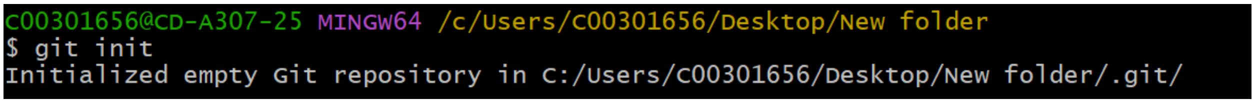
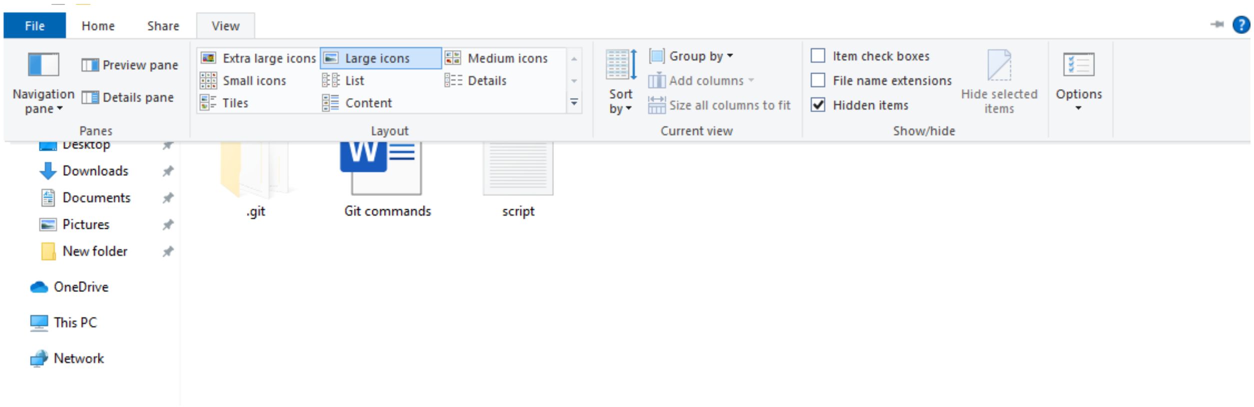
Init

Creates a new empty git repository in your selected file



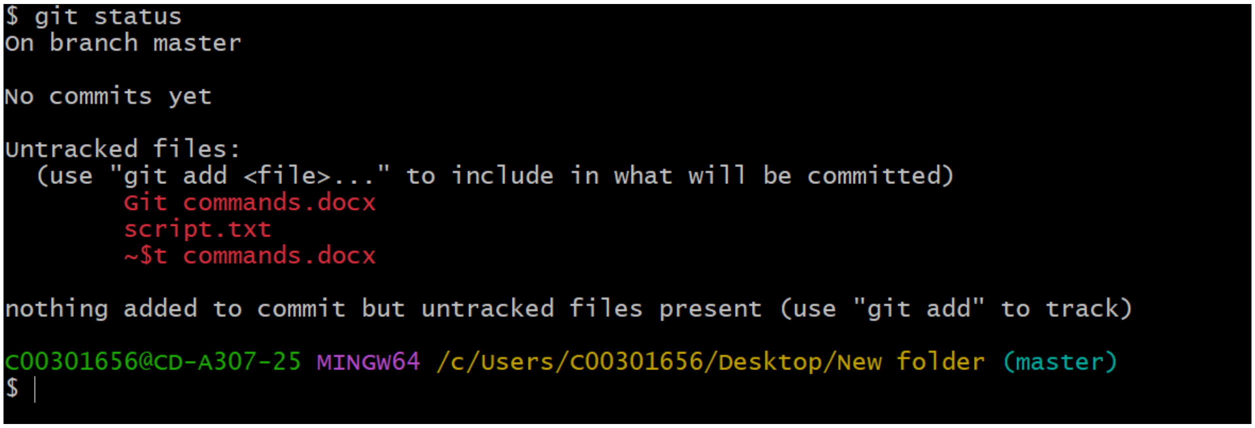
```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder
$ git init
Initialized empty Git repository in C:/Users/C00301656/Desktop/New folder/.git/
```



