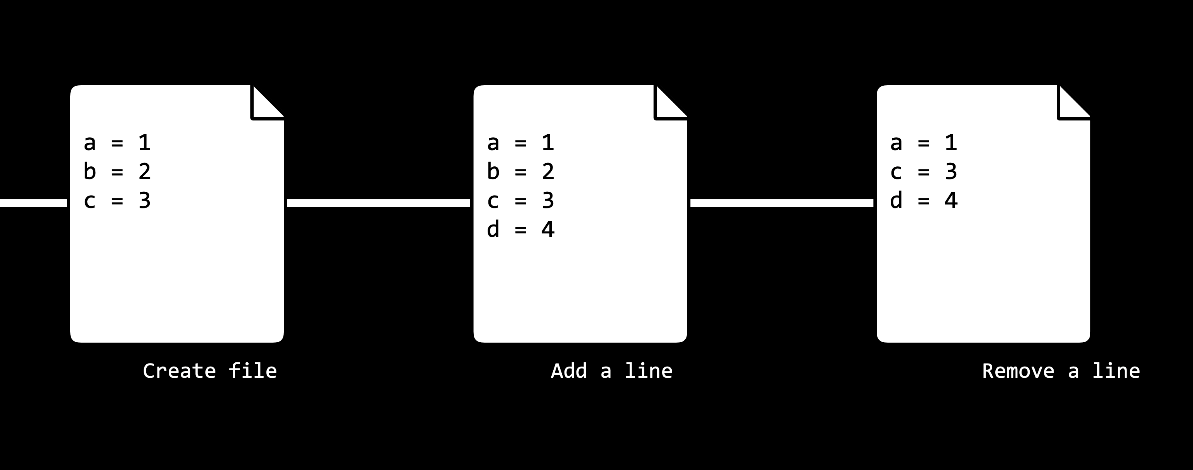
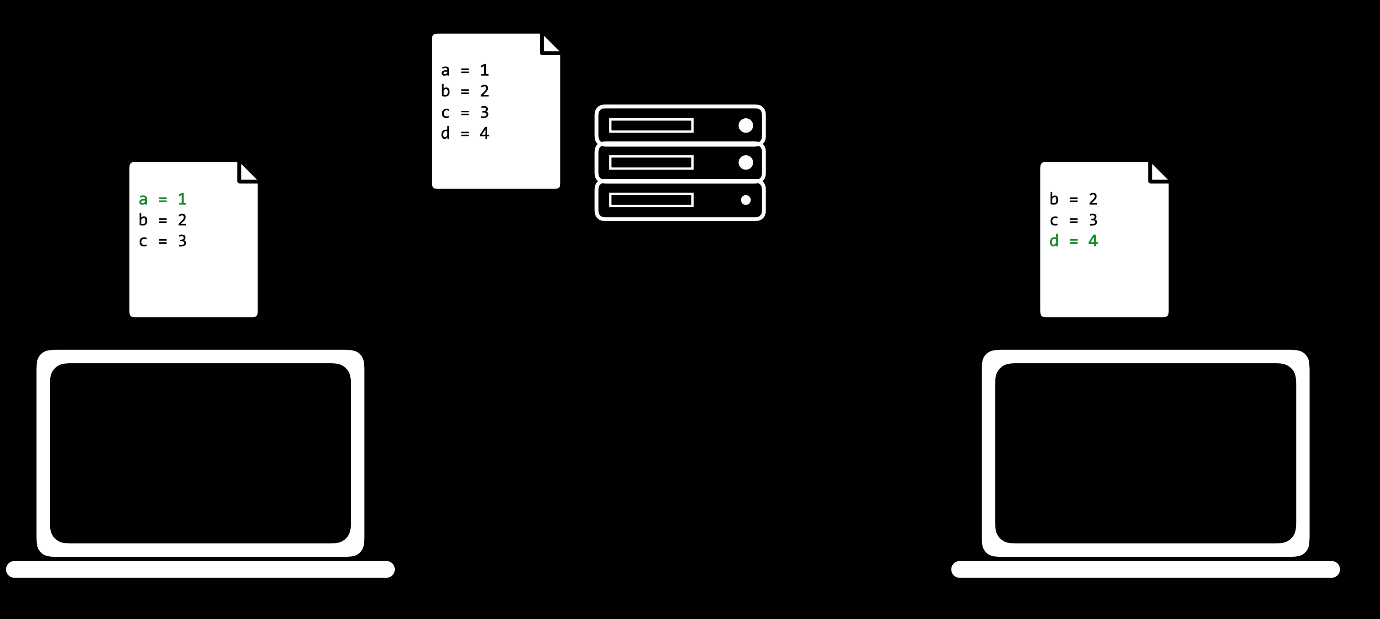
**[Git](https://cs50.harvard.edu/web/2020/notes/1/" \l "git)**

