What is Git?

Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.[1] Most programmer nowadays is using github instead of common tools such as Google Drive. It is used to manage the change to the system, which is from program, website to all kind of project that we are dealing with. When you want to start working with git, you must install git to your computer first, you can check whenever you have already had git on your computer or not by using the command git –version. If your computer does not have one, you can download it online. We recommend you using the Visual Studio Code for using Git since it supports many different programming languages and also Git



Reference: <https://git-scm.com/>

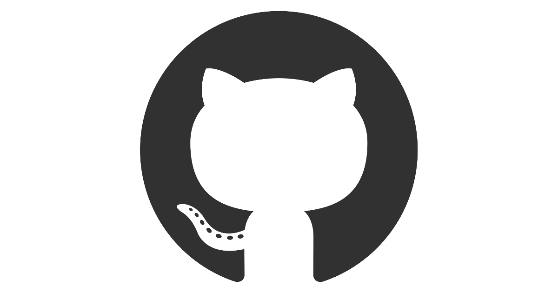
Repository:

Repository is mean your own project. When we are talking about coding, it is usually used to describe the project that you upload into git which is the folder that contain all the detail that makes up your project. It will be containing all the folder and coding files that made up your program.



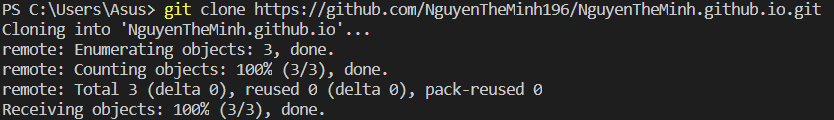
Github:

Github is the online website that contain the repository so that the other people can also join in and work with your project



Git clone:

This is a command that can be used to bring a repository that is hosted online in website such as github to a folder on our local machine. This is the folder where you can do edit to the project offline and then send it back online to make changes to your project



The combination of git add – git commit – git push:

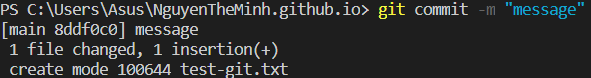
Git add:

This command is used to track down all the files and the changes related to the project. After the command git add, you are required to input the file name that you want to add or every file by using (.) function



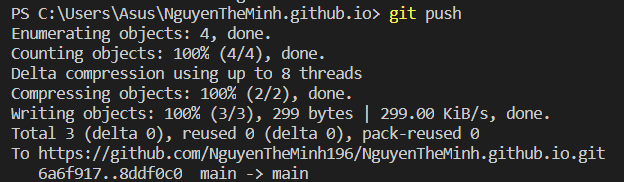
Git commit:

This is a command that is used to save all the changes that we have tracked from the Git add command to Git. For the Git commit command, you are required to insert a message to explain what you have done for it to come with all the changes you made. When the other see the changes that you made, this message will help them have a general idea of what you did.

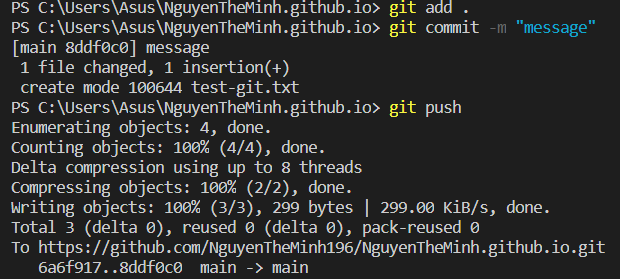


Git push:

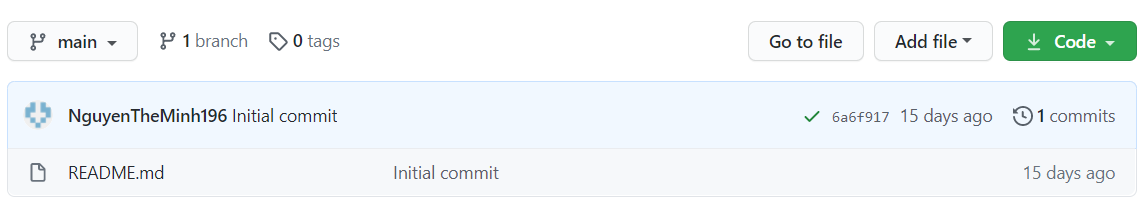
This command is used to upload all the files that we have saved in the git commit command to the online repository such as Github. After using all the command, the other people will be able to see the changes you made to the project. This series of command is used to record all the changes that you made and upload it online for the other to see.



