Git command line tutorial

Step 1: Create a local repository

cd (change directory) For example, if you have a 'projects' folder on your desktop, you'd do something like:

* cd ~/Desktop
* mkdir myproject
* cd myproject/
* git init

Step 2: Add a new file to the repo

Add a new file to the project, using any text editor you like to running a touch command.

* touch mnelson.txt
* ls
* git status

What this basically says is, "Hey, we noticed you created a new file called mnelson.txt, but unless you use the 'git add' command we aren't going to do anything with it."

Step 3: Create a commit

* git commit -m “my first commit”

Step 4: Create a new branch

Say you want to make a new feature but are worried about making changes to the main project while developing the feature. This is where [git branches](https://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is)come in.

Branches allow you to move back and forth between 'states' of a project. For instance, if you want to add a new page to your website you can create a new branch just for that page without affecting the main part of the project.

Once you're done with the page, you can [**merge**](http://git-scm.com/docs/git-merge) your changes from your branch into the master branch.

Let's say you are on the master branch and want to create a new branch to develop your web page. Here's what you'll do: Run**“**[git checkout -b <my branch name>](http://git-scm.com/docs/git-checkout)**”**. This command will automatically create a new branch and then 'check you out' on it, meaning git will move you to that branch, off of the master branch.

* git checkout -b my\_branch

You'll want to change the URL in the first command line to what GitHub lists in this section since your GitHub username and repo name are different.

* git remote add origin https://github.com/cubeton/mynewrepository.git
* git push -u origin master

Step 5: Push a branch to GitHub

Now we'll **push** the commit in your branch to your new GitHub repo. This allows other people to see the changes you've made.

If they're approved by the repository's owner, the changes can then be merged into the master branch.

To push changes onto a new branch on GitHub, you'll want to run [**git push**](http://git-scm.com/docs/git-push)**origin yourbranchname**. GitHub will automatically create the branch for you on the remote repository:

* git push origin my-new-branch

git push origin master: push the local changes to the remote repository.

git pull origin master –allow-unrelated-histories: pull the remote repository from GitHub and ignore the different files

Create new branch:

* git checkout -b yilinBranch
* git pull origin master
* git status
* git add ‘.\myFolder\yilinfile.sql’
* git status
* git commit -m ‘initial commit’
* git push --set-upstream origin yilinBranch