**Why Clone?**

First, what is cloning?

*to make an identical copy*

What's the value of creating an identical copy of something, and how does this relate to Git and version control?

Why would you want to create an identical copy? Well, when I work on a new web project, I do the same set of steps:

* create an
* create a
* create a
* create an
* create
* create
* add starter HTML code in
* add configuration files for linting (validating code syntax)
  + [HTML linting](http://htmlhint.com/)
  + [CSS linting](https://stylelint.io/)
  + [JavaScript linting](http://eslint.org/)
* [configure my code editor](http://editorconfig.org/)

...and I do this *every time* I create a new project!...which is a lot of effort I'm putting in for each new project. I didn't want to keep doing these same steps over and over, so I did all of the steps listed above one last time and created a starter project for myself. Now when I create a new project, I just make an identical copy of that starter project!

The way that cloning relates to Git is that the command we'll be running on the terminal is

Wanna try cloning an existing project? Let's see how Git's

***Verify Terminal Location***

*TIP: Now before you clone anything, make sure you are located in the correct directory on the command line. Cloning a project creates a new directory and places the cloned Git repository in it. The problem is that you can't have nested Git repositories. So make sure the terminal's current working directory isn't located in a Git repository. If your current working directory is not in your shell's prompt, type*

**Cloning The Blog Repository**

Ready? Let's get cloning!

The command is

*[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/59082be0-c1af-4839-8ae5-0a182dc5bfe8#)*

**Git Clone Output Explanation**

Let's look briefly at the output that

The first line says "Cloning into 'course-git-blog-project'...". Git is creating a directory (with the same name of the project we're cloning) and putting the repository in it...that's pretty cool!

The rest of the output is basically validation - it's counting the remote repository's number of objects, then it compresses and receives them, then it unpacks them.

**Clone Project And Use Different Name**

You just cloned the blog project for this course. Awesome job!

The command you ran in the terminal was:

...which created a directory named

What if you want to use a different name instead of the default one? Yes, you could just run the command above and manually rename it in Finder/Windows Explorer or use

**QUIZ QUESTION**

Why don't you check out [the documentation for](https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository#Cloning-an-Existing-Repository) and pick the correct way to do it from the options below. The command should clone the blog project repo and store it in a directory named

* git clone-new-dir <https://github.com/udacity/course-git-blog-project> blog-project     git clone <https://github.com/udacity/course-git-blog-project> --out blog-project     git clone <https://github.com/udacity/course-git-blog-project> --rename blog-project    git clone <https://github.com/udacity/course-git-blog-project> blog-project

**SUBMIT**

***Not In A Git Repository?***

*WARNING: Here's a very important step that often gets missed when you first start working with Git. When the*

*Remember to use the Terminal's command prompt as an aid - if you're in a directory that is a Git repository, the command prompt will include a name in parentheses.*

*[Mac's Terminal application. The terminal shows the starting 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/59082be0-c1af-4839-8ae5-0a182dc5bfe8#)*

*[Mac's Terminal application. The terminal uses 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/59082be0-c1af-4839-8ae5-0a182dc5bfe8#)*

**Look At The Project**

So you've cloned the project to your computer, and you've

Open up the

*[The blog project loaded in Chrome.](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/59082be0-c1af-4839-8ae5-0a182dc5bfe8#)*

**Git Clone Recap**

The

This command:

* takes the path to an existing repository
* by default will create a directory with the same name as the repository that's being cloned
* can be given a second argument that will be used as the name of the directory
* will create the new repository inside of the current working directory

**Helpful Links**

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

**Status Update**

At this point, we have two Git repositories:

* the empty one that we created with the
* the one we cloned with the

How can we find any information about these repositories? Git's controlling them, but how can we find out what Git knows about our repos? To figure out what's going on with a repository, we use the

Supporting Materials

[**Git Repository .zip file**](https://video.udacity-data.com/topher/2020/March/5e714611_course-git-blog-project-master/course-git-blog-project-master.zip)