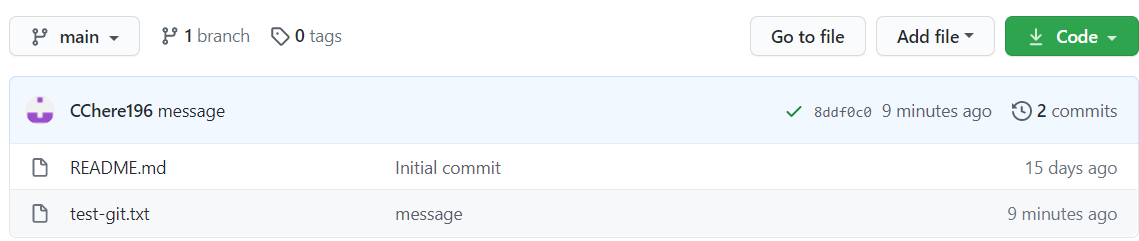
Full example for the combination:



Before the combination git add – git commit – git push:



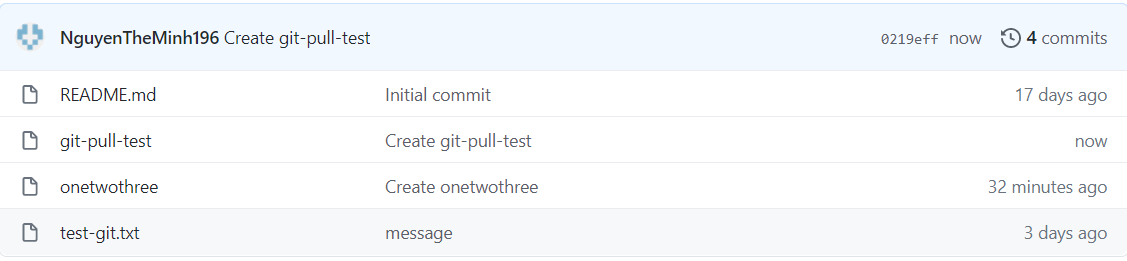
After the combination git add – git commit – git push:

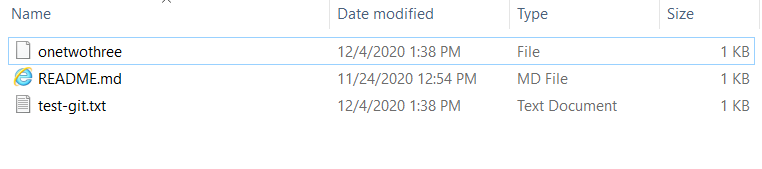


Git pull:

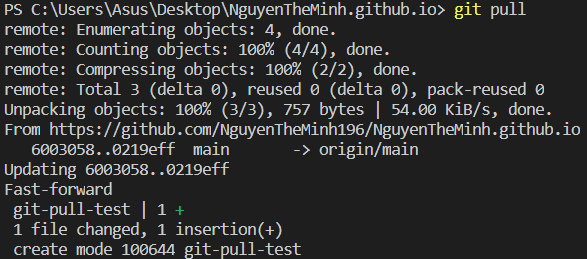
This command is used to download all the changes from the online repository to our local machines. This can also be used to fix the problem of having 2 files of the same name while one online and the other in your local machine.

Before git pull:

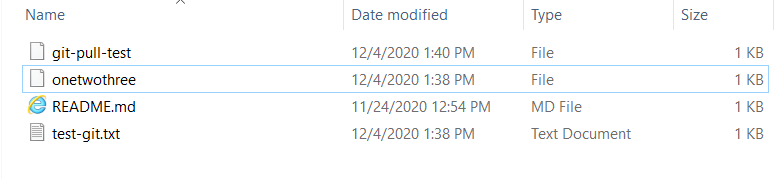




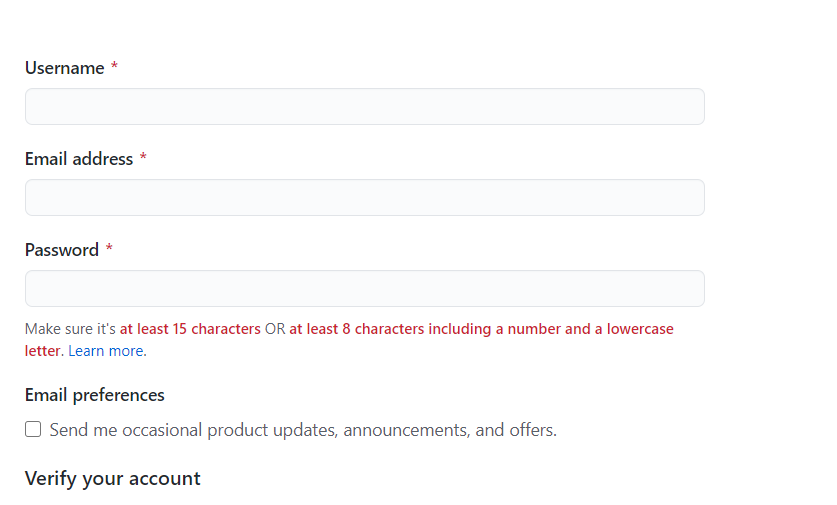
The code git pull

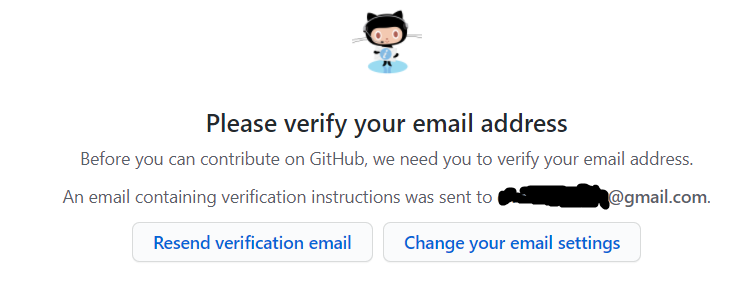


After git pull



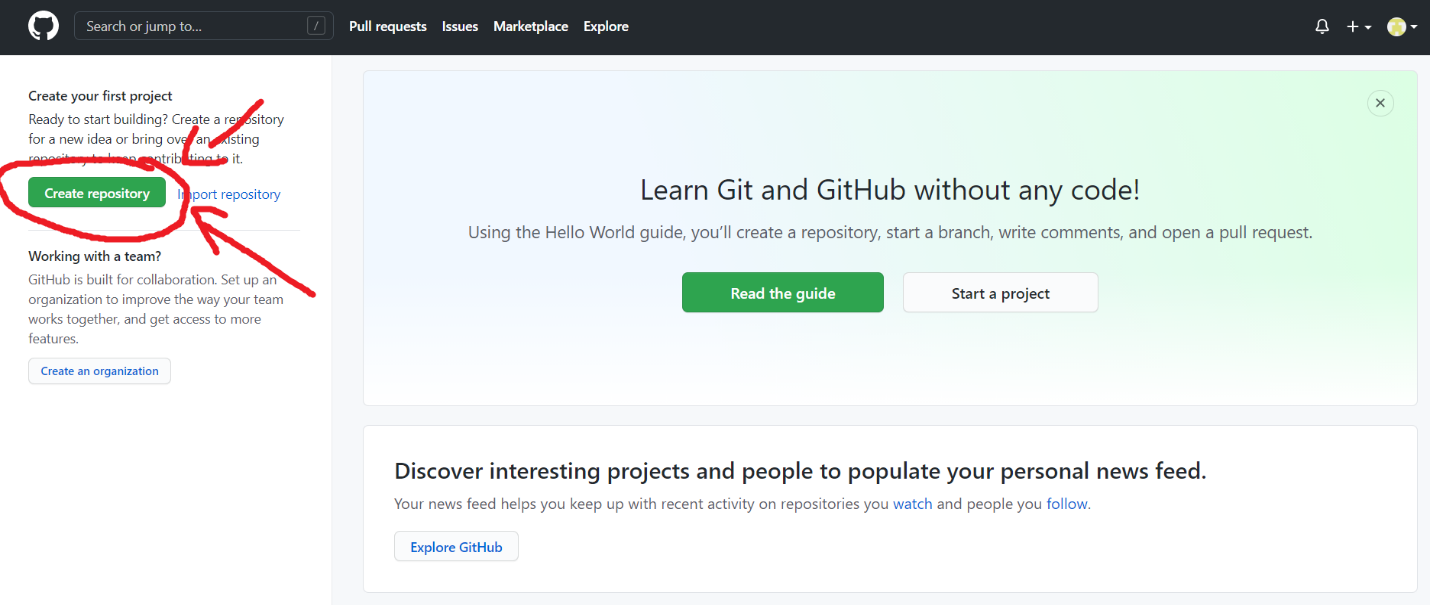
Sign into github:  
When signing into Github, you are required to input the username, email address, password and finally verify the account by solving the puzzle below. After finishing setting up the account, you will be asked some questions about the reason you choose to use github from what kind of work do you do, programming experience to why you choose github and your interest. After answered all those questions, you will be asked to verify your email. Finally, you have finished signing up for Github.



