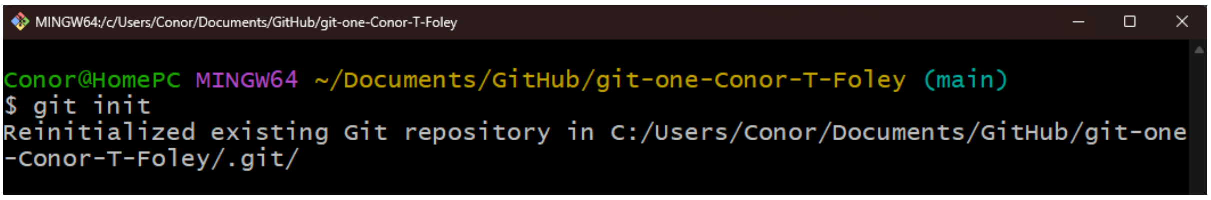
Git Lab 1 – Git Commands

git init

Creates a new repository in the folder we created in step one.

Creates a hidden file in the root directory .git which can be brought to view using the view tab and then checking the “Hidden Folders” check box.

This creates a master repository on the machine that the user is sitting at.



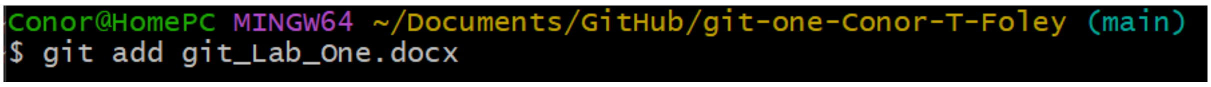
MINGW64:/c/Users/Conor/Documents/GitHub/git-one-Conor-T-Foley

```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (main)
$ git init
Reinitialized existing Git repository in C:/Users/Conor/Documents/GitHub/git-one
-Conor-T-Foley/.git/
```

Git add

This command prepares a file to the next commit by sending it to the staging area. This staging area is where all the files that you want to include in your next commit need to be added to. Using the git status command you can check what files have and haven’t been added to the staging area.

I have added both my text documents and they are now ready to be committed



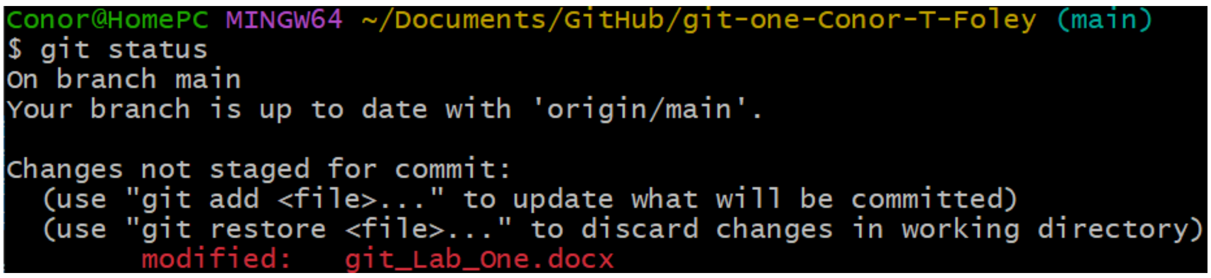
```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (main)
$ git add git_Lab_One.docx
```

git status

Gives us a readout of what files are in the staging area and are ready to be committed.

Files shown in Green have successfully been added using the git add command and are in the staging area now ready to be committed. Files shown in Red have not been added to the staging area and won’t be committed with the next commit.

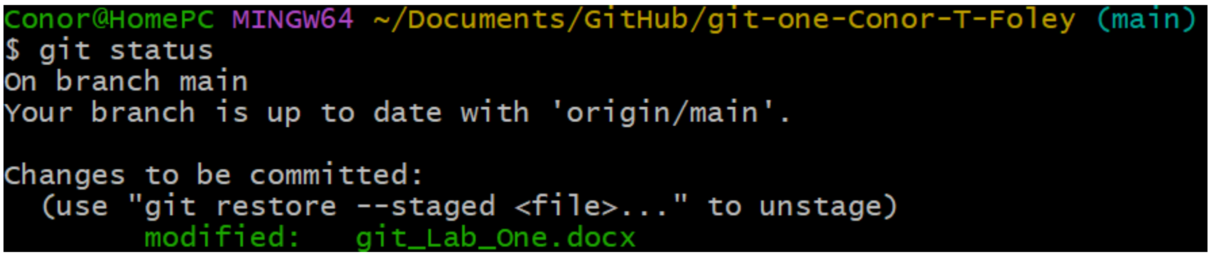
This is before adding, the file is in red



```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   git_Lab_One.docx
```

This is after using the add command to add the file to the staging area



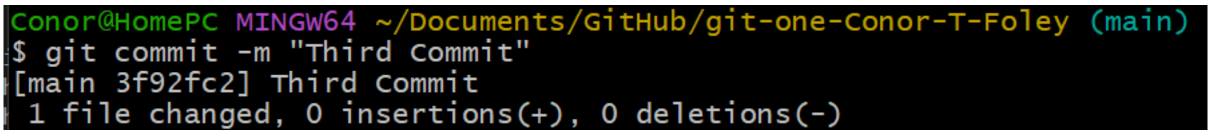
```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   git_Lab_One.docx
```

git commit

This command creates a snapshot of the staged files. In other words, a new revision point, a point along the projects timeline where the changes made between commits can easily be viewed.

Putting -m after the commit command allows me to add a comment to the commit, in this example "Third Commit" which allows me to keep track of what has been changed and at what time along the timeline



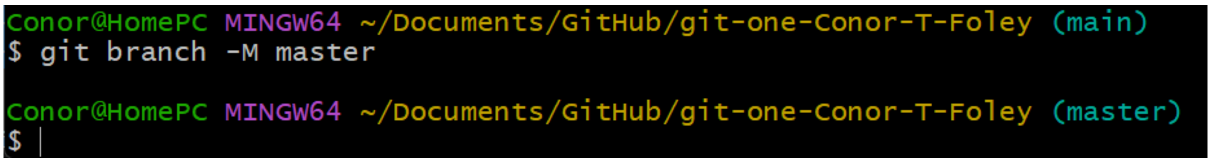
```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (main)
$ git commit -m "Third Commit"
[main 3f92fc2] Third Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
```

git branch -M master

This command has allowed me to change the name of my main branch to "master"

The git branch command is use for branch management. It can list the branches I have, create a new branch, delete a branch or rename an existing branch.

Note the branch name has switched to master at the end of the second line.



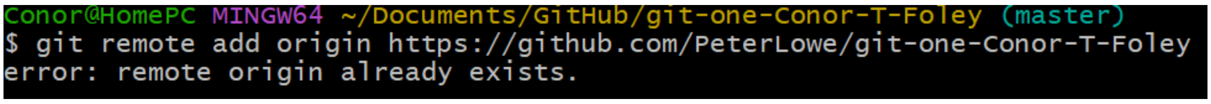
```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (main)
$ git branch -M master

Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (master)
$ |
```

git remote

This command creates a connection between my local repository that I created with git init and the remote repository created by Pete on GitHub. Note the error is because this connection has already been established at an earlier time and this screenshot is for demonstration purposes only.

The command reads as follows, git remote tells git you intend to setup a remote repository connection, add origin tells git this is where the remote repository will be and finally you add the url ofd the remote repository at the end.

