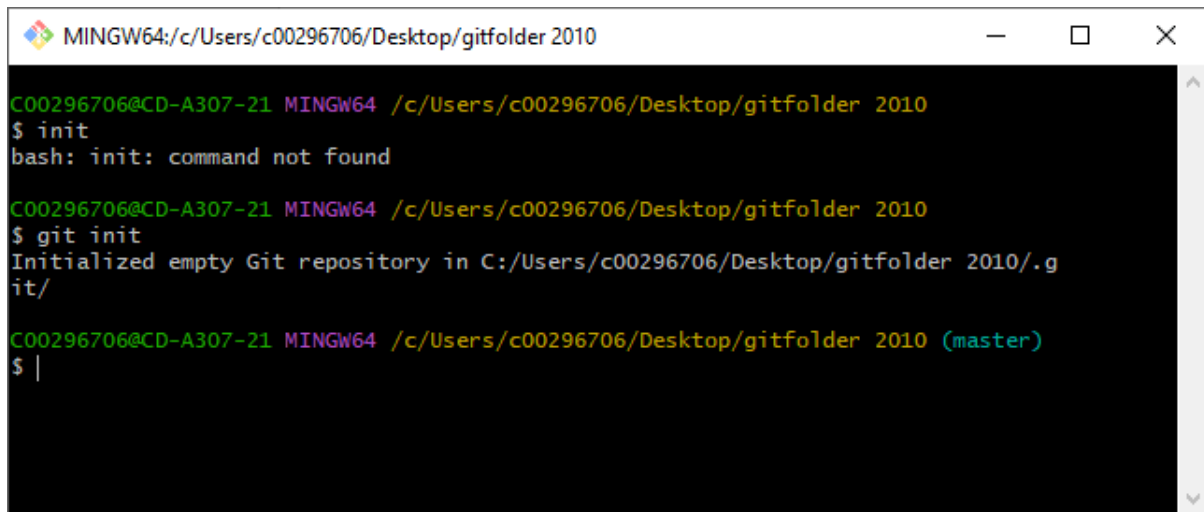


Init

git init creates a new repository in the folder that Git Bash is launched in



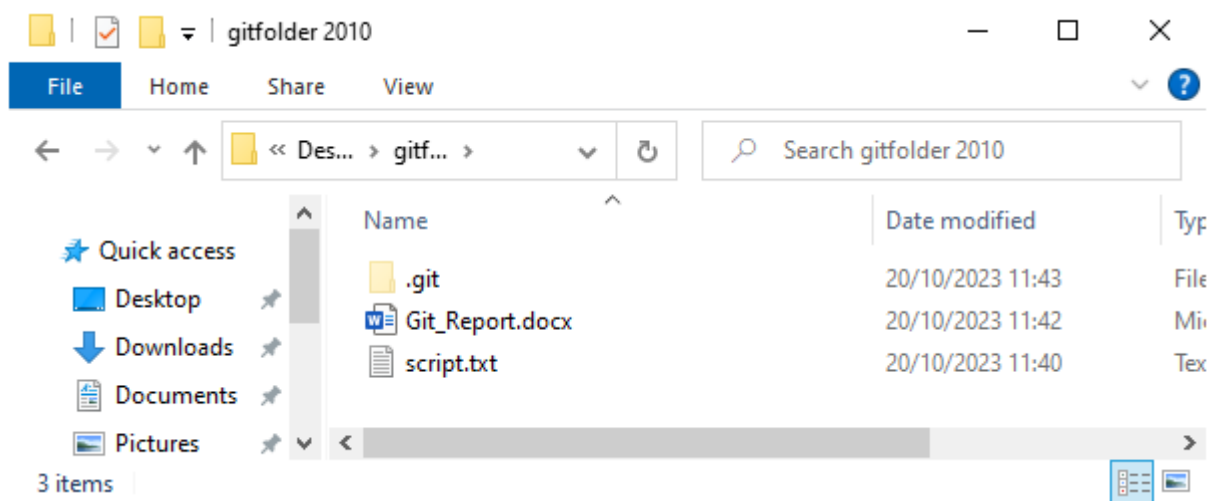
```
MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010

C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010
$ init
bash: init: command not found

C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010
$ git init
Initialized empty Git repository in C:/Users/c00296706/Desktop/gitfolder 2010/.git/

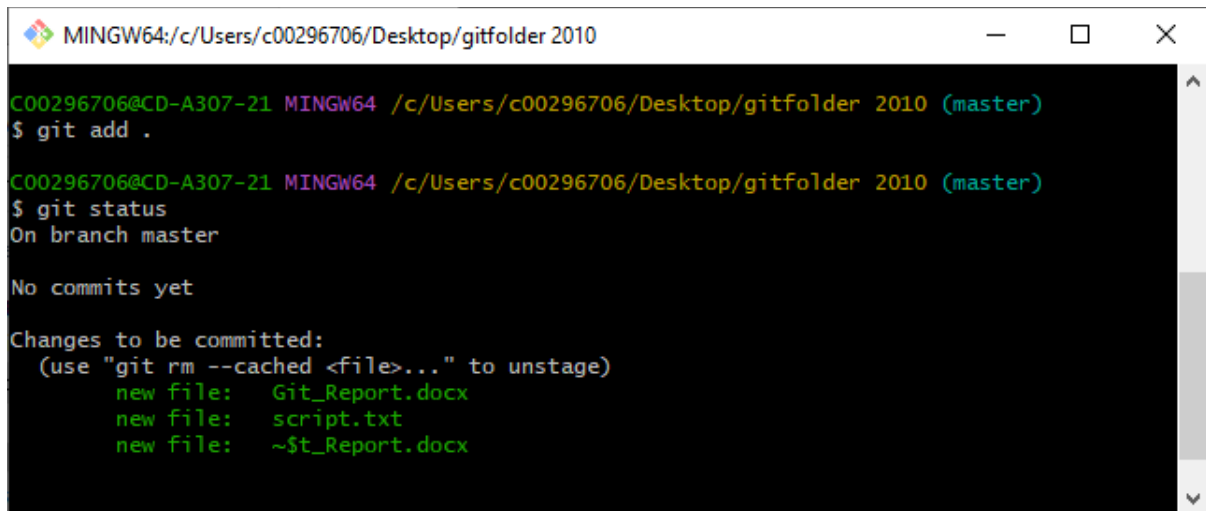
C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ |
```