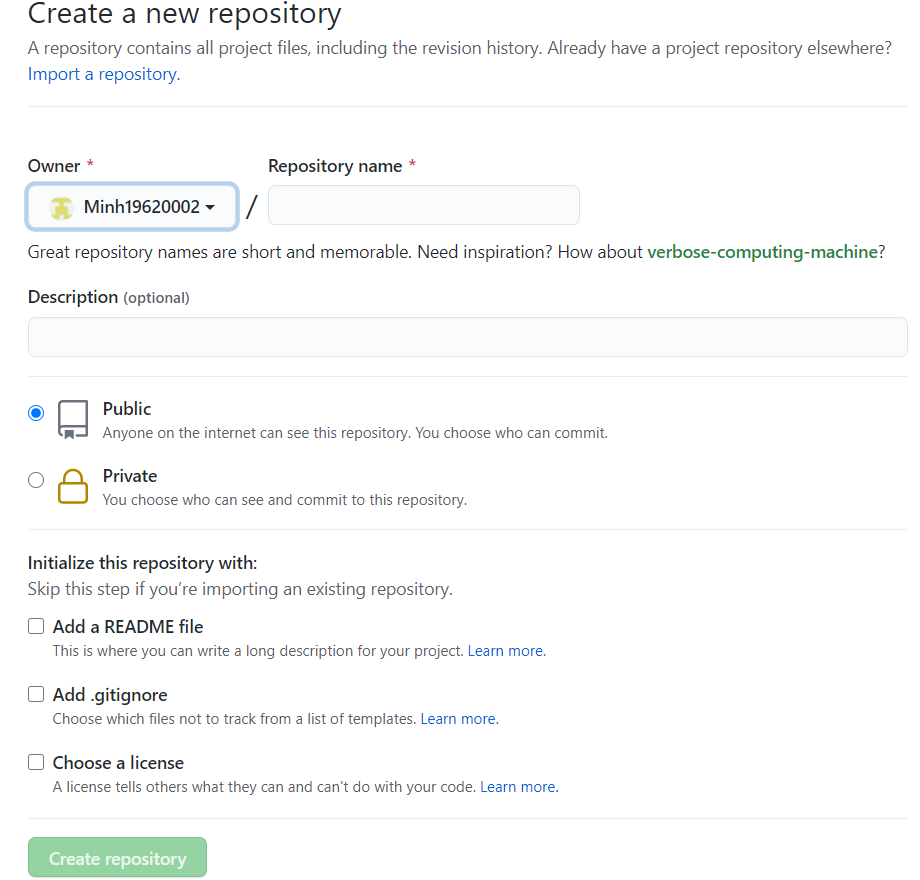


Creating a repository:

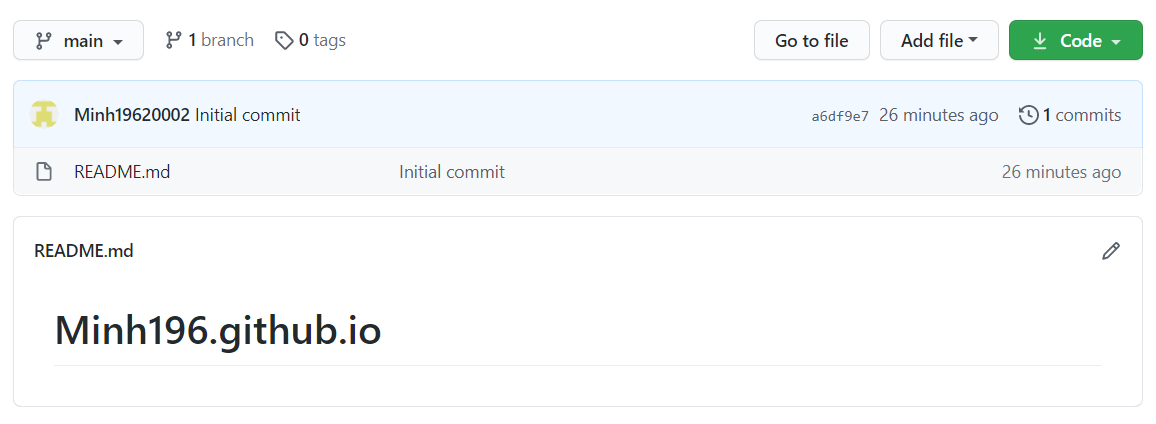
The first step that you should be doing after signing up for a github account is to either create a repository or import other repository. You can create one by selecting the “Create repository” button on the left. This is also the place where it will show all the repository you are currently dealing with and your teams.



