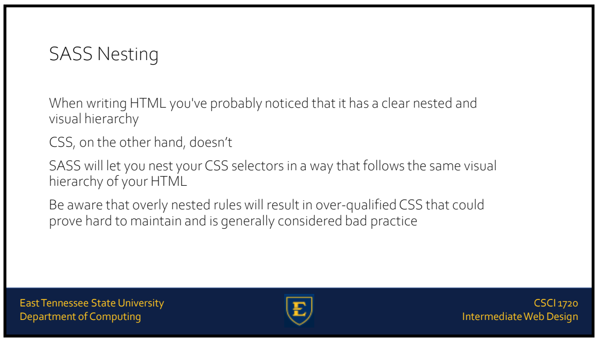
SASS variables can be inserted as CSS properties using #{}

```
$border-side:top;
$border-color:blue;
$border-style:ridge;
$border-width:15px;
…
border-#{$border-side} :
  $border-width $border-style $border-
color;
```

```
border-top : 15px ridge blue
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Nesting

When writing HTML you've probably noticed that it has a clear nested and visual hierarchy

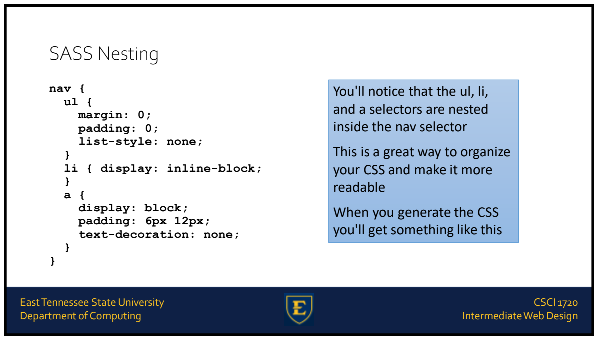
CSS, on the other hand, doesn't

SASS will let you nest your CSS selectors in a way that follows the same visual hierarchy of your HTML

Be aware that overly nested rules will result in over-qualified CSS that could prove hard to maintain and is generally considered bad practice

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Nesting

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li { display: inline-block;
  }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

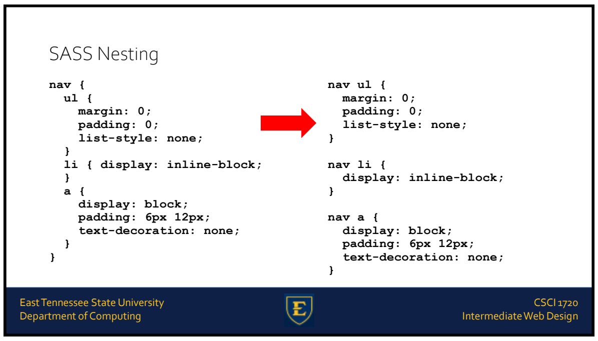
You'll notice that the ul, li, and a selectors are nested inside the nav selector

This is a great way to organize your CSS and make it more readable

When you generate the CSS you'll get something like this

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



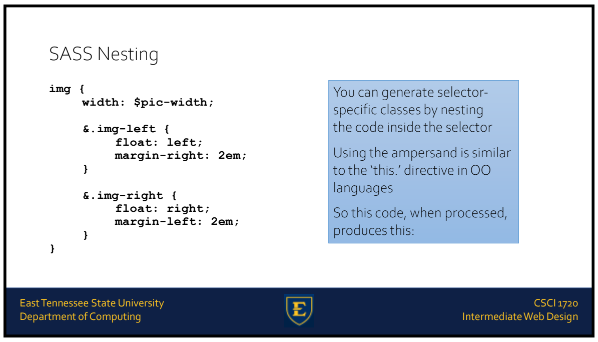
SASS Nesting

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li { display: inline-block;
  }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}

nav li {
  display: inline-block;
}

nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Nesting

```
img {
    width: $pic-width;

    &.img-left {
         float: left;
         margin-right: 2em;
    }

    &.img-right {
         float: right;
         margin-left: 2em;
    }
}
```

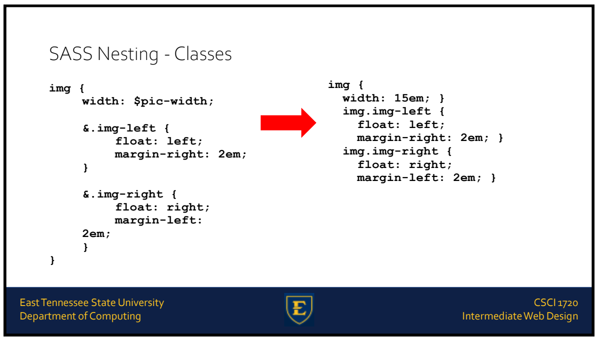
You can generate selector-specific classes by nesting the code inside the selector

Using the ampersand is similar to the 'this.' directive in OO languages

So this code, when processed, produces this:

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Nesting - Classes

```
img {
    width: $pic-width;

    &.img-left {
         float: left;
         margin-right: 2em;
    }

    &.img-right {
         float: right;
         margin-left:
    2em;
    }
}
```

```
img {
  width: 15em; }
  img.img-left {
    float: left;
    margin-right: 2em; }
  img.img-right {
    float: right;
    margin-left: 2em; }
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

A note on output styles

When developing with Sass sometimes there is a need to adjust the output style of the CSS. Sass's default CSS style is good but might not be applicable for all situations.

Sass supports four different output styles

:~~nested~~

:~~compact~~

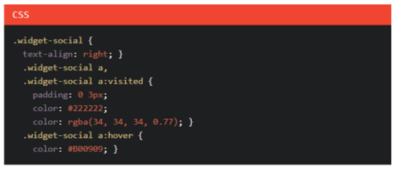
:expanded

:compressed

To understand what each of the above settings output consider the following snippet of code:

A note on output styles

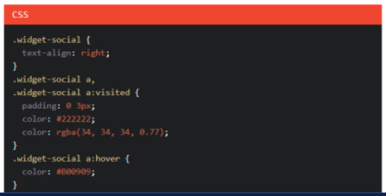
:nested



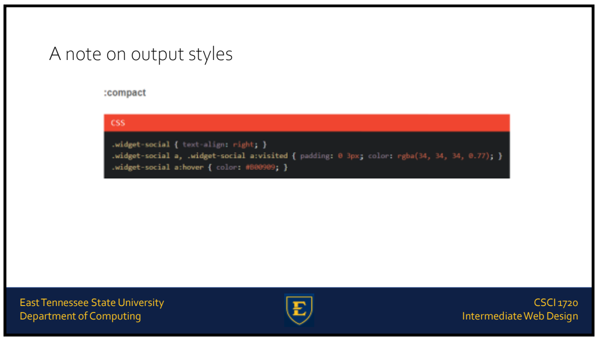
```css
.widget-social {
  text-align: right; }
  .widget-social a,
  .widget-social a:visited {
    padding: 0 3px;
    color: #222222;
    color: rgba(34, 34, 34, 0.77); }
  .widget-social a:hover {
    color: #B00909; }
```

A note on output styles

:expanded



```css
.widget-social {
  text-align: right;
}
.widget-social a,
.widget-social a:visited {
  padding: 0 3px;
  color: #222222;
  color: rgba(34, 34, 34, 0.77);
}
.widget-social a:hover {
  color: #B00909;
}
```