By selecting the hidden items in the view menu, we can then view the .git file (we shouldn't mess with the insides of the folder)

Git status

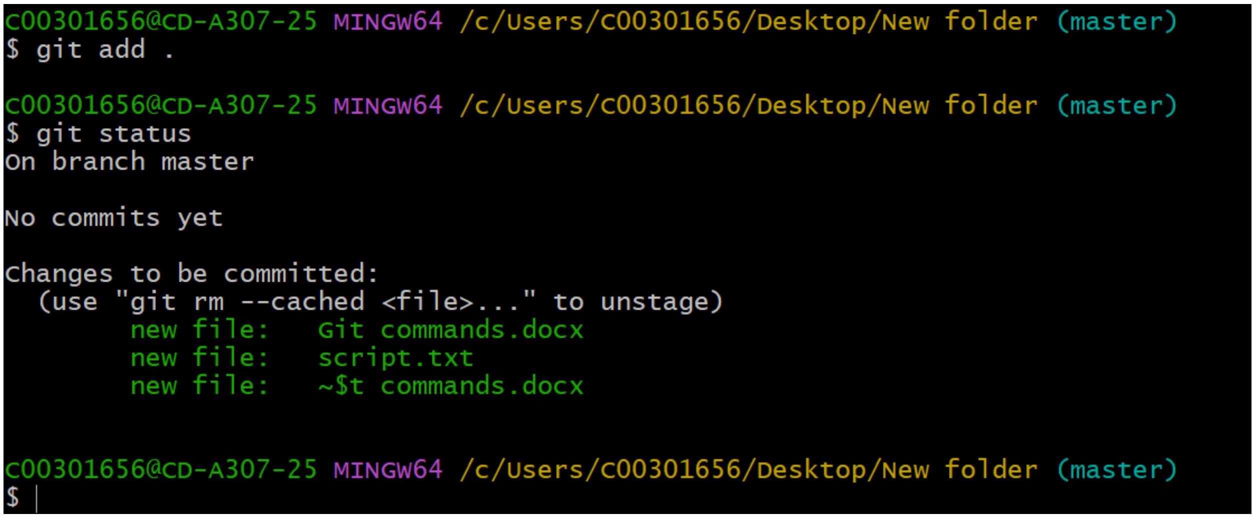
This gives us the current status of the working area (repository)
Such as if there has been a commit, what files are currently in the folder with the repository etc.
The command also shows the files in red, which as displayed in the git bash line, can be made „green" by using (git add)



```
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git commands.docx
        script.txt
        ~$t commands.docx

nothing added to commit but untracked files present (use "git add" to track)

C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ 
```

