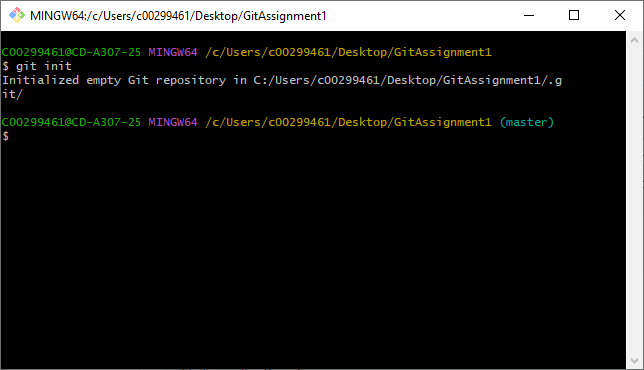
# Init

git init is used to create a new repository in the place where you started your Git Bash from. It creates a hidden .git folder which is the directory for this project. It will also create an empty branch for you to work in. It will mainly be used at the beginning of a project, but you can use it in an existing repository without issue.

Syntax: git init



# Add

git add is used to bring a file found in your bash folder to the ‘stage,’ making it ready for the next commit. This acts as a ‘screenshot’ of the file, so if you update it before committing you will need to use this command to stage it again. You can check if it worked with the git status command, after which the file will be highlighted in green if it’s staged.

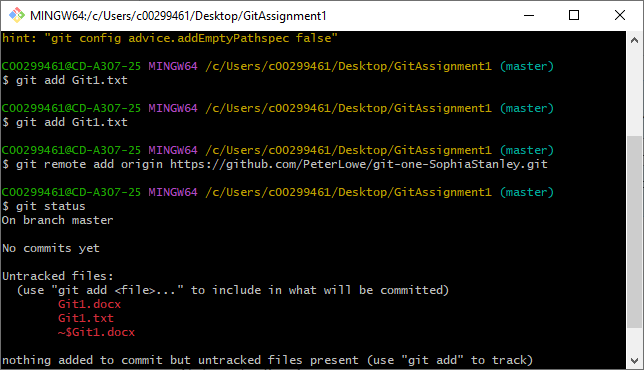
Syntax: git add

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated



# Status

git status is used to show the details of your repository. This includes what branch you’re currently on and details on your files, for example if they’ve been modified but not staged. If you have a remote repository, it will also tell you how much this version of the repository differs between the remote one.

Syntax: git status

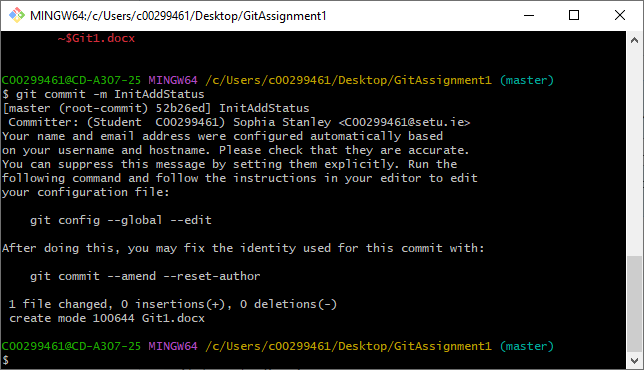
A screenshot of a computer program

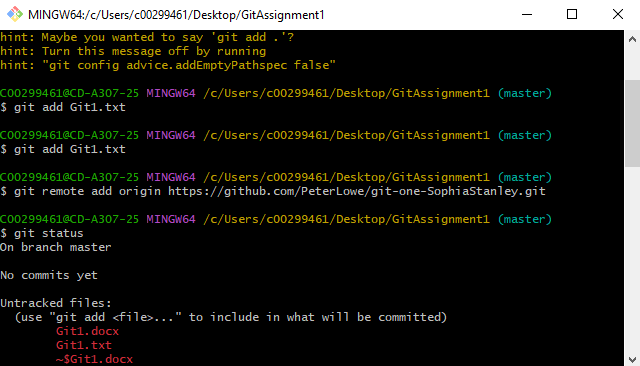
Description automatically generated

# Commit

git commit is used to create a save point for your repository. If adding is a screenshot of your files, then committing could be seen as a screenshot of your whole repository. You cannot commit until you’ve made some changes since the previous time, such as by staging another file.

Syntax: git commit





# Push

The git push command updates any remote repositories with the most recent commit to ensure that they’re both up to date, by sending all of the new content within the local copy to the remote one. You should specify which repository you want to update, as if you have more than one remote it will either be sent to the one configured as the ‘default’ or the one named origin.

Syntax: git push [repository]

A screenshot of a computer program

Description automatically generated

# Remote

You can use git remote to create additional versions of your repository that exist outside of your local drive. For creating a new one, you will need to include a name and the URL of where the new copy will be stored. Simply typing ‘git remote’ brings up a list of your remote copies, and adding -v gives a more detailed list.

Syntax: git remote add [name] [URL]

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Log

git log displays a list of every commit you have made in this repository, starting from the earliest. Depending on your screen size, this can fill up the whole screen after about 5 commits, after which you can press the space bar to scroll down to the end, and q to quit out of the list.

Syntax: git log

A screenshot of a computer program

Description automatically generated

# Stash

git stash is used for when you want to create a record of your current directory, without needing to commit. You can use git stash pop to retrieve the contents from a stash to put in the repository.

Syntax: git stash

A screenshot of a computer program

Description automatically generated

# Branch

git branch can be used to create another branch in a repository. This is a great way of sectioning off certain parts to do different things rather than having everything combined into one branch.

Syntax: git branch [name]

A screenshot of a computer

Description automatically generated

# Checkout

If you’re working with multiple branches, you can use the git checkout command to specify which one you want to be on currently. Make sure your current branch has been fully committed or stashed, otherwise it will not allow you to run the command.

Syntax: git checkout [branch]

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated