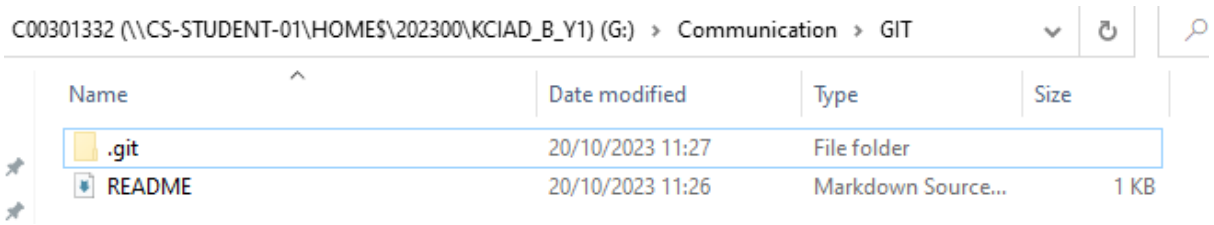
GIT

Init

The init code command creates empty repository in my student folder

```
C00301332@CD-A307-24 MINGW64 /g/Communication/GIT
$ git init
Initialized empty Git repository in //CS-STUDENT-01/HOME$/202300/KCIAD_B_Y1/C00301332/Communication/GIT/.git/
```

We can check this by going the address of the repository and showing hidden files (.git)



Add

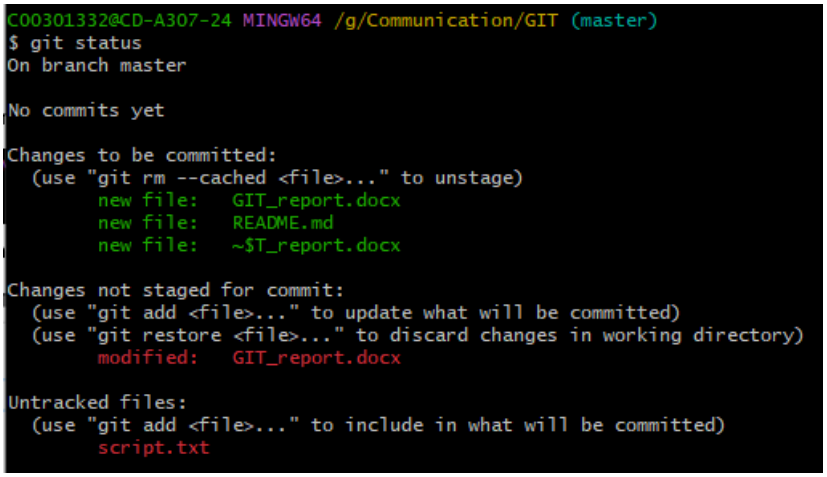
The add code command will add . all the files in the folder, we can check this with status command

The git add . adds all files, if we want to specify we use git add <name of the file>

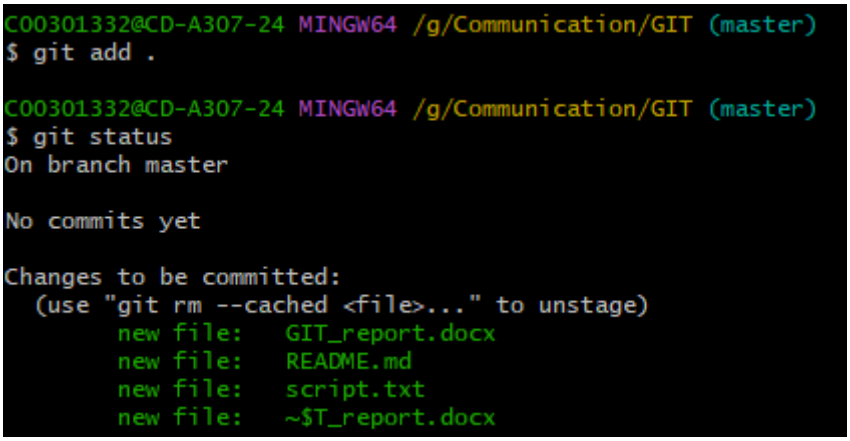
For example:

```
C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
$ git add README.md

C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   GIT_report.docx
        new file:   README.md
        new file:   script.txt
        new file:   ~$T_report.docx
```

Before using the git add . command:

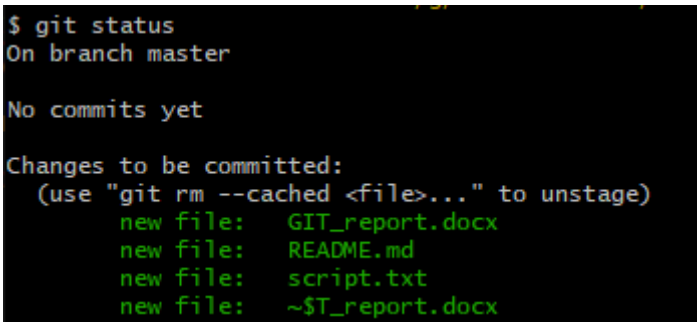


After using the git add . command:



Status

I used status command when showing the add command, but what it does is that it shows the status of the folder in which we are using the git bash



Commit

The commit command sets “milestones” in the project timeline, something we can come back in later

There is Committer: in this instance me

We can see that the commit name is first_commit, then we have info about committer and we can see that all the files in the folder were changed

```
C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
$ git commit -m "first_commit"
[master (root-commit) 05cdee4] first_commit
 Committer: (Student  C00301332) Pavel Dobias <C00301332@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

 4 files changed, 6 insertions(+)
 create mode 100644 GIT_report.docx
 create mode 100644 README.md
 create mode 100644 script.txt
 create mode 100644 ~$T_report.docx

C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
```

Remote

The git remote add <name> <url> is used to add remote server to the folder we are using the git bash in

We can check what remote servers we have by using git remote -v

```
C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
$ git remote add origin https://github.com/PeterLowe/git-one-PavelDobias2002.git

C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
$ git remote -v
origin  https://github.com/PeterLowe/git-one-PavelDobias2002.git (fetch)
origin  https://github.com/PeterLowe/git-one-PavelDobias2002.git (push)
```

Push

Git push updates remote node with current node that we are using, we are pushing local nodes to the remote ones. We push the local to remote.

To push we need to complete authentication in browser and needs to be associated remote node

In this instance its: origin main which we created before using git remote add

```
C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
$ git branch -M main

C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 28.14 KiB | 5.63 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PeterLowe/git-one-PavelDobias2002.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Branch

We use git branch name_of_the_branch to create sequence of commit points which needs to be merged to main

```
CO0301332@CD-A307-24 MINGW64 /g/Communication/GIT (master)
$ git branch -M main
```

```
On branch main
Your branch is up to date with 'origin/main'.
```

We can check all our branches using git branch -a

```
CO0301332@CD-A307-24 MINGW64 /g/Communication/GIT (main)
$ git branch -a
* main
  seperate
  remotes/origin/main
```

Checkout

We use the git checkout branch_name to switch to that branch from current one

I am already on the branch main

```
CO0301332@CD-A307-24 MINGW64 /g/Communication/GIT (main)
$ git checkout main
Already on 'main'
M       script.txt
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
```

I created new branch separate and switched to it using git checkout

```
CO0301332@CD-A307-24 MINGW64 /g/Communication/GIT (main)
$ git branch seperate

CO0301332@CD-A307-24 MINGW64 /g/Communication/GIT (main)
$ git checkout seperate
Switched to branch 'seperate'
M       script.txt
```

Log

Git log displays commit history

Here we can see who did the commit, what was committed and the name of the comit

```
C00301332@CD-A307-24 MINGW64 /g/Communication/GIT (main)
$ git log
commit 7b3739be6fb9cf4b0e312fbcbcba632c7b4585e7 (HEAD -> main, origin/main)
Author: (Student  C00301332) Pavel Dobias <C00301332@setu.ie>
Date:   Fri Oct 20 12:30:03 2023 +0100

    third_commit

commit 2eefd3a6920766a5f7c61ea433e0e3cb9e2cdaa9 (seperate)
Author: (Student  C00301332) Pavel Dobias <C00301332@setu.ie>
Date:   Fri Oct 20 12:24:13 2023 +0100

    second_commit

commit 9a68f5c7ac6a76582b9bdcd5c3a9b8563630065c
Author: (Student  C00301332) Pavel Dobias <C00301332@setu.ie>
Date:   Fri Oct 20 11:51:28 2023 +0100

    first_commit
```

Clone

Git clone copies existing repository, creating new copy of it in different folder

to clone we open different folder in it we open git bash and we use the url of the reposiotry we want to copy to that folder

```
C00301332@CD-A307-24 MINGW64 /c/Users/C00301332/Desktop/New folder
$ git clone https://github.com/PeterLowe/git-one-PavelDobias2002.git
Cloning into 'git-one-PavelDobias2002'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 13 (delta 4), reused 11 (delta 2), pack-reused 0
Receiving objects: 100% (13/13), 112.72 KiB | 1.22 MiB/s, done.

Resolving deltas: 100% (4/4), done.
```