After pressing the Create repository button, it will lead you to this page where you are required to input information.

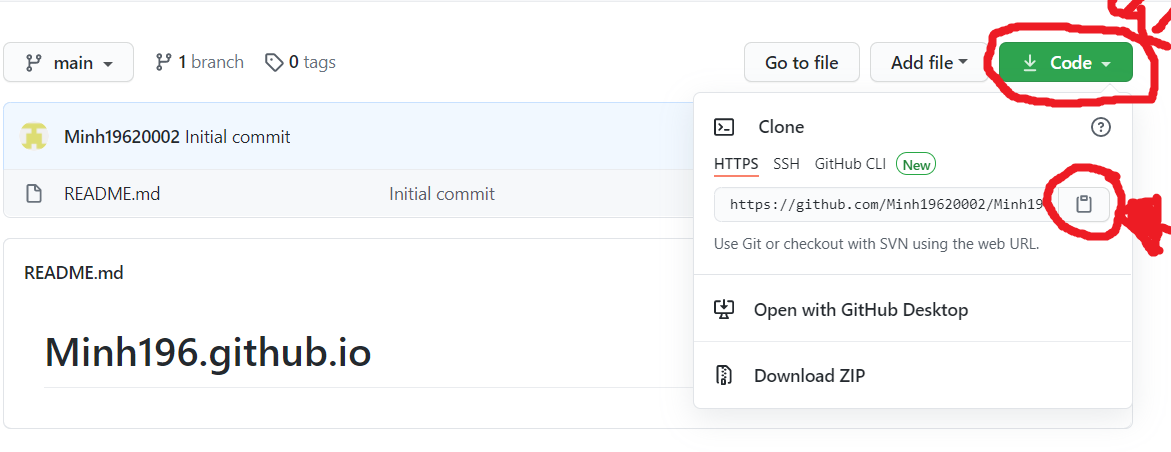


A github repository will be looking like example 3 when you create one.



**Git clone:**

This command is used to clone the repository from online to your local machine which can later be used to edit directly



You can clone it to your local machine by using the command: git clone (the link you just copied). An important thing is that it will clone to the directory that you are currently at which is in the desktop in the example:



cd:

cd which is also known as change directory is an important command that you also must be mindful of when dealing with. We can know where we are currently are from the first part of the code which is C:\Users\Asus in this case and when I use the code cd <a file directory>, I arrived at the folder. With this code, we can go to a specific folder to clone the repository online. There are 2 primary kinds of cd which is going in by using the command cd <a file directory> and going out by using cd. which bring you back to file before that. It is crucial for you to know where you are so as you will not get lost and forget where you put the project at.



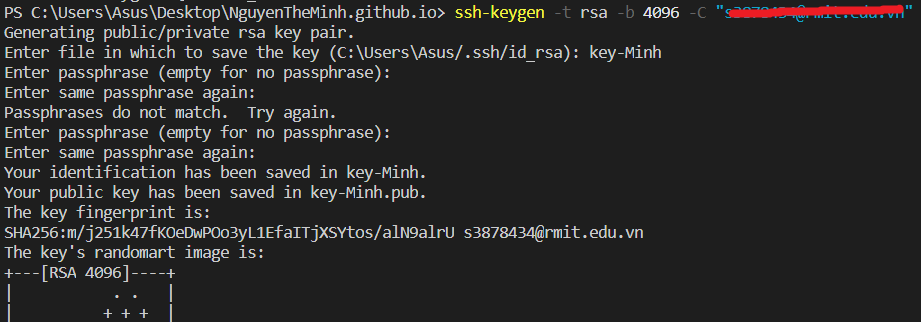


Git config commands:

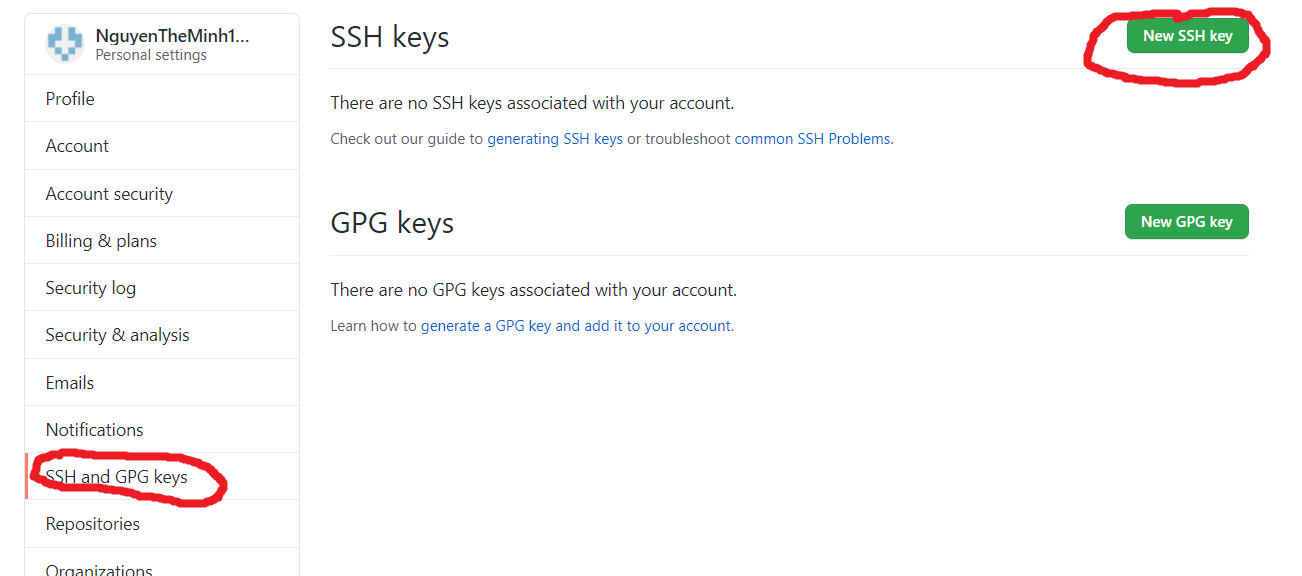
Git config commands are a bunch of commands that will help you setup your git on your local machine. The two must fix when you are working with Git are creating a name and email for your local email when you are uploading to the online repository so that the other will know who upload those files and how to contact you. The command for that is git config –global user.name/email:



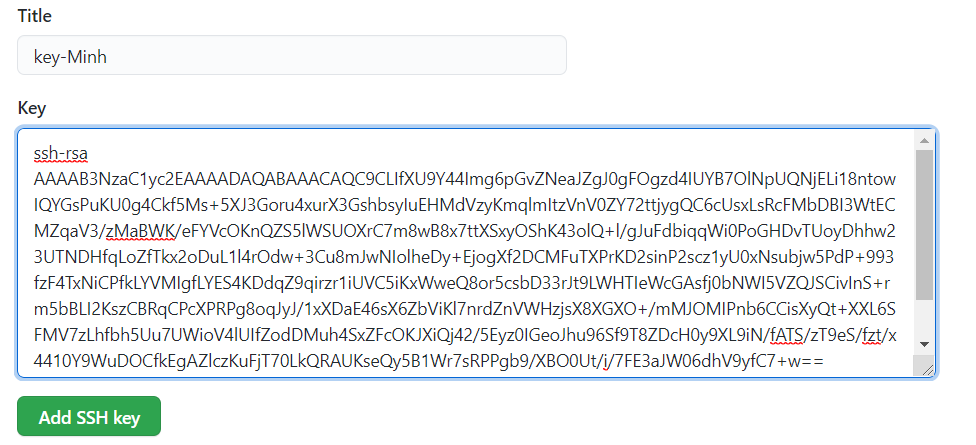
Etc: SSH key  
If you want the github account to understand when you are pushing some files online that It is you who are that account, you need to connect something called SSH key from your local machine to your github account. To do that, firstly, you have to generate a ssh key in your computer first by typing the command ssh-keygen -t rsa -b 4096 -C <your github email address>. After that, you are required to input the file’s name where it will keep the key and a password.



After creating the key, if you get into the folder where you are currently at (directory), you can see there is 2 new files. One is the ssh-key’s name, and the other is ssh-key’s name pub. The file with the pub means that it is something that can be shown to the public while the other one is something you should only keep it for yourself. When you upload the file online, the program will know your private ssh-key that will lead to your public key to identify that the pusher is you not some anonymous computer. That’s why the next step is to get to github and input the ssh key to your account. To do that, you must go into setting and look for the SSH and GPG keys and click the “New SSH key” button



After clicking the button, you will be lead to a place for you to setup your key and in this case, I will be seting up mine public key by putting in the title and the content inside the file key. After finish putting in the content, you just need to add the SSH key and then later on, when you are working with github, they will be able to know it is you by using your private ssh-key.



