Git Hub Steps:

1. **Git Initial Repository Setup:**

Note: There are two ways to accomplish connecting a local repository to a remote repository. The first is when you already have a **local** folder with code and you want to connect it to a **remote** repository in GitHub. The following is to make that connection.

Git Repositories now have a standard branch called "Main" that is created when a git repository is initialized. This is a change from the previous standard branch called "Master". Our computers still initialize new **local** repositories as "Master". To be in sync with GitHub, we need to configure our **local** machines to init as Main instead of Master.. running this line of code will permanently set the config to Main instead of Master…

Run this once: **git config --global init.defaultBranch main**

CONNECTING EXISTING LOCAL PROJECT TO REMOTE REPO
IN TERMINAL..
1. CD into your main code folder / whatever folder holds your projects / code files. (Each code project should have its own folder as we cannot have nested repositories. To start a new project)

2. Create a folder (mkdir folder-name) for your new project **OR** cd into the already existing project you want to commit to the **remote.**

3. Run **git init** to initialize the new **local** repo in the current project folder.

4. If you do not have any files to commit yet, you will not be able to make a commit. It will throw an error. So you will need to create something, even if it is as simple as **touch test.txt** just to create something to commit. If you already have files.. nothing to sweat!

5. Next step is to go to your GitHub account in the browser.

6. In your account, you will want to be on the page with your repositories (not projects!).. There should be a green button that says "new".

7. Click on "new"!

8. Type in the name of the project… HIGHLY RECOMMEND (for many reasons).. NO CAPITAL LETTERS AND NO SPACES.. hyphen/dashes/underscores are ok!

9. SET TO PUBLIC if you want to be able to share with others

10. DO NOT CREATE A README.. it will make your life easier if it isn't created!

11. Click create!

12. On the next page you will see a set of instructions for connecting via command line (terminal).. We are going to run SOME of these commands.. in the same order listed, but not all of them…

13. Back in your terminal (which should still be in the same folder as your project… check with "pwd") we are going to run the following commands

    a. git add -A (sets up all uncommitted files for staging)
    b. git commit -m "message" (adds file for committing w/ msg)
    c. git remote add origin <paste the URL>
    (connects local repo to remote repo.. should see the line of code to copy / paste in the instructions)
    d. git push -u origin main

14. You're all set!

Further commits can be done with:

1. **git add -A**
2. **git commit -m "msg"**
3. **git push**

2. **GIT PROJECT SETUP WHEN LOCAL FOLDER HAS NOT BEEN STARTED YET, BUT REMOTE PROJECT EXISTS**

1. In terminal:
    1. cd into folder where you want your new project folder to be created
2. In GitHub
    1. In GitHub / in browser.. navigate to your project repo you want to be connected to and work on locally / on your computer.
    2. There should be a green code button with a drop down menu.
    3. In the drop down menu there should be a link to the project folder that looks like a url. **COPY THIS LINK**

3. In Terminal:
    1. Run: **git clone** <<paste link here>>
    2. Should be all set!

Can double check the link is correct by running: **git remote -v**

This will tell you what url the git file is connected to. If there’s an issue.. you can always delete the git init file without losing any code.. “rm -rf .git”… this will delete the git init file and allow you to create a new .git by running “git init”. BE VERY CAREFUL WITH “rm -rf” it can literally erase your entire computer… make sure you put the exact name of the file you want to delete after the rm -rf <<namehere>>. Otherwise you might regret it!