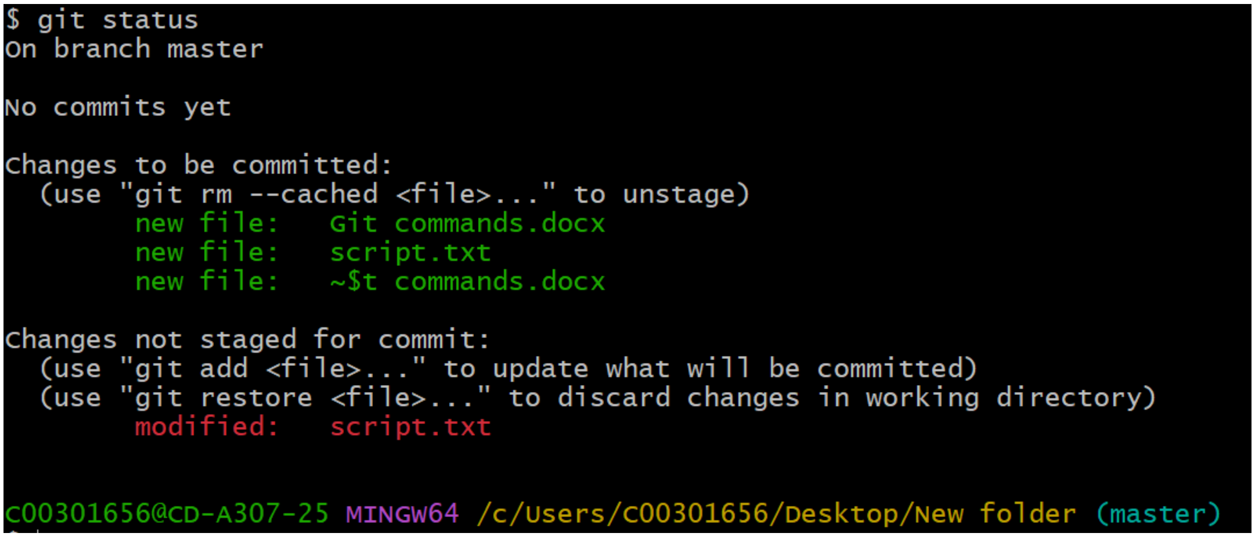
Git add

Git add on its own, doesnt seem to do anything spectacular, however once we
type the git status command again, we can see that all the files that were previously red, have turned green – meaning we can now push and commit.



```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git add .

C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Git commands.docx
        new file:   script.txt
        new file:   ~$t commands.docx


C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ |
```

If we however changed a file after adding it, the git status command will show us that the file has been modified and needs to be added again.

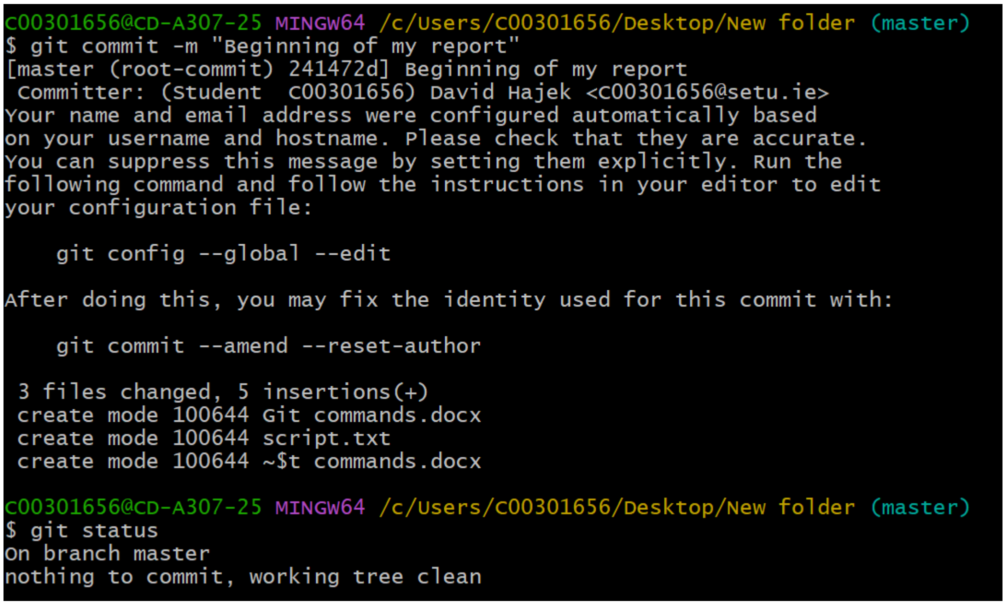


```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Git commands.docx
        new file:   script.txt
        new file:   ~$t commands.docx

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:   script.txt


C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
```

Git commit (-m „your text“)

The command will create a point of revision in your local repository (basically a save)
you commit to the work you have done and (save) it.

The -m „your text“ basically gives the save a name, giving us a easy to acess „Point in time“ to come back to, if we didn’t name them, we would have to guess and check what the commit did at the time.



```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git commit -m "Beginning of my report"
[master (root-commit) 241472d] Beginning of my report
 Committer: (Student  C00301656) David Hajek <C00301656@setu.ie>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 3 files changed, 5 insertions(+)
 create mode 100644 Git commands.docx
 create mode 100644 script.txt
 create mode 100644 ~$t commands.docx

C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git status
On branch master
nothing to commit, working tree clean
```

We can also see the commit history online by clicking the commit buttons with a clock icon

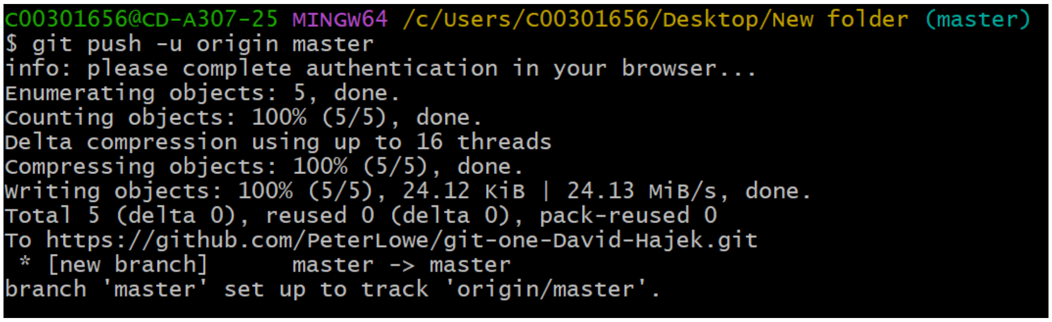
Git push -u origin (name of the defaut branch**, master in this case**)

The name of the file can be found in the cyan color after the colorful string of text

Git push updates the remote node with our current node that we are working on

Git push -u origin (name)

Pushes the local master to the on server origin (online)



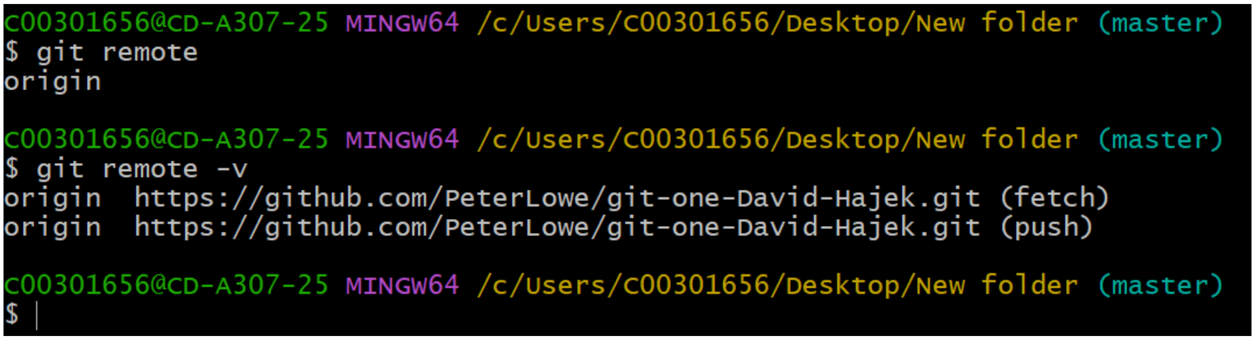
```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 24.12 KiB | 24.13 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PeterLowe/git-one-David-Hajek.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Git remote (-v)

Git remote lists all the repositorys on the remote server

Git remote add (name) (url)
Creates a new remote repository

Git remote rm (name)
Delets a remote repository (online)

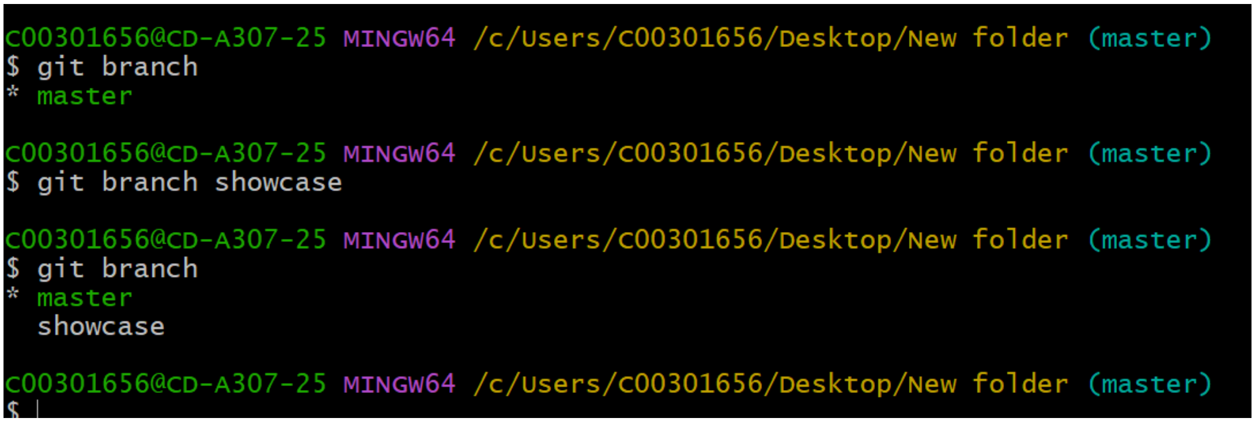