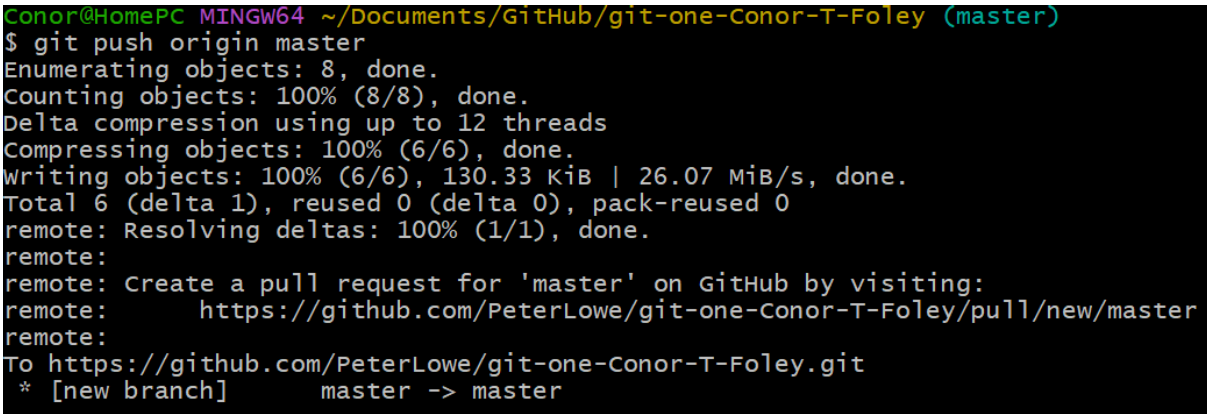


```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (master)
$ git remote add origin https://github.com/PeterLowe/git-one-Conor-T-Foley
error: remote origin already exists.
```

git push

The git push command sends content from the local repository that we created with git init to the remote repository which was created on git hub for us by Pete.

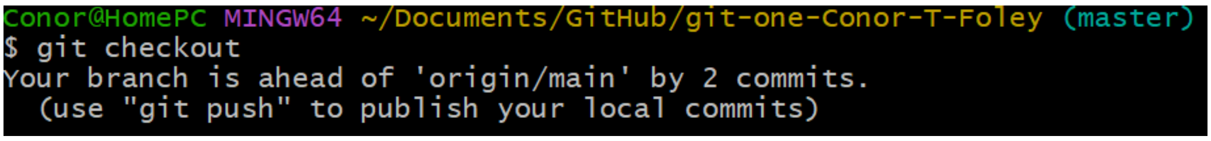
We can do this because we have established the connection to the remote repository in the last step using the git remote command.



```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (master)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 130.33 KiB | 26.07 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/PeterLowe/git-one-Conor-T-Foley/pull/new/master
remote:
To https://github.com/PeterLowe/git-one-Conor-T-Foley.git
 * [new branch]      master -> master
```

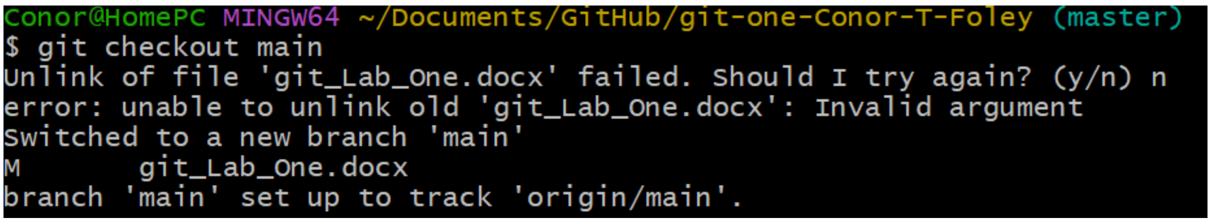
git checkout

This command allows me to see if the branch I am currently working on is up to date with the main branch. This will tell me if there are “dirty files” or files which have been edited and differ from the version of those files stored along the main branch. You can see the checkout command is telling me the branch I’m working on is 2 commits ahead of main. These are the two commits I have done from home today so far.



```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (master)
$ git checkout
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)
```

By using the command git checkout main , I have switched back to the main branch and now when I commit and push, the files will be in the correct repository.

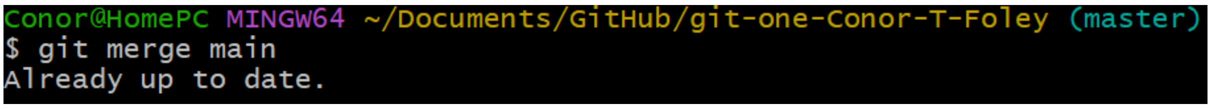


```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (master)
$ git checkout main
Unlink of file 'git_Lab_One.docx' failed. Should I try again? (y/n) n
error: unable to unlink old 'git_Lab_One.docx': Invalid argument
Switched to a new branch 'main'
M       git_Lab_One.docx
branch 'main' set up to track 'origin/main'.
```

git merge

This command allows me to merge branches to the main branch of my repository.