* [Git](https://git-scm.com/) is a command line tool that will help us with version control in several different ways:
  + Allowing us to keep track of changes we make to our code by saving snapshots of our code at a given point in time.



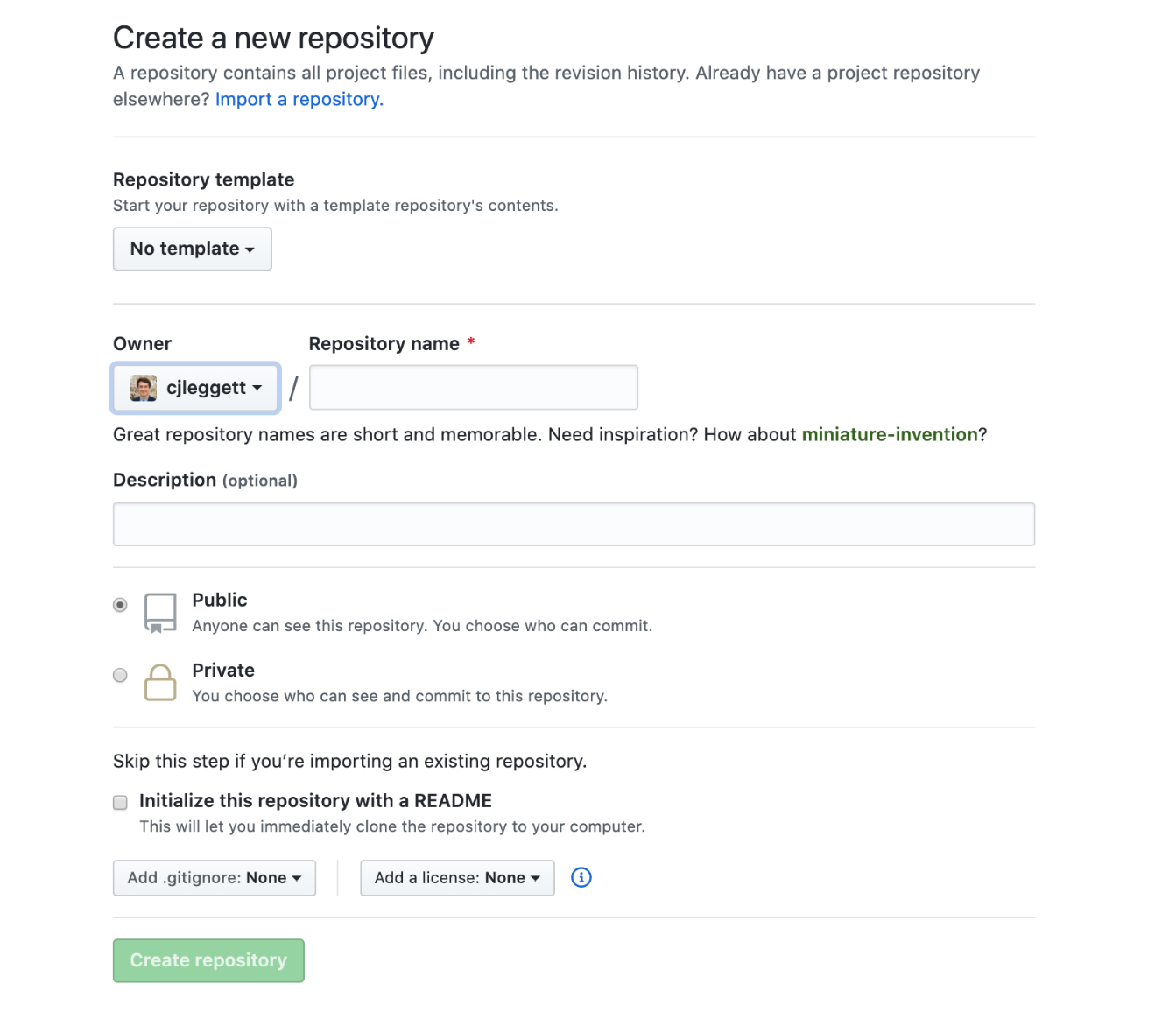
* + Allowing us to easily synchronize code between different people working on the same project by allowing multiple people to pull information from and push information to a repository stored on the web.



