LaunchSchool
An Online School for Developers

1.
Create a git repository on your machine (aka, your "local")

— — — — —

git init
· you can turn any directory on your machine into a git repository by issuing "git init". Make sure not to nest git repositories. You are allowed to create other folders within a git repository, just not another git repository.

or

git clone <remote git url>
· you can download an existing project with "git clone", and this will also be a git project.

This directory is now your project directory and also the local git repository for your project. You can create new files and folders in this git repository, but don't create another git repository under it.

You only need to do this one time to set up this git repository.

2.
Stage Changes

— — — — —

git add <path/to/file>
· now, from your git repository (also your project directory), you can "stage changes" to ready them for commit. These are changes you make to any file or folder within the git repository, including, modifying a file contents, adding new files/folders, deleting files/folders.

There are some shortcuts you can use, such as "git add ." or "git add -u", but in the beginning, practice by "git add" every file you're staging.

You can use another helpful git command, **git status**, to see all modified but unstaged files, and also all staged files. You should get familiar with "git status" and use it extensively.

3.
Package Staged Changes Into A Commit

— — — — —

git commit -m "a commit message"
· this will package all the "staged" files from step #2 into a commit, along with a message.

In git, commits are what you push and pull around to/from other repositories. Commits will not show up with "git status". To see a list of commits in the repository, you can use **git log**.

4.
Push Commits To Remote Repository

— — — — —

git push origin master
· this will push all commits that's not yet on "origin" to "origin".

"origin" in the above code example is the alias to a remote git repository, usually an associated repository hosted on Github.

Where did "orgin" come from? You have to first tell your local git repository that from now on, we're going to alias "origin" to the url of a remote git repository. So in order to push/pull to/from "origin", you had to first have issued **git remote add origin <url of remote git repository>**.