Before you can make commits or do anything else with a git repository, the repository needs to actually exist. To create a new repository with Git, we'll use the

The

***Required Commands***

*Heads up! We'll be using the following terminal commands in this lesson:*

*If you're not sure how to use them, check out our course*[*Shell Workshop*](https://www.udacity.com/course/shell-workshop--ud206)*!*

*We'll also be using the idea of the current working directory, the directory that your shell is "looking at" right now. Using*

**Create Course Directories**

We're about to create a new project for this course. Since we're all professionals, we want our projects organized. If you already have a location on your computer where you put all your projects, then keep doing what you're doing. I'll be storing all of my work in a directory called

If you want to follow along with me:

* create a directory called
* inside that, create another directory called
* use the

If you're a copy/paster like me, just run this command on the terminal -

If you're all set, then your terminal should be "inside" the

*[The Terminal application located in the new-git-project directory.](https://classroom.udacity.com/nanodegrees/nd104-ent/parts/e325d5cc-ccd1-4ca8-a562-7eed09ab76c7/modules/a141ebc6-acee-494b-927f-2016d6fe0f9c/lessons/59288694-18d7-477e-948a-014fd2354293/concepts/fa8f761a-d0a2-4be1-a5b9-60116ea4ecd1#)*

**Git Init**

Fantastic work - we're all set up and ready to start using the

This is one of the easiest commands to run. All you have to do is run

*[The terminal application showing the](https://classroom.udacity.com/nanodegrees/nd104-ent/parts/e325d5cc-ccd1-4ca8-a562-7eed09ab76c7/modules/a141ebc6-acee-494b-927f-2016d6fe0f9c/lessons/59288694-18d7-477e-948a-014fd2354293/concepts/fa8f761a-d0a2-4be1-a5b9-60116ea4ecd1#)*

**QUIZ QUESTION**

Did you run

After running

* Yes - I now see the word "master".  No, it looks exactly the same.

**SUBMIT**

**Git Init's Effect**

Running the

Let's take a brief look at the contents of the

*WARNING: Don't directly edit any files inside the*

*[Mac's Finder window showing the contents of the hidden ".git" directory.](https://classroom.udacity.com/nanodegrees/nd104-ent/parts/e325d5cc-ccd1-4ca8-a562-7eed09ab76c7/modules/a141ebc6-acee-494b-927f-2016d6fe0f9c/lessons/59288694-18d7-477e-948a-014fd2354293/concepts/fa8f761a-d0a2-4be1-a5b9-60116ea4ecd1#)*

**.Git Directory Contents**

*We're about to take a look at the*

Here's a brief synopsis on each of the items in the

* **config file** - where all *project specific* configuration settings are stored. From the [Git Book](https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration):  *Git looks for configuration values in the configuration file in the Git directory (.git/config) of whatever repository you’re currently using. These values are specific to that single repository.*For example, let's say you set that the global configuration for Git uses your personal email address. If you want your work email to be used for a specific project rather than your personal email, that change would be added to this file.
* **description file** - this file is only used by the GitWeb program, so we can ignore it
* **hooks directory** - this is where we could place client-side or server-side scripts that we can use to hook into Git's different lifecycle events
* **info directory** - contains the global excludes file
* **objects directory** - this directory will store all of the commits we make
* **refs directory** - this directory holds pointers to commits (basically the "branches" and "tags")

Remember, other than the "hooks" directory, you shouldn't mess with pretty much any of the content in here. The "hooks" directory *can* be used to hook into different parts or events of Git's workflow, but that's a more advanced topic that we won't be getting into in this course.

**Further Research**

* [Git Internals - Plumbing and Porcelain](https://git-scm.com/book/en/v2/Git-Internals-Plumbing-and-Porcelain) (advanced - bookmark this and check it out later)
* [Customizing Git - Git Hooks](https://git-scm.com/book/en/v2/Customizing-Git-Git-Hooks)

**Git Init Recap**

Use the

Running this command creates a hidden

**Helpful Links**

* [Initializing a Repository in an Existing Directory](https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository#Initializing-a-Repository-in-an-Existing-Directory)
* [git init docs](https://git-scm.com/docs/git-init)
* [git init Tutorial](https://www.atlassian.com/git/tutorials/setting-up-a-repository)