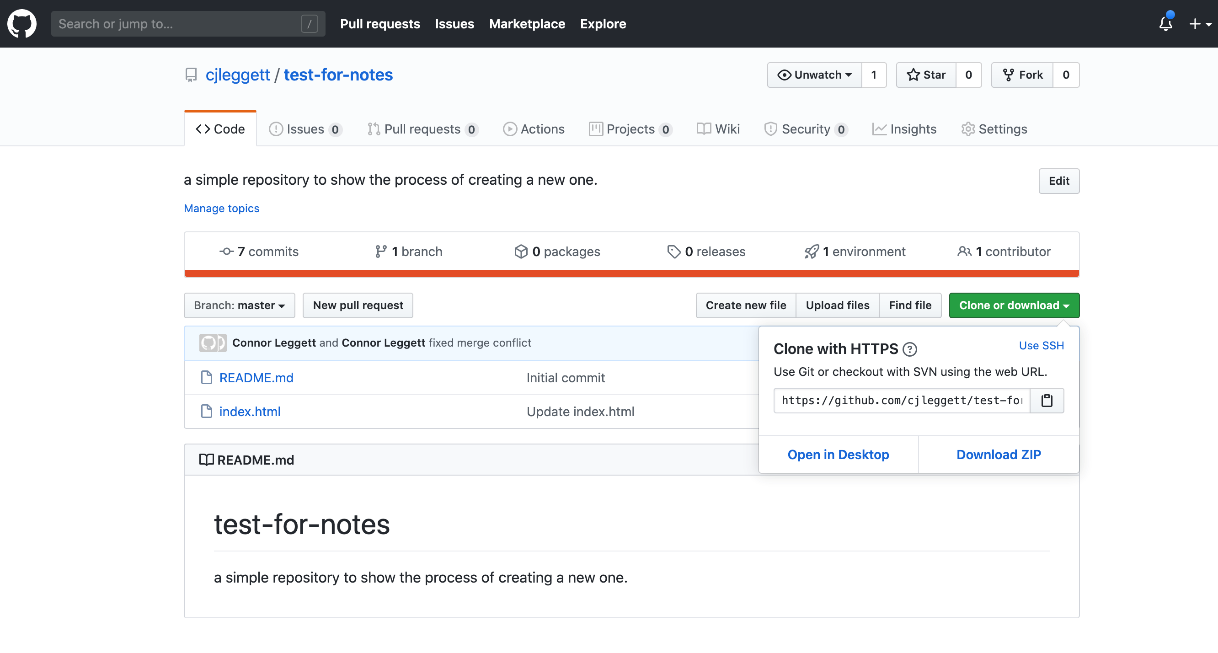
* + Allowing us to make changes to and test out code on a different *branch* without altering our main code base, and then merging the two together.
  + Allowing us to revert back to earlier versions of our code if we realize we’ve made a mistake.
* In the above explanations, we used the word **repository**, which we haven’t explained yet. A Git repository is a file location where we’ll store all of the files related to a given project. These can either be remote (stored online) or local (stored on your computer).

