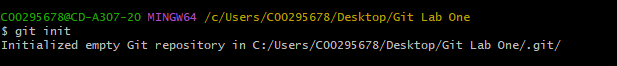
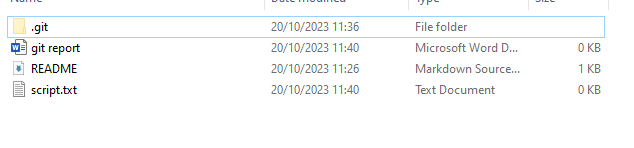
# Init

Git init is used to initialise a git repository



It creates the repository in the folder directory GitBash was opened in

It also creates a new folder as a hidden item

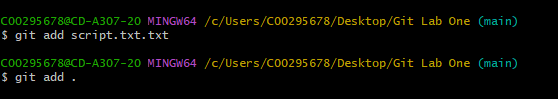


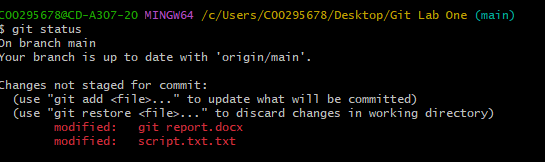
This .git folder indicates that this is the folder where the repository was created.

# Add

Git add sends files to the “staging area”. In this area, they are prepared to be committed. Add should be used anytime a file has been updated by the user and hasn’t been updated in the repository. This includes any new files introduced, and any previous files that have been updated.

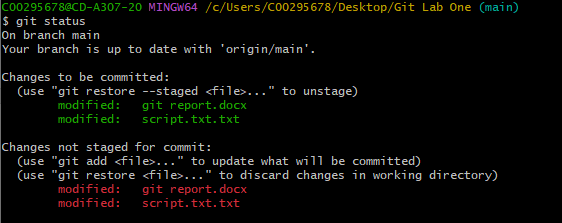
The user can specify any file within their folder by typing the filename after “git add”, and they can add all files in the folder by using a “.”





# Status

Git status is used to display the current status of files in the source folder and the staging area. It will display what files are yet to be added and which files are yet to be committed. A red file is one that hasn’t been added to the staging area yet, while green files are files that exist in the staging area, but are yet to be committed. Git status should be used following any add or commit done by the user to verify the process has been completed correctly.



# Commit

Git commit is used to fully introduce a file from the staging area into the repository. As stated previously, a git status will show uncommitted files as green. Committed files won’t appear when a git status is called, a signal that the commit was successful. Unliked an add, however, a commit must come with a message. The line must read “git commit -m “<commit title>”. This means that when the commit is viewed in the repository, there is a title included. If this is not done, git will open up VI, a program that is much harder to use than a “-m” function.