Git branching:

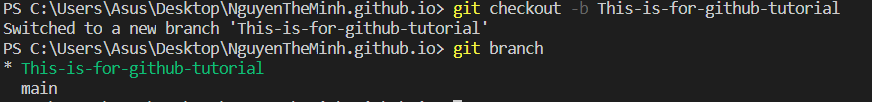
When we are working with a project, you are bound to encounter bugs and problems that you need to fix. That is why you will be needing this feature. Branching is creating the same program and you can change it without affecting the original. With the new same program, you can add feature, fix bugs all you want and when there is no more problem, you can merge it with the master branch for it to be have all the feature that you want. This is to avoid any problem that will break your entire program just because some new lines of code that you just created. There are many different branching commands in git, but we will look at some of the most common commands.

Git branch will show you all the branch that your github currently has and in this example, it only has one master branch which is the main branch. The star sign next to the master mean that we are currently inside that branch.



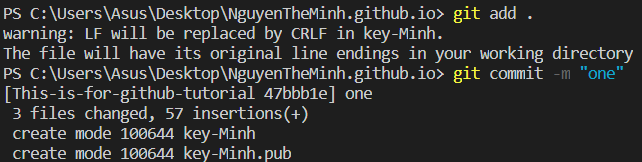
Git checkout -b:

Git checkout is used to switch between branches but when combining with -b, it will create a new branch. The full command is Git checkout -b <branch’s name>. It is important for you to name the branch’s name specifically so that when the other look at it, they will also be able to understand what this file is for.

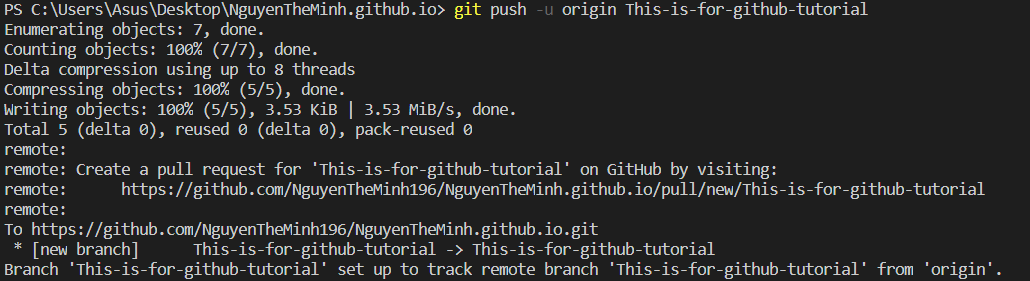


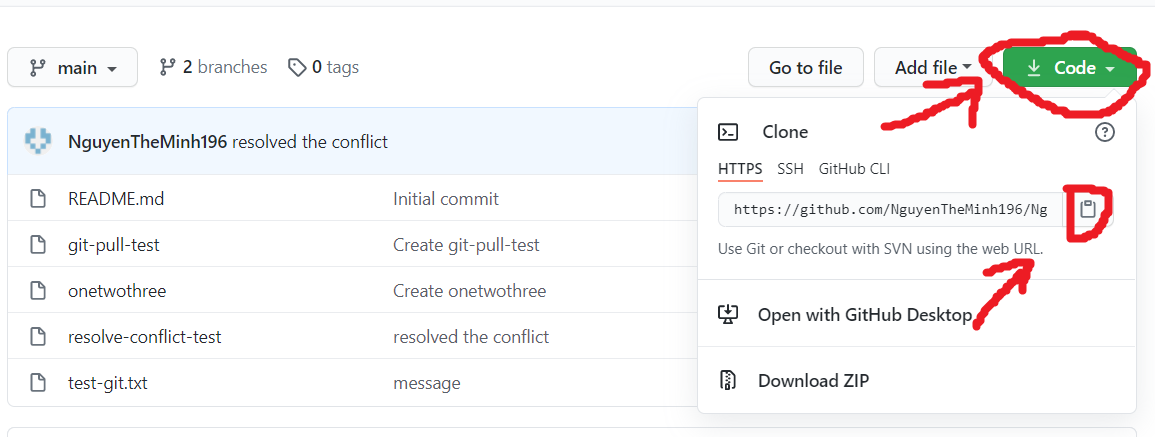
Making changes to the branches

After making changes in the branch, you use the combination add and commit.