A hidden .git folder is created that contains this repository



Add

git add adds files to the directory, I will use git add . to add all the new files, we can see that all the files are added using the “git status” command that I will go over next

A screenshot of a MINGW64 terminal window. The title bar shows the path 'MINGW64:/c:/Users/c00296706/Desktop/gitfolder 2010'. The terminal has a black background with green and white text. It shows the execution of 'git add .' followed by 'git status'. The status output indicates that three new files are staged for commit: 'Git_Report.docx', 'script.txt', and '~\$t_Report.docx'.

```
MINGW64:/c:/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git add .

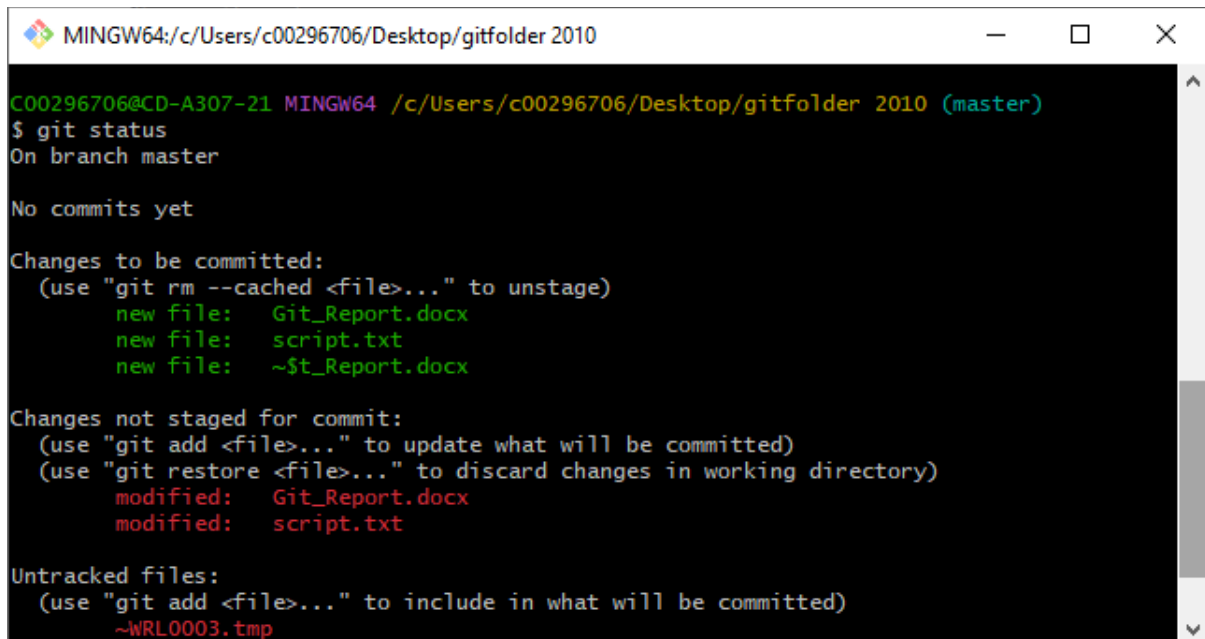
MINGW64:/c:/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git status
On branch master

No commits yet

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

Status

"git status" will list all the files in the directory, modified files will be red and added files will be green



