Git & GitHub

What they are and how to use them

I always thought about Git and GitHub as the same thing. In fact, they are separate things. There are also GUIs that allow you to manage the remote repositories.

While Git is the language of source code control, GitHub is a web-based repository storage solution. You don’t have to store repositories on GitHub, and there are competitors. GitHub was the first and remains the most popular.

A repository is just a folder where files are stored. The files don’t have to be code files, they can be text files or anything else. For this discussion we’ll stick to software and text.

Putting a bunch of files in a folder, doesn’t make it a repository. The repository must be created. It’s a good idea to start a new repository on your local machine. Git can manage a repository stored anywhere, including the local machine.

Before starting download Windows Terminal if you don’t already have it. Then go to [www.git-scm.com](http://www.git-scm.com) and download a copy of Git.

**Creating your first Repository**

Create a local directory where you want your repository to exist. This folder will be the repository. You can put the files you want to track into this folder before or after creating the repository.

The command:

git init

Will create the repository, effectively creating a subfolder called .git. This folder will hold all the SCCS tracking information.

Before doing anything else with this folder create a file called .gitignore, using any editor in the root of the repository. To this file add \*.log to ignore log files.

Add a new line \*\*/logs # if you have a folder that contains log files

Save the file

Before proceeding we must commit this .gitignore file.

Git status will show that the .gitignore exists in you repository but is not being tracked.

Git add .gitignore #adds the file to be committed (ie staged)

Git commit

The editor you chose will pop up allowing you to indicated what change was made. You must add an explanation of what was changed / added. If you don’t explain the change, the commit will be aborted. The commit message is important and should explain what is changing clearly.

To commit another file:

Git add filename # adds one file

Git add . # adds all files

Git commit -m “short message” # a shortcut that doesn’t open a notepad file

Git log # shows what branch you are on and where you are in the SCCS (HEAD) normally

# at the end of the chain

**Making changes to files**

We don’t make changes to files on the main branch of the repository. Instead, we create a branch, edit the files in the branch, and when we’re satisfied that the changes are what we want, we merge them back to the main branch.

To create a branch:

Git branch branchname # make the name meaningful

Example:

Git branch bug-FixBadCall # or something else meaningful

Git status will now show a new branch with the current branch preceded by ‘\*’ indicating which branch you are currently on. To switch to the new branch, we just created we type:

Git checkout branchname # switches to the new branch

If there are multiple branches:

Git checkout branchname

Once on the new branch, make changes to the files as required. When the changes are made do:

Git add filename # file name changed

Or

Git add . # add all files that were changed

Then

Git commit

If we need to delete a branch:

Git branch -d branchname # deletes the branch. The ‘main’ branch is usually not deleted

So, it’s a two step process:

Git branch branchname # create the branch

Git checkout branchname # switch to the branch

There is a shortcut that creates the branch and switches to it in one step:

Git branch -b branchname

**Remote Hosting**

If you are working alone, there is no need to remotely host your repos, but if you are working on a project with others remote hosting is essential. There are a few companies who offer remote git hosting. The most popular and well known is GitHub.com, and that’s the one we’ll cover.

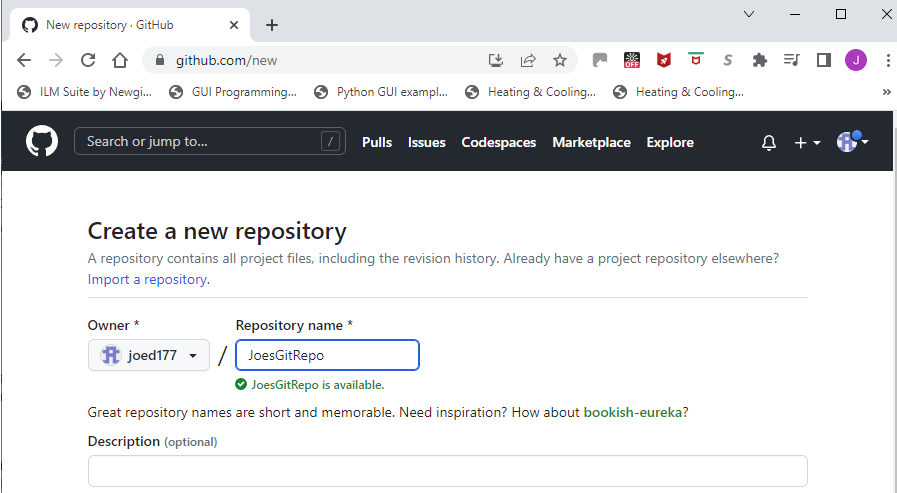
If the number of people in the team is small, typically five or less, and the number of minutes of processing is small, GitHub is free. GitHub provides unlimited public or private repos. If you intend to do more than what’s allowed, you can purchase more capacity. GitHub provides a mobile client and the Git Desktop. The cost of additional features is modest.

