GIT & SASS

WITH GUY ROUTLEDGE

@GUYROUTLEDGE | #FEWD

AGENDA

- Version Control
- Sass

OBJECTIVE

By the end of the session you'll be able to:

- Keep your project backed up and well organised
- Add superpowers to your CSS

VERSION CONTROL

Have you ever lost or overwritten some work that took you ages?

VERSION CONTROL

Version control is a system that enables you to take regular "snapshots" of your work in progress called "commits". This enables:

- Recording the history of a project
- Constant backups of your work
- A more organised development workflow
- A safer way to collaborate with others
- Tools for publishing your work online

VERSION CONTROL

Keeping your code organised helps you be more efficient.

Keeping your work backed up helps you stay sane.

Going back over the history of a project helps you learn.

COMMITS

Think of each commit as if you copied your whole project folder and saved it somewhere safe.

This folder is labeled with a unique ID and recorded in a chronological timeline.

If you want to view what a file looked like at any point in history you can open up the folder with a certain commit ID and take a look at the files inside.

COMMITS

Instead of making a complete copy of your project for every snapshot, Version control systems can just save the differences between files.

This keeps the history small in file size and quick to work with.

VERSION CONTROL

The most popular version control system in use today is called `git`.

Projects are stored in something called a `repository` which is just like a collection of files and folders.

You have a `local` repo on your laptop and a `remote` repo online and the two are connected.

VERSION CONTROL

`git` is a command line tool but there are lots of visual applications that make the process easier to use.

VERSION CONTROL PROCESS

A typical `git` workflow goes like this:

- Create a project folder
- Initialize it as a git project
- Write some code
- Check the status of what files/folders have changed
- Mark one or more files to be added into the next snapshot
- Take a snapshot and write a message describing what you did and why
- Write some more code
- Create the next snapshot
- Rinse and repeat often
- Push your changes to an online repository

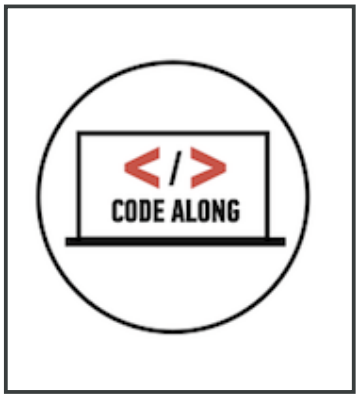
VERSION CONTROL PROCESS

Commit often even as you experiment.

Push regularly once everything is in good shape.

COMMIT MESSAGES

Think of each commit message like an email. You have a brief subject and (an optional) longer description to describe what you did and *why*



GIT FOR FINAL PROJECTS

GITHUB DESKTOP WORKFLOW

- Create a new project via the `+` button
- Select the `create` tab
- Add a name for your project folder
- Choose where to save it
- Write some code
- Switch back to Github and check the `changes` tab
- Select the files to be committed
- Write a commit message
- Click `commit to master`
- Write some more code, commit again
- Check the history of changes
- Publish your repo to Github
- Code, commit and sync work regularly

SASS

SASS

Sass is a *pre-processor* that gives CSS super powers.

It's very similar to CSS but has lots of extra features that make writing your styling code faster, more efficient, more maintainable and more organised.

Sass stands for "Syntastically Awesomes Style Sheets" and was invented by Hampton Catlin in 2006.

SASS

I (almost) never write normal CSS any more. I write Sass.

Sass syntax and CSS syntax are almost idential though, so learning CSS first is no bad thing. The identical syntax also makes it easy to switch to Sass from CSS.

SASS

Sass provides lots of features that aren't possible or don't exist in normal CSS.

SASS

Sass can do this because the code you write is run through a `compiler` before it reaches the browser.

The end result is just normal CSS

SASS FILES

We write Sass in a file with a `.scss` file extension.

Any file with a `.scss` extension will be compiled into a corresponding `.css` file of the same name.

SASS FILES

Files that start with an `_` underscore character aren't compiled but can be `@import`ed into a Sass file that will be compiled.

These files are called Sass partials and might look like `_nav.scss` or `_about-page.scss`.

This helps with code organisation and instead of having one massive CSS file, we can have lots of smaller Sass files that get combined into one by the compiler.

SASS

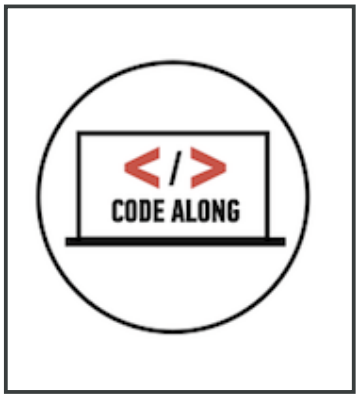
The Sass compiler enables us to do more powerful things like:

- Combining and compressing files
- Nested selectors and media queries
- Variables
- Maths
- Colour manipulation
- Functions
- Mixins
- Loops
- Conditional Statements
- And much more!

SASS SYNTAX

Sass looks a bit like this:

```
.site-header {
    .logo {
        float:left;
    }
    .nav {
        @include horizontal-nav;
        color:$color-brand;
    }
}
.site-content {
    // more styles
}
```



SASSY CSS

http://codepen.io/guyroutledge/pen/ORvqaL

COMPILING SASS

COMPILING SASS

We can use a tool like Codepen to compile our styles when experimenting.

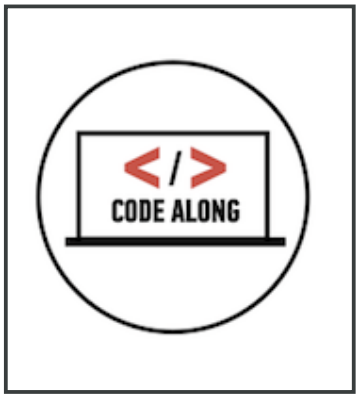
But if we want to use Sass in a project we're building locally, we need an application that can compile Sass for us.

COMPILING SASS

Sass is often compiled via the command line but this is quite advanced for beginners.

Here are some free apps you can use instead:

- Scout
- Koala
- Compass



COMPILING SASS WITH KOALA