```
MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010

C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git status
On branch master

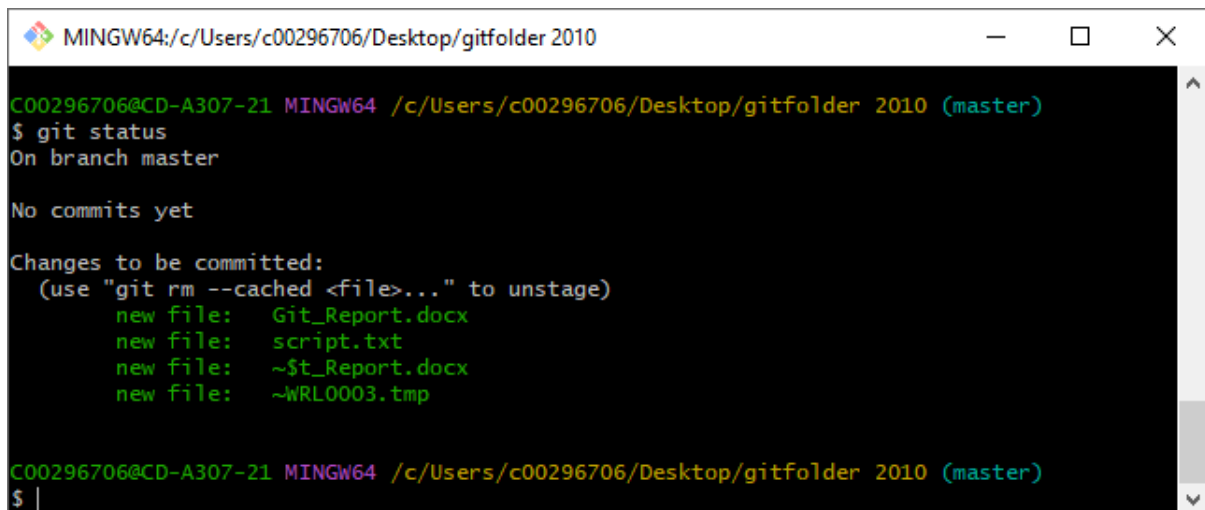
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Git_Report.docx
        new file:   script.txt
        new file:   ~$t_Report.docx

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Git_Report.docx
        modified:   script.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ~WRL0003.tmp
```

After using git add .



```
MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010

C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git status
On branch master

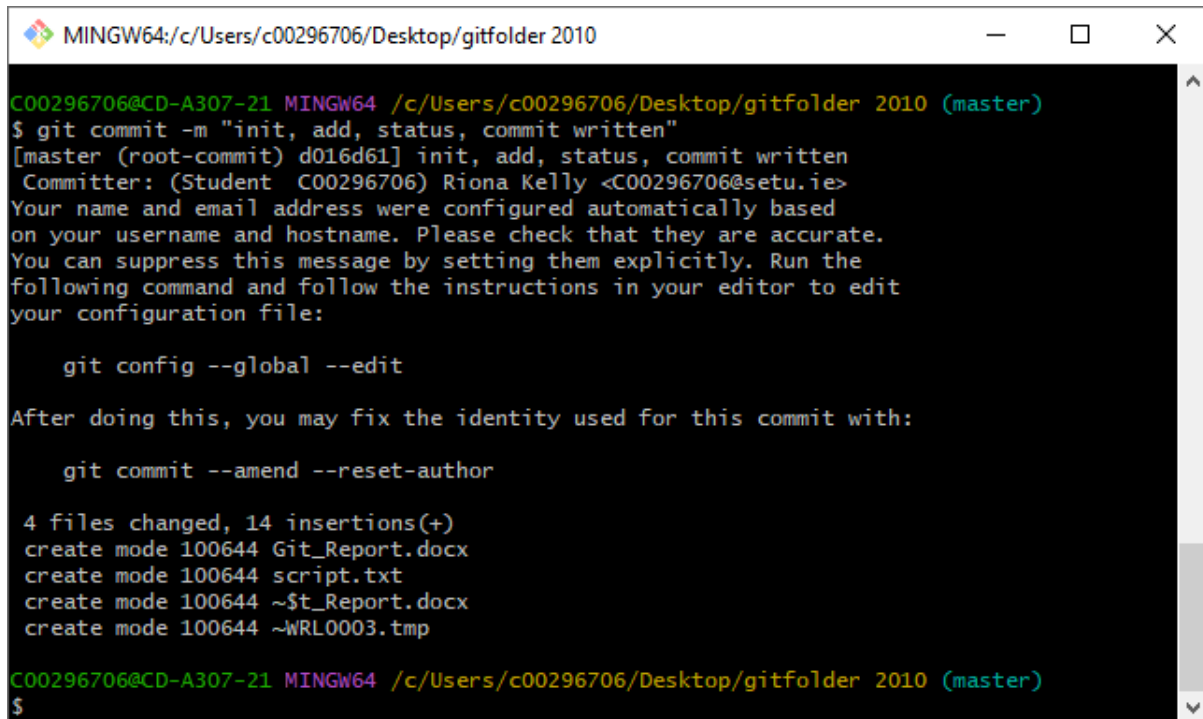
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Git_Report.docx
        new file:   script.txt
        new file:   ~$t_Report.docx
        new file:   ~WRL0003.tmp

