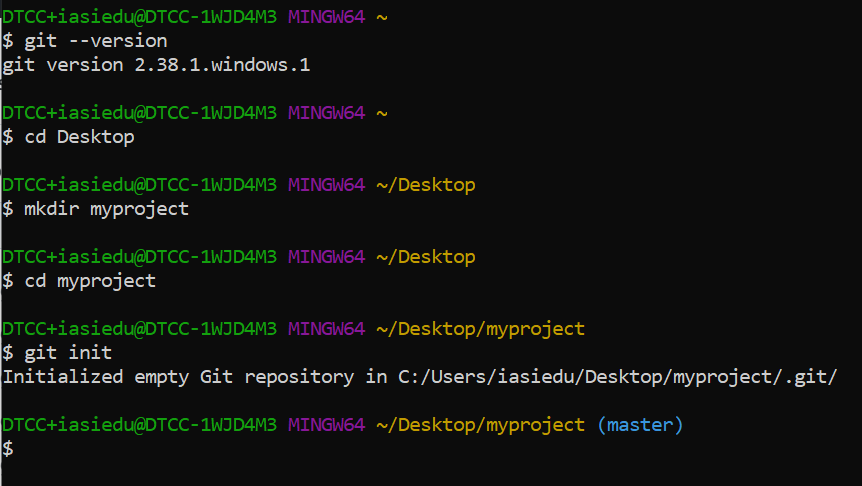
Notes:

* Get the version of git.
* Changed current directory to Desktop.
* Created a new directory(mkdir) ‘myproject’.
* Changed directory into ‘myproject’.
* Created git repository(git init).



* List files in the directory

Text

Description automatically generated

* Create html file in sublime text3 and save it as index.html

Text

Description automatically generated

* List files
* Check git status and see if it is part of the repo.
* Git is aware of the file but has not yet added it to the repo.
* Notes.doc is this word document saved in the repo. AS I update it, different versions will be created.

Text

Description automatically generated

Git staging environment.

Staged files are ready to be committed to the repo.

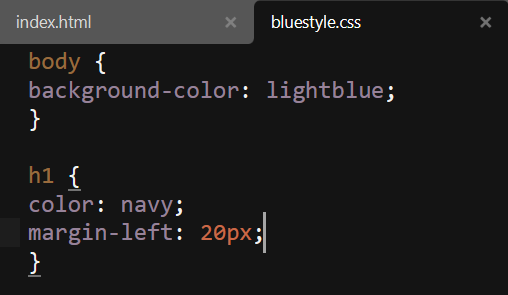
* Git add to add the html file to the staging area.

Text

Description automatically generated

Add more Files.

* A README.md file to describe the repo.(use a txt doc)
* This word document.
* A css file
* An updated html file.

 Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Git Commit

Adding commits keeps tracks of your progress. Git considers each “commit” as a save point. It is a point in the project where you can go back to if you find a bug or want to make a change. When we commit we should use a message to tell everyone what changes have been made.

The commit command performs a commit, and the -m “message” adds a message.

Text

Description automatically generated

Commit Without Stage

It is possible to commit changes directory without staging. The -a option will stage every changed and tracked file.

Let’s update the index.html and this document which is always been updated.

Use –short to check status, then commit.