Since I have been working on both a main and master branch, I have now merged them together, so all files and changes appear along one timeline.

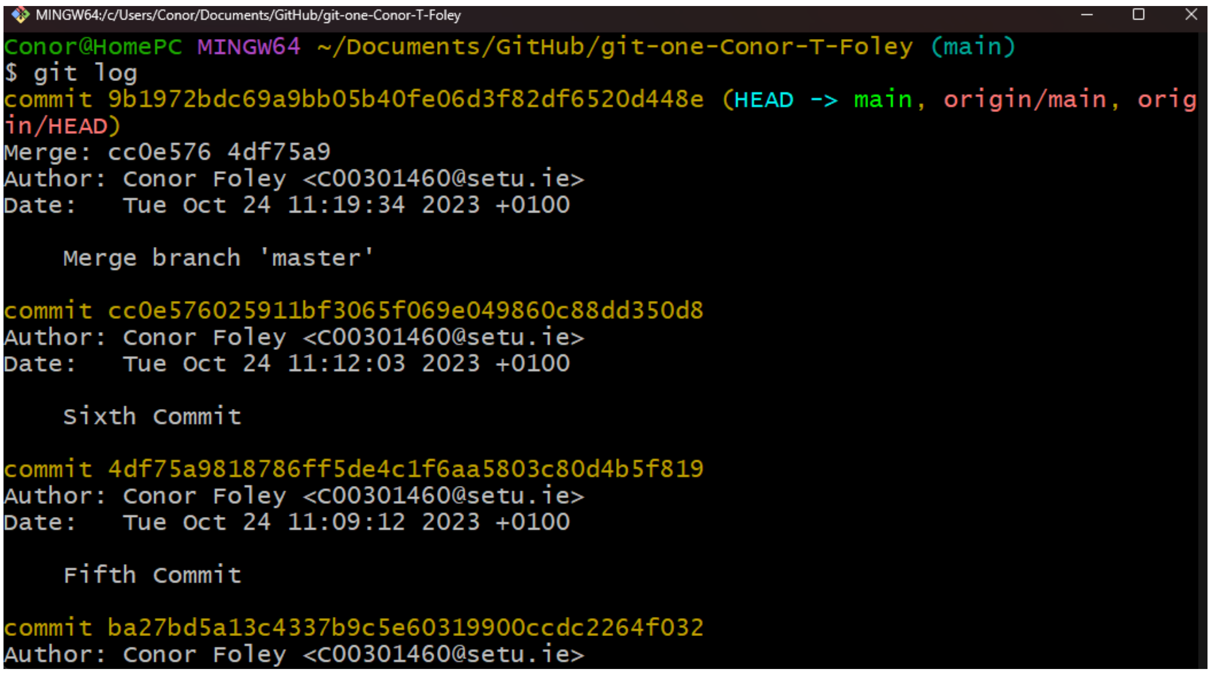


```
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (master)
$ git merge main
Already up to date.
```

git log

This command brings up a log of all activity in the current repository. It allows you to see all commits and at what date and time they were done.

You can see all the recent commits and the merge I performed between main and master.



```
MINGW64:/c/Users/Conor/Documents/GitHub/git-one-Conor-T-Foley
Conor@HomePC MINGW64 ~/Documents/GitHub/git-one-Conor-T-Foley (main)
$ git log
commit 9b1972bdc69a9bb05b40fe06d3f82df6520d448e (HEAD -> main, origin/main, orig
in/HEAD)
Merge: cc0e576 4df75a9
Author: Conor Foley <C00301460@setu.ie>
Date:   Tue Oct 24 11:19:34 2023 +0100

    Merge branch 'master'

commit cc0e576025911bf3065f069e049860c88dd350d8
Author: Conor Foley <C00301460@setu.ie>
Date:   Tue Oct 24 11:12:03 2023 +0100

    Sixth Commit

commit 4df75a9818786ff5de4c1f6aa5803c80d4b5f819
Author: Conor Foley <C00301460@setu.ie>
Date:   Tue Oct 24 11:09:12 2023 +0100

    Fifth Commit

commit ba27bd5a13c4337b9c5e60319900ccdc2264f032
Author: Conor Foley <C00301460@setu.ie>
```