Repositories on GitHub can be public or private. Private repos can not be seen by others, unless you provide a link. Public repositories are visible to others, but others can’t change the code, though they can make a copy (clone) of the code for their own use.

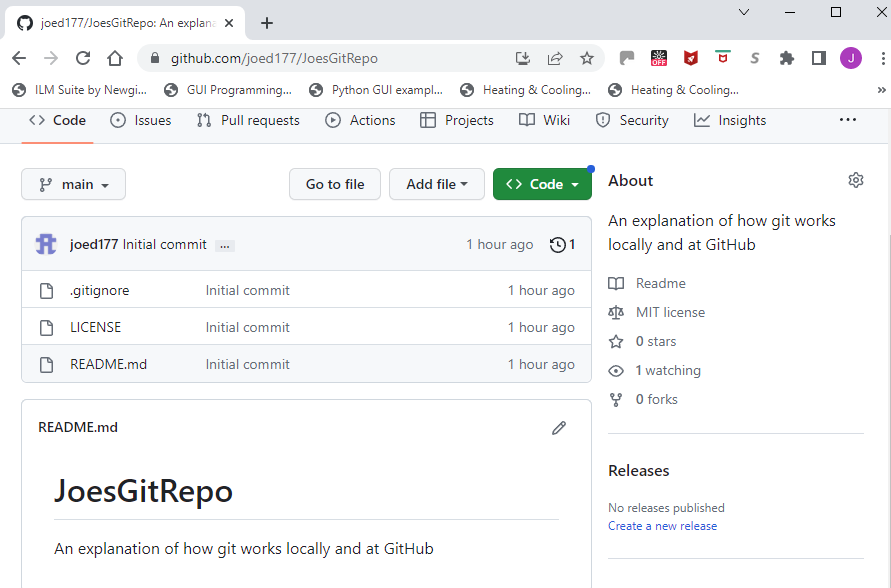
GitHub offers 11,000 actions such as building and deploying your code in addition to source control. Some actions are free, others are fee based.

Obtain a free account and log into the account to follow along.

We are logged into GitHub.com and we are creating a repository called JoesGitRepo.



We’re choosing to make this repo public. We’ll add a brief description of the repo, then we’ll add a readme file, we’ll select the MIT license, and a .gitignore file using the VisualStudio filter. Visual Studio creates lots of files that change every time it is run such as a .exe and .sio files and these will be different on different machines, so there’s no need to track changes to these files. Then we’ll click the Create repository button in the lower right hand section of the screen. With that the repo is created.



Notice that three files have been created and committed by GitHub. Also notice that we are on the main branch. At this point, there is only one branch. Eventually we’ll want to pull down a copy of the JoesGitRepo repository. When we do this for the first time, the folder containing the local repository should not have a .gitignore, or a README.md.

Once the remote repo has been created, we will want to bring it down to our local machine. To do this, we’ll want to clone the remote repo. If we click the Code button outlined in green above, you will have the option to clone the repo. The first time you do this you will be asked to log in. Once you do this, your credentials will be saved and you will not have to log in again for this repo.

Select HTTPS, outlined in green, then click the icon outlined in red which makes a copy of the URL in the image below. Then on your local machine enter:

Git clone <https://github.com/joed177/JoesGitRepo.git>

A folder named JoesGitRepo will be created on the local machine, and it will include the files that came down from GitHub.