[**GitHub**](https://cs50.harvard.edu/web/2020/notes/1/#github)

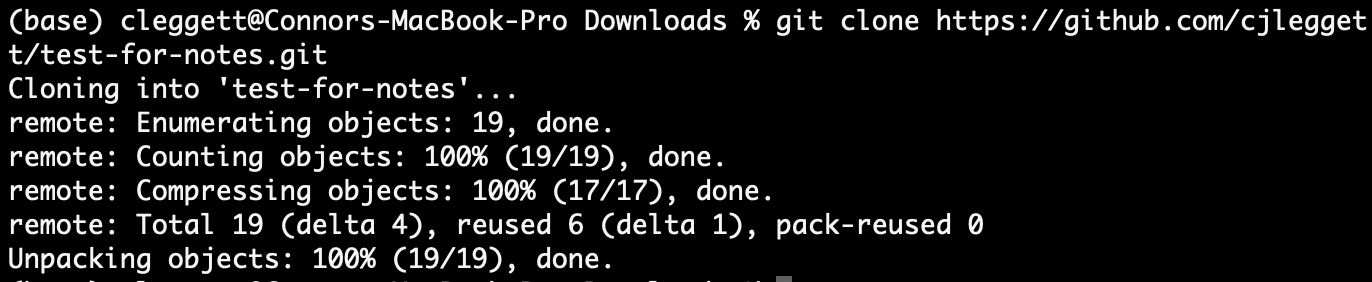
* [GitHub](https://www.github.com/) is a website that allows us to store Git repositories remotely on the web.
* Let’s get started by creating a new repository online
  1. Make sure that you have a GitHub account set up. If you don’t have one yet, you can make one [here](https://github.com/join?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home).
  2. Click the **+** in the top-right corner, and then click “New repository”
  3. Create a repository name that describes your project
  4. (Optional) Provide a description for your repository
  5. Choose whether the repository should be public (visible to anyone on the web) or private (visible just to you and others you specifically grant access)
  6. (Optional) Decide whether you want to add a README, which is a file describing your new repository.



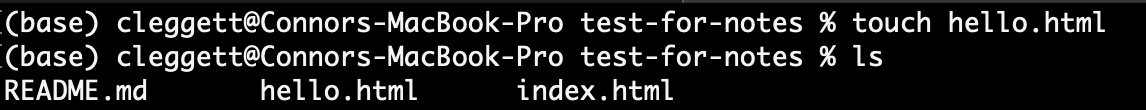
* Once we have a repository, we’ll probably want to add some files to it. In order to do this, we’ll take our newly created *remote* repository and create a copy, or clone, of it as a *local* repository on our computer.
  1. Make sure you have git installed on your computer by typing git into your terminal. If it is not installed, you can download it [here](https://git-scm.com/downloads).
  2. Click the green “Clone or Download” button on your repository’s page, and copy the url that pops down. If you didn’t create a README, this link will appear near the top of the page in the “Quick Setup” section.



* 1. In your terminal, run git clone <repository url>. This will download the repository to your computer. If you didn’t create a README, you will get the warning: You appear to have cloned into an empty repository. This is normal, and there’s no need to worry about it.



* 1. Run ls, which is a command that lists all files and folders in your current directory. You should see the name of the repository you’ve just cloned.
  2. Run cd <repository name> to change directory into that folder.
  3. Run touch <new file name> to create a new file in that folder. You can now make edits to that file. Alternatively, you can open the folder in your text editor and manually add new files.
  4. Now, to let Git know that it should be keeping track of the new file you’ve made, Run git add <new file name> to track that specific file, or git add . to track all files within that directory.



[**Commits**](https://cs50.harvard.edu/web/2020/notes/1/#commits)

* Now, we’ll start to get into what Git can be really useful for. After making some changes to a file, we can *commit* those changes, taking a snapshot of the current state of our code. To do this, we run: git commit -m "some message" where the message describes the changes you just made.
* After this change, we can run git status to see how our code compares to the code on the remote repository
* When we’re ready to publish our local commits to Github, we can run git push. Now, when we go to GitHub in our web browser, our changes will be reflected.
* If you’ve only changed existing files and not created new ones, instead of using git add . and then git commit..., we can condense this into one command: git commit -am "some message". This command will commit all the changes that you made.
* Sometimes, the remote repository on GitHub will be more up to date than the local version. In this case, you want to first commit any changes, and then run git pull to pull any remote changes to your repository.