Requirements

1. Free github.com account
2. Watched and learned the **command line videos (CLI)**

Making a PUBLIC ssh key

NEVER share your PRIVATE key

1. Open the terminal
2. type in:

```
ssh-keygen -t rsa -b 4096 -C "youremail@email.com"
```

3. Change directory (**cd**) into the ~/.ssh directory

```
cd ~/.ssh
```

4. Type this in to copy the key you just created directly to your clipboard. Once you type this, you can be ready to paste it.

FOR macOS

```
cat ~/.ssh/id_rsa.pub | pbcopy
```

FOR Linux

```
cat ~/.ssh/id_rsa.pub | xclip -selection clipboard
```

FOR Windows

```
cat ~/.ssh/id_rsa.pub | clip
```

1. Adding the **public key** we just made to your github account
2. This will allow you to authenticate your computer to the github server
3. Click your **profile picture** top right of page
4. Click **Settings**
5. Click **SSH & GPG Keys**
6. Click **New SSH Key** on the top right
7. Paste the key from step 4 in **Making a public key** into the **Key** box.

Hot Terms

Repository - A folder for your project. Git keeps all the project's files and their history here.

Branch - A separate area in your project. It's like a new working copy where you can try out things without affecting the main part.

Remote - Another place where your code is stored, usually on the internet, where everyone can share their changes.

Commit - Another place where your code is stored, usually on the internet, where everyone can share their changes.A saved change in your project. It's like a checkpoint that keeps a record of what you did.

Merge - Putting pieces of code from different branches together into one branch.

Pull - Getting changes from the remote project and adding them to your local version.

Push - Sending your saved changes to the remote project so others can see or use them.

Top Commands

Initialize git inside of the current directory

```
git init
```

Clone or copy a repo from an existing repository's URL. I added a sample URL here

```
git clone git@github.com:eugeniosp3/pragmatiCoders_ETL_ELT_Video.git
```

git add [file] or git add . will move all changes to staging and mark them for inclusion into the next commit

```
git add .
```

git status is used when you want to check the status of what's ready to be commited or included in the next commit

```
git status
```

git commit requires you to enter -m and a message. This message will be useful to describe what you are adding with this commit to the repository/branch.

```
git commit -m "commit message/note"
```

This command tells Git to push the local, on your computer, information to the main repository to the upstream remote repository, of whatever alias. Commonly it will be origin.

```
git push -u alias branch
```

List all of your branches.

```
git branch
```

Change to a different branch

```
git branch branchName
```

Show a list of all of the current configured remote repos. You use this when you want to check if you're on the right remote repo.

```
git remote -v
```

You're going to use this when you're wanting to pull any updates from a repo on a remote, server, to your local repo. Manual update on github.com and you want it to show up locally.

```
git pull
```

When you want to merge a branch into the current branch you would use merge. If there is a conflict you may have to force it, but forcing things is NOT recommended.

```
git merge
```