A note on output styles

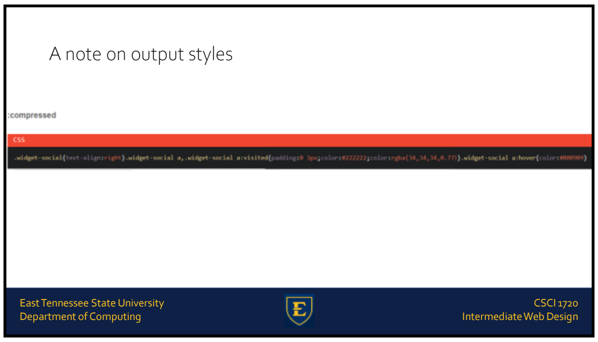
:compact

CSS

```
.widget-social { text-align: right; }
.widget-social a, .widget-social a:visited { padding: 0 3px; color: rgba(34, 34, 34, 0.77); }
.widget-social a:hover { color: #B00909; }
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



A note on output styles

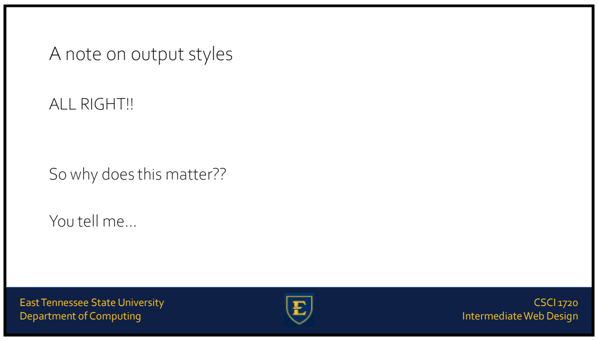
:compressed

CSS

```
.widget-social{text-align:right}.widget-social a,.widget-social a:visited{padding:0 3px;color:#222222;color:rgba(34,34,34,0.77)}.widget-social a:hover{color:#B00909}
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



A note on output styles

ALL RIGHT!!

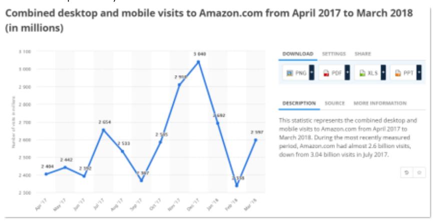
So why does this matter??

You tell me…

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

A note on output styles



Combined desktop and mobile visits to Amazon.com from April 2017 to March 2018 (in millions)

This statistic represents the combined desktop and mobile visits to Amazon.com from April 2017 to March 2018. During the most recently measured period, Amazon.com had almost 2.6 billion visits, down from 3.04 billion visits in July 2017.

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

A note on output styles

I'm not good with math, but if I've figured it right, this boils down to 3,009,259 hits A SECOND!

Three … freakin … MILLION hits … a … SECOND!!!

So, as a web developer, what does that mean to you?

Here's a hint: Whitespace

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

Output:

This comes down to what you are using for compiling. If you're running with the command-line you can just pass in the setting using the --style flag like so:

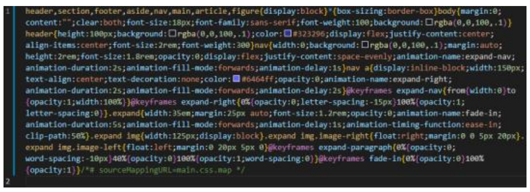
```
sass --watch style.scss:style.css --style compressed
```

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

Output:

`sass --watch style.scss:style.css --style compressed`



```
header,section,footer,aside,nav,main,article,figure{display:block}*{box-sizing:border-box}body{margin:0;
content:"";clear:both;font-size:18px;font-family:sans-serif;font-weight:100;background:rgba(0,0,100,.1)}
header{height:100px;background:rgba(0,0,100,.1);color:#323296;display:flex;justify-content:center;
align-items:center;font-size:2rem;font-weight:300}nav{width:0;background:rgba(0,0,100,.1);margin:auto;
height:2rem;font-size:1.8rem;opacity:0;display:flex;justify-content:space-evenly;animation-name:expand-nav;
animation-duration:2s;animation-fill-mode:forwards;animation-delay:1s}nav a{display:inline-block;width:150px;
text-align:center;text-decoration:none;color:#6464ff;opacity:0;animation-name:expand-right;
animation-duration:2s;animation-fill-mode:forwards;animation-delay:2s}@keyframes expand-nav{from{width:0}to
{opacity:1;width:100%}}@keyframes expand-right{0%{opacity:0;letter-spacing:-15px}100%{opacity:1;
letter-spacing:0}}.expand{width:35em;margin:25px auto;font-size:1.2rem;opacity:0;animation-name:fade-in;
animation-duration:5s;animation-fill-mode:forwards;animation-delay:1s;animation-timing-function:ease-in;
clip-path:50%}.expand img{width:125px;display:block}.expand img.image-right{float:right;margin:0 0 5px 20px}.
expand img.image-left{float:left;margin:0 20px 5px 0}@keyframes expand-paragraph{0%{opacity:0;
word-spacing:-10px}40%{opacity:0}100%{opacity:1;word-spacing:0}}@keyframes fade-in{0%{opacity:0}100%
{opacity:1}}/*# sourceMappingURL=main.css.map */
```

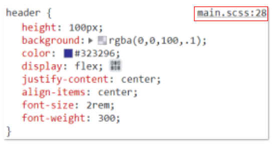
Output:

`sass --watch style.scss:style.css --style compressed`

What does this mean?? `/*# sourceMappingURL=main.css.map */`

This is a file (we never need to do anything with it) that maps the CSS back to the SCSS code that generated it

Browsers' Developer Tools can then use this mapping to help with troubleshooting



```
header {                                   main.scss:28
    height: 100px;
    background: ▸ rgba(0,0,100,.1);
    color: #323296;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 2rem;
    font-weight: 300;
}
```

SASS Partials

Break….take a breath…

SASS Partials

You can create partial SASS files that contain little snippets of CSS that you can include in other SASS files

This is a great way to modularize your CSS and help keep things easier to maintain

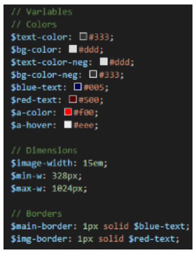
A partial is simply a SASS file named with a leading underscore. You might name it something like _variables.scss

The underscore lets SASS know that the file is only a partial file and that it should not be generated into a CSS file

SASS partials are used with the @import directive

SASS Partials

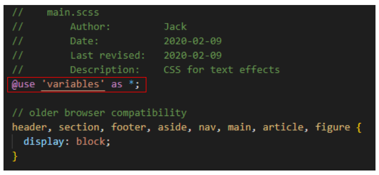
File name: _variables.scss



```
// Variables
// Colors
$text-color: #333;
$bg-color: #ddd;
$text-color-neg: #ddd;
$bg-color-neg: #333;
$blue-text: #005;
$red-text: #500;
$a-color: #f00;
$a-hover: #eee;

// Dimensions
$image-width: 15em;
$min-w: 328px;
$max-w: 1024px;

// Borders
$main-border: 1px solid $blue-text;
$img-border: 1px solid $red-text;
```

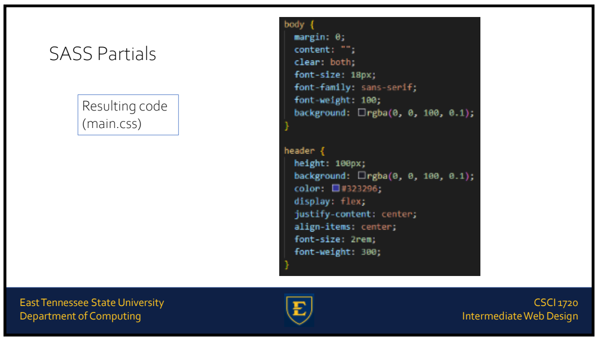
SASS Partials

Resulting code



```
//    main.scss
//        Author:         Jack
//        Date:           2020-02-09
//        Last revised:   2020-02-09
//        Description:    CSS for text effects
@use 'variables' as *;

// older browser compatibility
header, section, footer, aside, nav, main, article, figure {
  display: block;
}
```

