1. Intro

Make commits of our very own.

Git init

Git clone

Git log

Git status

Have download sheet with all terms ready.

Git add = add files from the working directory to the staging index

Git commit = Take files from the staging index and save them to the repository

Git diff = Really cool, Displays the difference between two versions of a file.

1. Git Add

We haven't done anything specific with Git just yet, but it's watching this directory (since it's a Git project), and it knows that we've created a couple of new files. What's also pretty neat about the output of the git status command is that it's telling us that the files are untracked by Git.

git add index.html

git add css/app.css js/app.js

git add . 🡨 stages the remaining files so you don’t have to type everything

1. Git Commit

Atom Editor Setup

git config --global core.editor "atom --wait"

Sublime Text Setup

git config --global core.editor "'C:/Program Files/Sublime Text 2/sublime\_text.exe' -n -w"

VSCode Setup

git config --global core.editor "code --wait"

git commit -m "Initial commit"

You need to use git add before you can commit any files, otherwise git status will display.

1. Commit Messages

Do

* do keep the message short (less than 60-ish characters)
* do explain what the commit does (not how or why!)

Do not

* do not explain why the changes are made (more on this below)
* do not explain how the changes are made (that's what git log -p is for!)
* do not use the word "and"
* if you have to use "and", your commit message is probably doing too many changes - break the changes into separate commits
* e.g. "make the background color pink and increase the size of the sidebar"

Above all, be consistent in how you write your commit messages!

Make sure to explain what was changed. Don't just state that changes were made or where they were made.

<https://udacity.github.io/git-styleguide/>

1. Git Diff

The git diff command can be used to see changes that have been made but haven't been committed, yet.

1. Having Git Ignore Files

If you want to keep a file in your project's directory structure but make sure it isn't accidentally committed to the project, you can use the specially named file, .gitignore (note the dot at the front, it's important!). Add this file to your project in the same directory that the hidden .git directory is located. All you have to do is list the names of files that you want Git to ignore (not track) and it will ignore them.

Globbing lets you use special characters to match patterns/characters. In the .gitignore file, you can use the following:

blank lines can be used for spacing

# - marks line as a comment

\* - matches 0 or more characters

? - matches 1 character

[abc] - matches a, b, \_or\_ c

\*\* - matches nested directories - a/\*\*/z matches

a/z

a/b/z

a/b/c/z

1. Outro

Git add to stage, commit to actually commit changes, good commit messages, helpful git diff, and the .gitignore file.