```
c00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git remote
origin

c00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git remote -v
origin  https://github.com/PeterLowe/git-one-David-Hajek.git (fetch)
origin  https://github.com/PeterLowe/git-one-David-Hajek.git (push)

c00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ |
```

Git branch (branch_name) – Creates a new branch in the local repository

Git branch shows all the current branches in the working repository

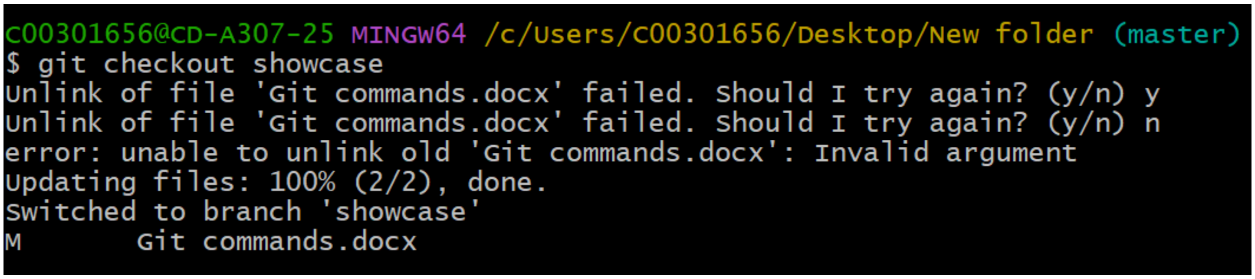
Git branch -m (branch_name) – Deletes a branch



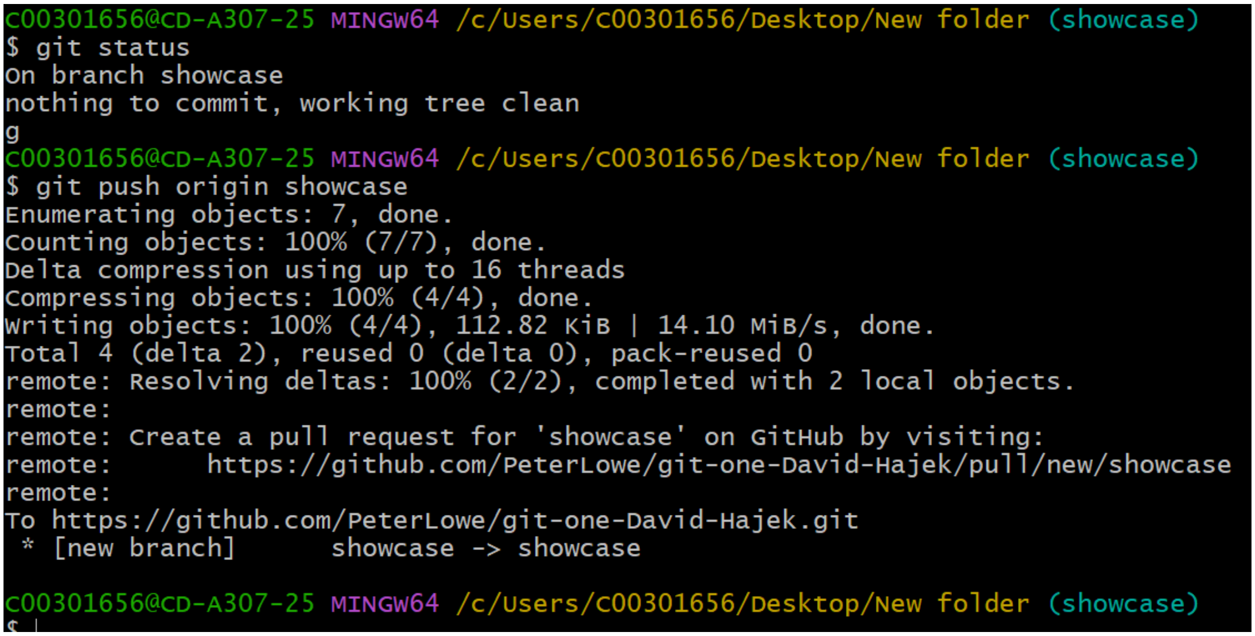
```
c00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git branch
* master

c00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git branch showcase

c00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git branch
* master
  showcase

c00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ |
```

Git checkout
Switches the current working directory (which was master) to a different branch (in this case the branch called showcase I created earlier)

The command can be repeated to go back to the master branch again.



```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git checkout showcase
Unlink of file 'Git commands.docx' failed. Should I try again? (y/n) y
Unlink of file 'Git commands.docx' failed. Should I try again? (y/n) n
error: unable to unlink old 'Git commands.docx': Invalid argument
Updating files: 100% (2/2), done.
Switched to branch 'showcase'
M       Git commands.docx
```



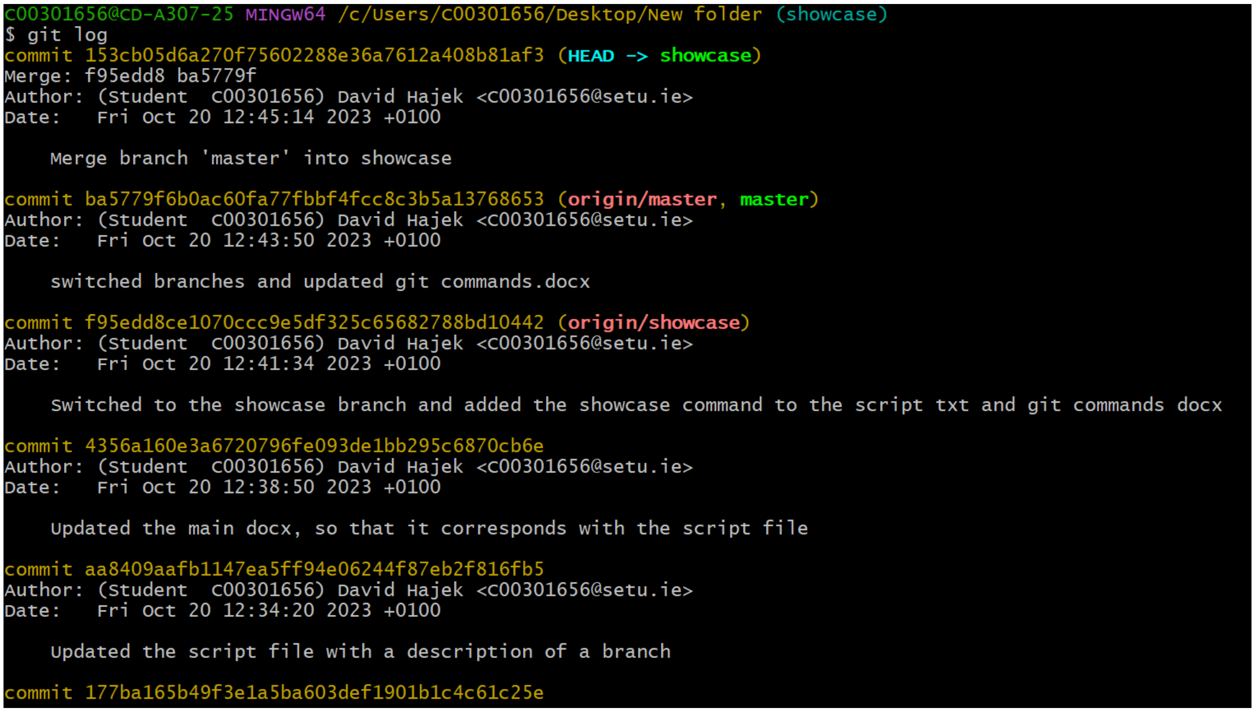
```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (showcase)
$ git status
On branch showcase
nothing to commit, working tree clean
g
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (showcase)
$ git push origin showcase
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 112.82 KiB | 14.10 MiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'showcase' on GitHub by visiting:
remote:      https://github.com/PeterLowe/git-one-David-Hajek/pull/new/showcase
remote:
To https://github.com/PeterLowe/git-one-David-Hajek.git
 * [new branch]      showcase -> showcase