We used to use **`@import`** instead of **`@use`**, but that's now deprecated



SASS Partials

Resulting code (main.css)

```
body {
  margin: 0;
  content: "";
  clear: both;
  font-size: 18px;
  font-family: sans-serif;
  font-weight: 100;
  background: rgba(0, 0, 100, 0.1);
}

header {
  height: 100px;
  background: rgba(0, 0, 100, 0.1);
  color: #323296;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 2rem;
  font-weight: 300;
}
```

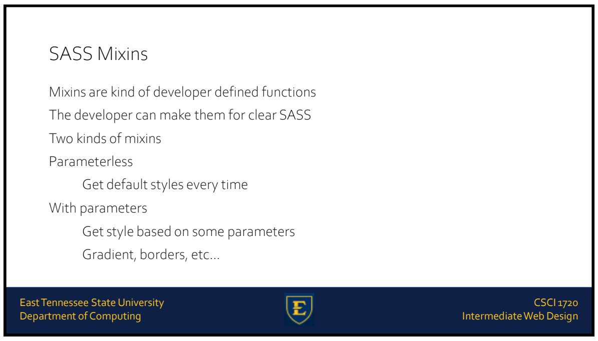
East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

SASS Mixins

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Mixins

Mixins are kind of developer defined functions
The developer can make them for clear SASS
Two kinds of mixins
Parameterless
    Get default styles every time
With parameters
    Get style based on some parameters
    Gradient, borders, etc...

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design

SASS

// Use **`@mixin mixin-name`**
// Then the styles are normal SASS

```
@mixin clearfix{
    zoom:1;
    content:"";
    height:0;
    clear:both;
}
```

```
ul#main-nav{
  @include clearfix;
  …
}
```

SASS Extend/Inheritance

SASS Extend/Inheritance

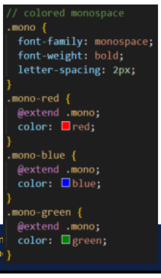
This is one of the most useful features of SASS

Using **`@extend`** lets you share a set of CSS properties from one selector to another

It helps keep your SASS very DRY ("Don't Repeat Yourself")

In our example we're going to create a simple series of messaging for errors, warnings and successes

SASS Extend/Inheritance



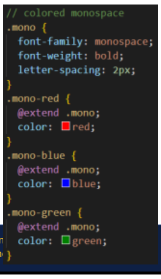
```
// colored monospace
.mono {
  font-family: monospace;
  font-weight: bold;
  letter-spacing: 2px;
}
.mono-red {
  @extend .mono;
  color: red;
}
.mono-blue {
  @extend .mono;
  color: blue;
}
.mono-green {
  @extend .mono;
  color: green;
}
```

What the above code does is allow you to take the CSS properties in .message and apply them to .success

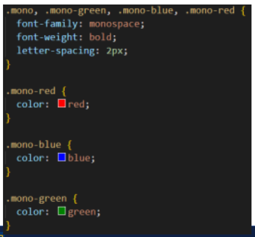
The magic happens with the generated CSS, and this helps you avoid having to write multiple class names on HTML elements

This is what it looks like:

SASS Extend/Inheritance

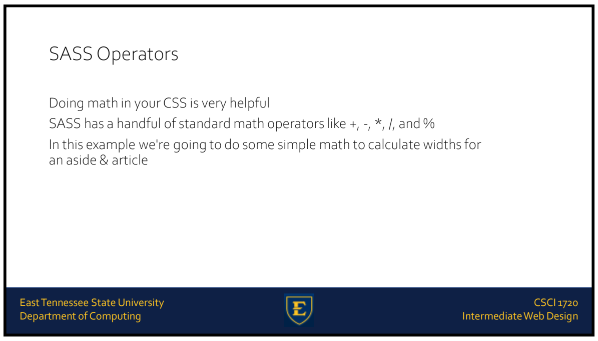


```
// colored monospace
.mono {
  font-family: monospace;
  font-weight: bold;
  letter-spacing: 2px;
}
.mono-red {
  @extend .mono;
  color: red;
}
.mono-blue {
  @extend .mono;
  color: blue;
}
.mono-green {
  @extend .mono;
  color: green;
}
```



```
.mono, .mono-green, .mono-blue, .mono-red {
  font-family: monospace;
  font-weight: bold;
  letter-spacing: 2px;
}

.mono-red {
  color: red;
}

.mono-blue {
  color: blue;
}

.mono-green {
  color: green;
}
```

SASS Operators



SASS Operators

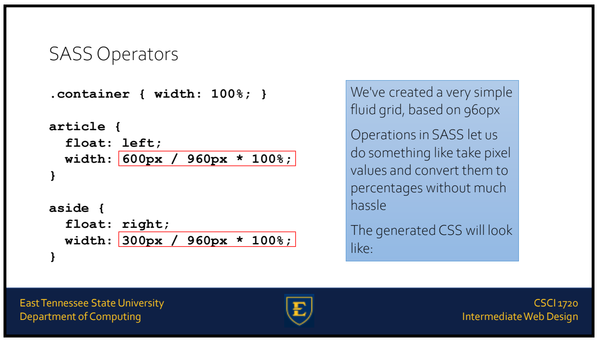
Doing math in your CSS is very helpful

SASS has a handful of standard math operators like +, -, *, /, and %

In this example we're going to do some simple math to calculate widths for an aside & article

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Operators

```
.container { width: 100%; }

article {
  float: left;
  width: 600px / 960px * 100%;
}

aside {
  float: right;
  width: 300px / 960px * 100%;
}
```

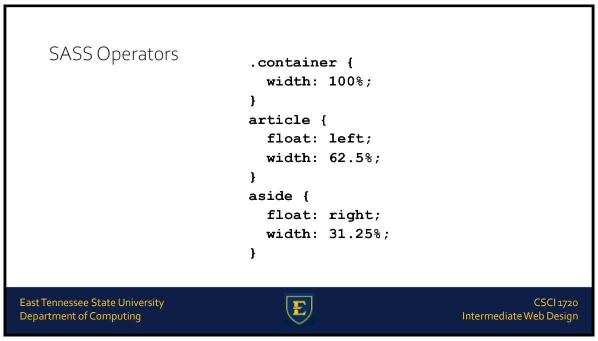
We've created a very simple fluid grid, based on 960px

Operations in SASS let us do something like take pixel values and convert them to percentages without much hassle

The generated CSS will look like:

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



SASS Operators

```
.container {
  width: 100%;
}
article {
  float: left;
  width: 62.5%;
}
aside {
  float: right;
  width: 31.25%;
}
```

East Tennessee State University
Department of Computing

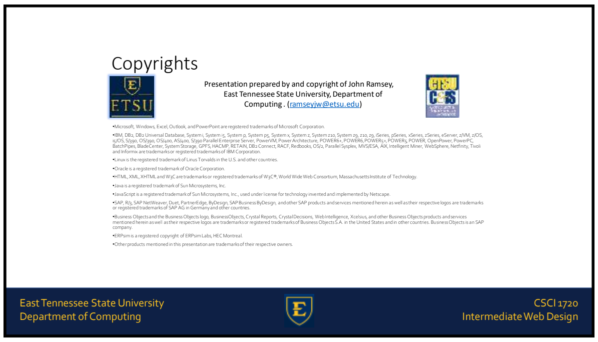
CSCI 1720
Intermediate Web Design

Lab this week

We'll create a page using Sass for styling

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design



Copyrights

East Tennessee State University
Department of Computing

CSCI 1720
Intermediate Web Design