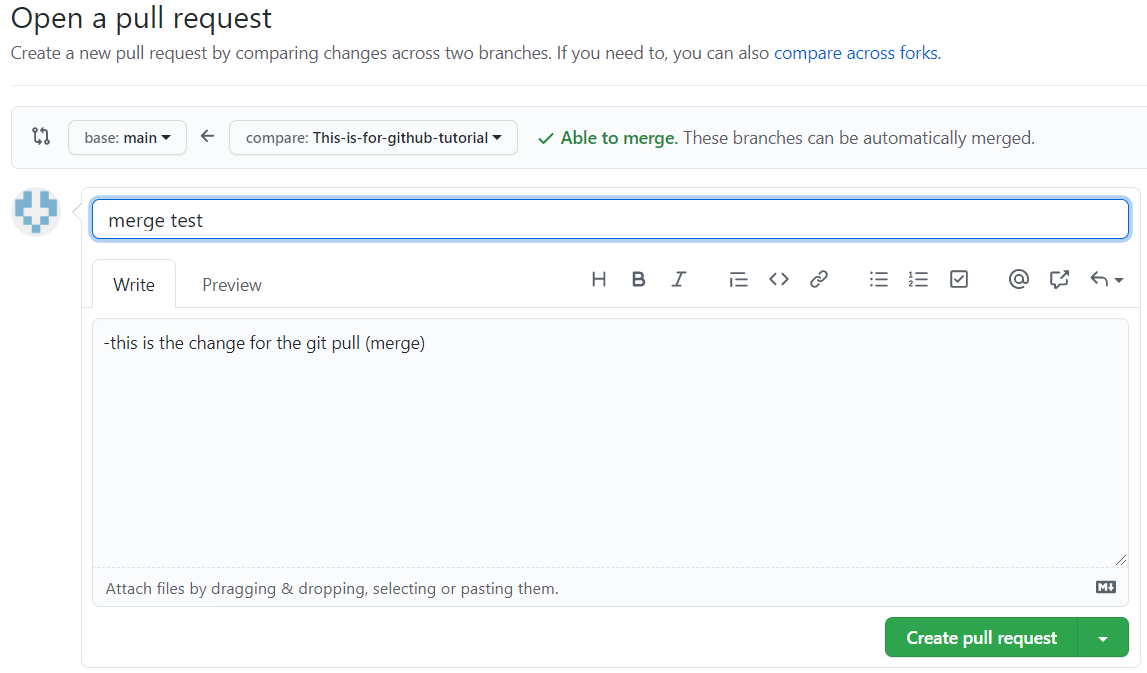


The problem here will be related to the final step which is the git push command. As you know, we have only created the branch locally so online, it has yet to even has that branch to make the changes yet. That is why you are required to push all the branch to the github. The command for this is: git push -u origin <file name>



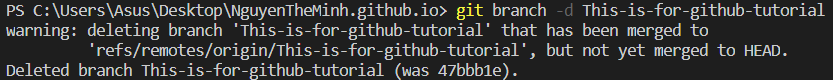


You can merge both branch and the master branch by using the git merge command but you should only do this when you have finished all of your problems so that there won’t be any problems when merging. You can also merge online by clicking the compare & pull request on Github. It will allow you to make request for the administrator of the project to accept the merge request.

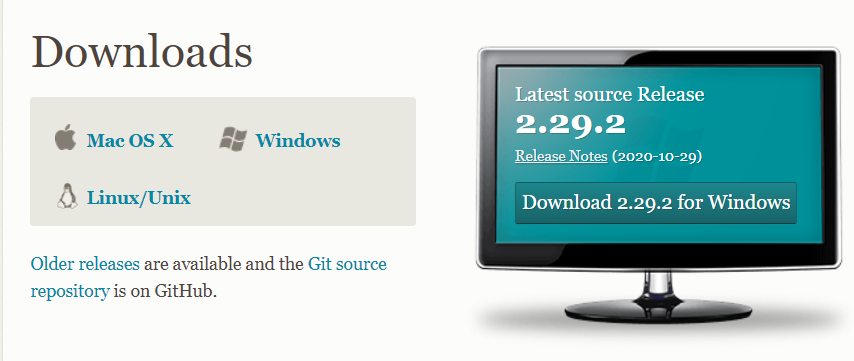


When you finish merging and no longer needed to use the branch anymore, you can use the command git branch -d (branch name)

This warning is for the case when you have accidently deleted the branch when you have yet to merge so make sure to merge before you delete the branch.



<https://git-scm.com/downloads>



Resolve conflict:

