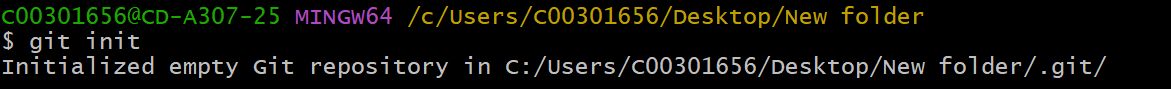
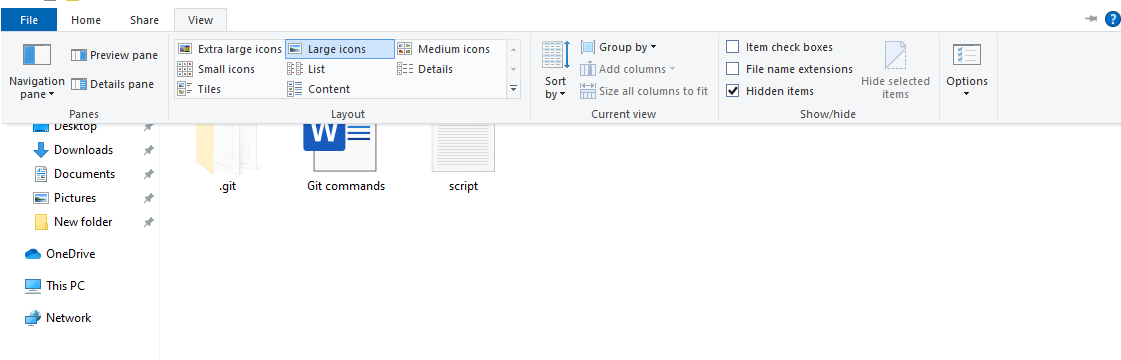
Init

Creates a new empty git repository in your selected file

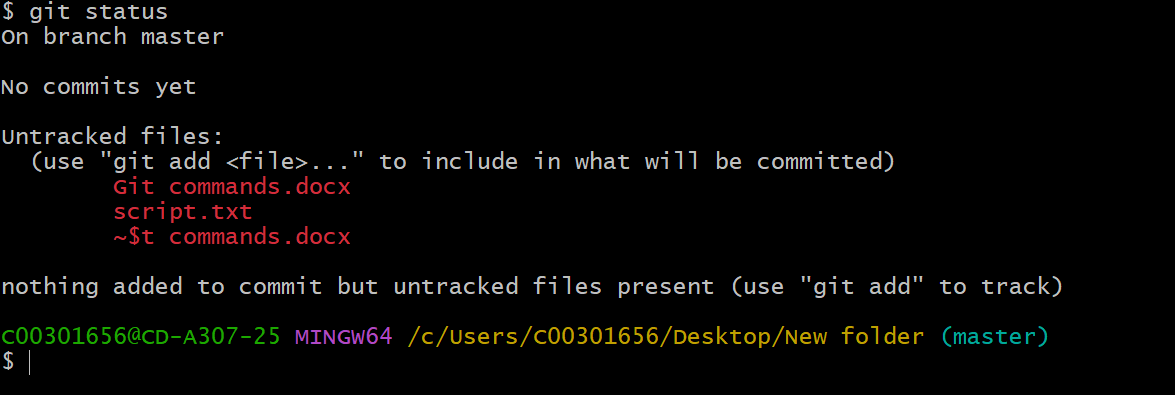




By selecting the hidden items in the view menu, we can then view the .git file (we shouldn’t mess with the insides of the folder)

**Git status**

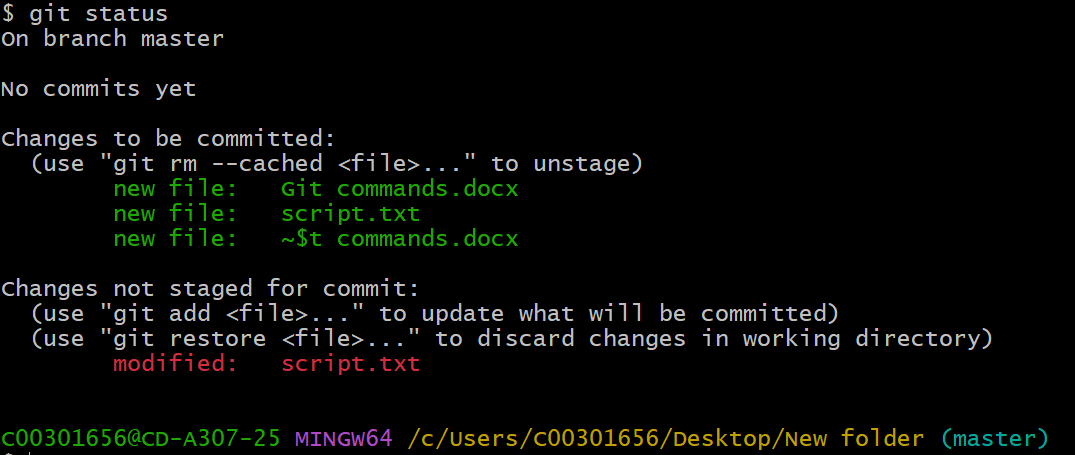
This gives us the current status of the working area (repository)  
Such as if there has been a commit, what files are currently in the folder with the repository etc.  
The command also shows the files in red, which as displayed in the git bash line, can be made „green“ by using (git add)



**Git add**

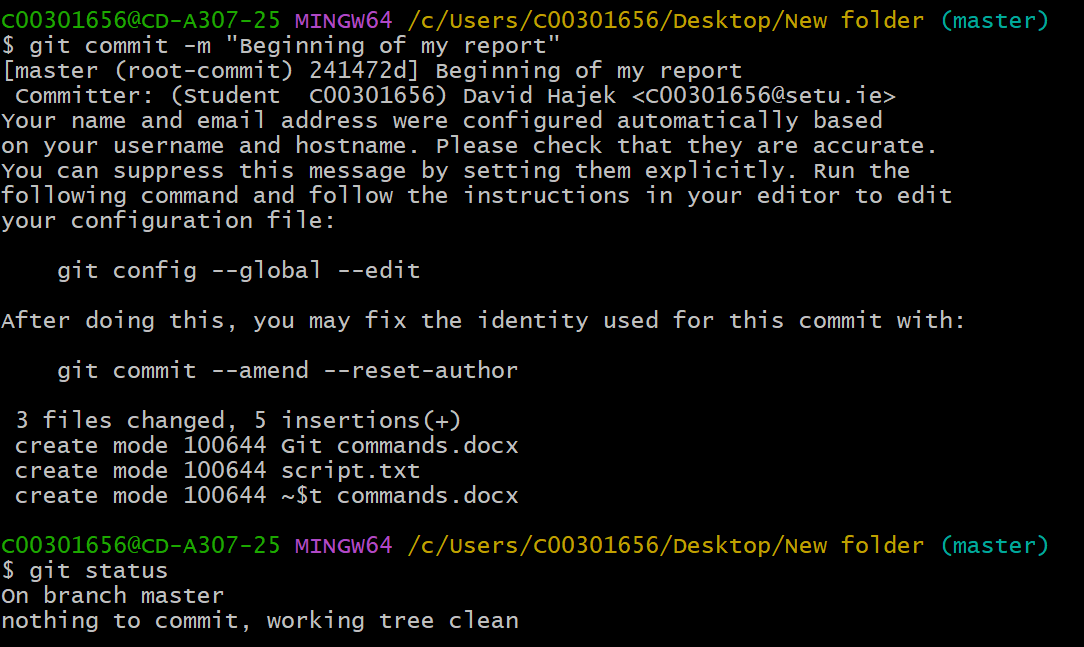
Git add on its own, doesnt seem to do anything spectacular, however once we   
type the git status command again, we can see that all the files that were previously red, have turned green – meaning we can now push and commit.



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 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.

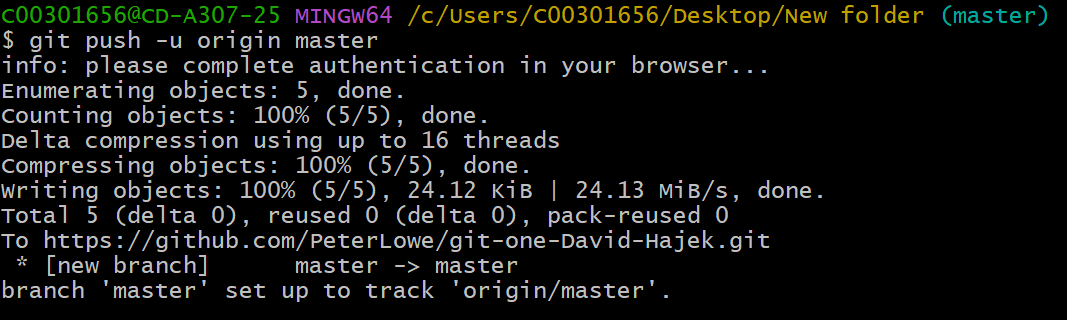
We can also see the commit history online by clicking the commit buttons with a clock icon

**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)

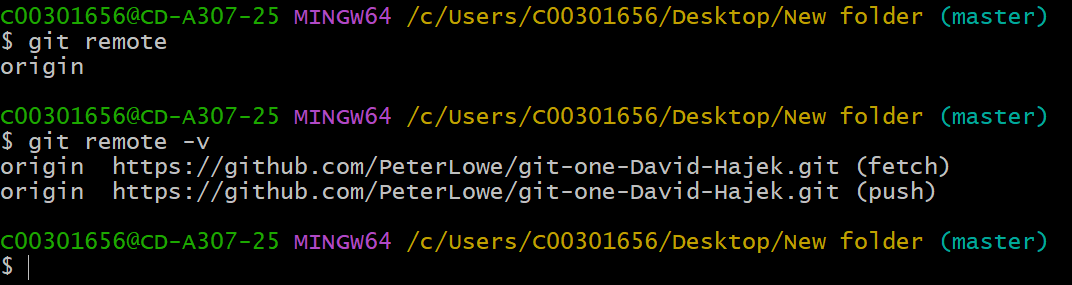
Pushes the local master to the on server origin (online)

**Git remote (-v)**

Git remote lists all the repositorys on the remote server

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

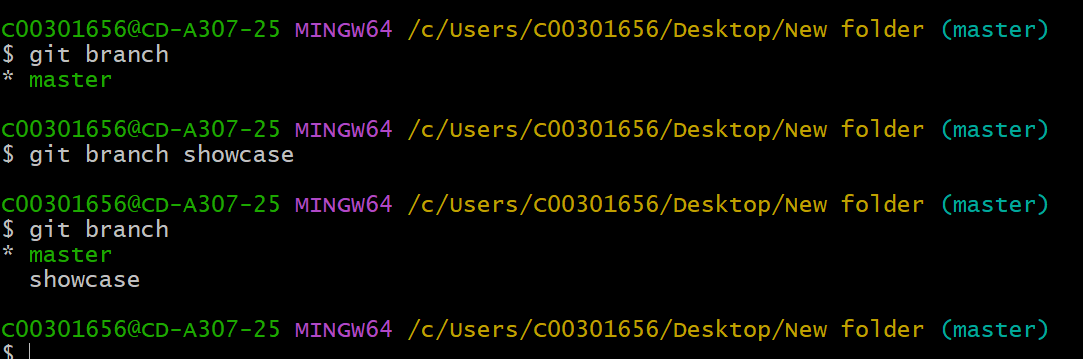
Git remote rm (name)  
Delets a remote repository (online)



**Git branch (branch\_name)** – Creates a new branch in the local repository

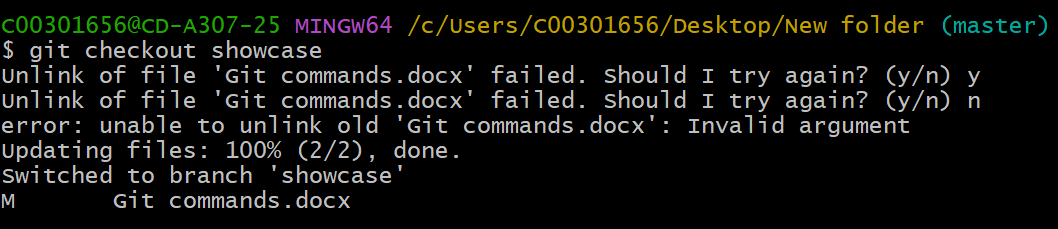
Git branch shows all the current branches in the working repository

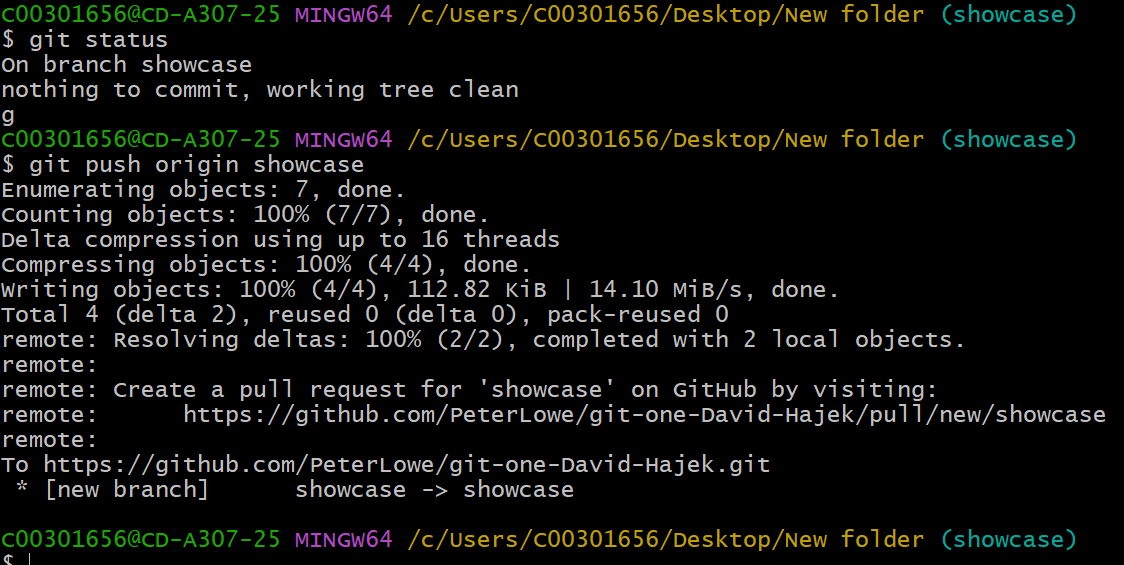
Git branch -m (branch\_name) – Deletes a branch



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.



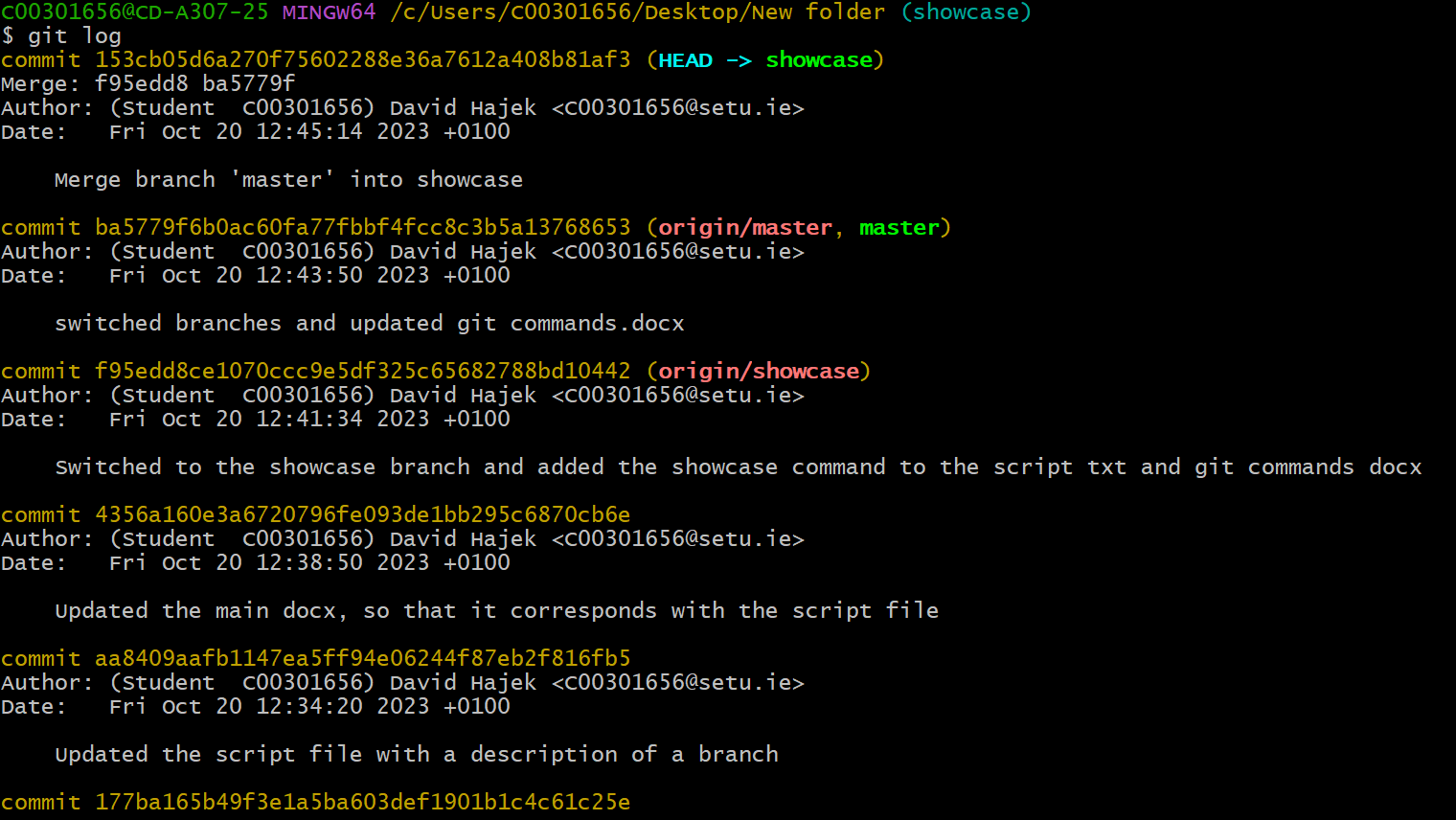


**Git log**

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

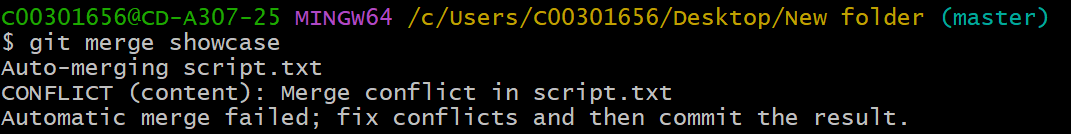
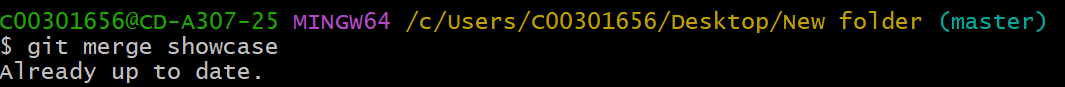
Git log –graph will show the branches paths

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



Git merge (branch name)

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



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