C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (showcase)
$
```

Git log

Showcases our entire commit history – can be navigated with (space) to move pages and with (q) to quit the log

Git log –graph will show the branches paths

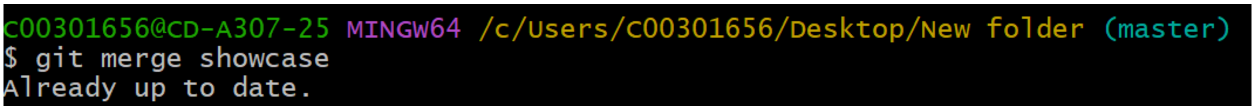
Git log –stat shows files that have been altered and changed



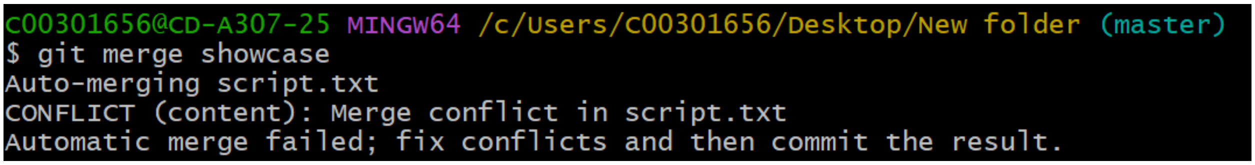
```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (showcase)
$ git log
commit 153cb05d6a270f75602288e36a7612a408b81af3 (HEAD -> showcase)
Merge: f95edd8 ba5779f
Author: (Student  C00301656) David Hajek <C00301656@setu.ie>
Date:   Fri Oct 20 12:45:14 2023 +0100

    Merge branch 'master' into showcase

commit ba5779f6b0ac60fa77fbbf4fcc8c3b5a13768653 (origin/master, master)
Author: (Student  C00301656) David Hajek <C00301656@setu.ie>
Date:   Fri Oct 20 12:43:50 2023 +0100

    switched branches and updated git commands.docx

commit f95edd8ce1070ccc9e5df325c65682788bd10442 (origin/showcase)
Author: (Student  C00301656) David Hajek <C00301656@setu.ie>
Date:   Fri Oct 20 12:41:34 2023 +0100

    Switched to the showcase branch and added the showcase command to the script txt and git commands docx

commit 4356a160e3a6720796fe093de1bb295c6870cb6e
Author: (Student  C00301656) David Hajek <C00301656@setu.ie>
Date:   Fri Oct 20 12:38:50 2023 +0100

    Updated the main docx, so that it corresponds with the script file

commit aa8409aafb1147ea5ff94e06244f87eb2f816fb5
Author: (Student  C00301656) David Hajek <C00301656@setu.ie>
Date:   Fri Oct 20 12:34:20 2023 +0100

    Updated the script file with a description of a branch

commit 177ba165b49f3e1a5ba603def1901b1c4c61c25e
```

Git merge (branch name)

Merges the main repository with the branch that you want to merge it with



```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git merge showcase
Already up to date.
```



```
C00301656@CD-A307-25 MINGW64 /c/Users/C00301656/Desktop/New folder (master)
$ git merge showcase
Auto-merging script.txt
CONFLICT (content): Merge conflict in script.txt
Automatic merge failed; fix conflicts and then commit the result.
```

It may appear as an error, but it can be fixed with updating the files.