C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ |
```

Commit

git commit creates a new commit, commits are snapshots of the repository at different times, I will use git commit -m "init, add, status, commit written" to create a new commit and leave a suitable comment



```
MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010

C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git commit -m "init, add, status, commit written"
[master (root-commit) d016d61] init, add, status, commit written
Committer: (Student C00296706) Riona Kelly <C00296706@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:

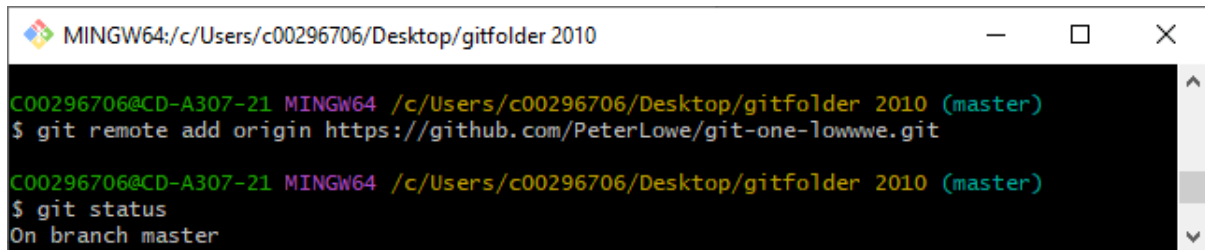
    git commit --amend --reset-author

4 files changed, 14 insertions(+)
create mode 100644 Git_Report.docx
create mode 100644 script.txt
create mode 100644 ~$t_Report.docx
create mode 100644 ~\WRL0003.tmp

C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$
```

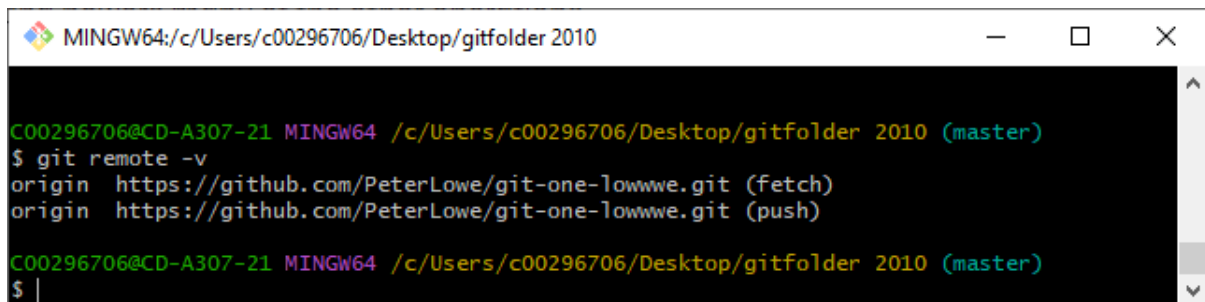
Remote

`git remote add <name> <url>` adds a new repository

A terminal window titled 'MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010'. The prompt is 'C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)'. The user enters '\$ git remote add origin https://github.com/PeterLowe/git-one-lowwwe.git'. The prompt changes to 'C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)'. The user enters '\$ git status'. The output is 'On branch master'.

```
MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010
C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git remote add origin https://github.com/PeterLowe/git-one-lowwwe.git
C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git status
On branch master
```

`git remote -v` lists the remote repositories

A terminal window titled 'MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010'. The prompt is 'C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)'. The user enters '\$ git remote -v'. The output is 'origin https://github.com/PeterLowe/git-one-lowwwe.git (fetch)' and 'origin https://github.com/PeterLowe/git-one-lowwwe.git (push)'. The prompt changes to 'C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)'. The user enters '\$ |'.

```
MINGW64:/c/Users/c00296706/Desktop/gitfolder 2010
C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ git remote -v
origin https://github.com/PeterLowe/git-one-lowwwe.git (fetch)
origin https://github.com/PeterLowe/git-one-lowwwe.git (push)
C00296706@CD-A307-21 MINGW64 /c/Users/c00296706/Desktop/gitfolder 2010 (master)
$ |
```

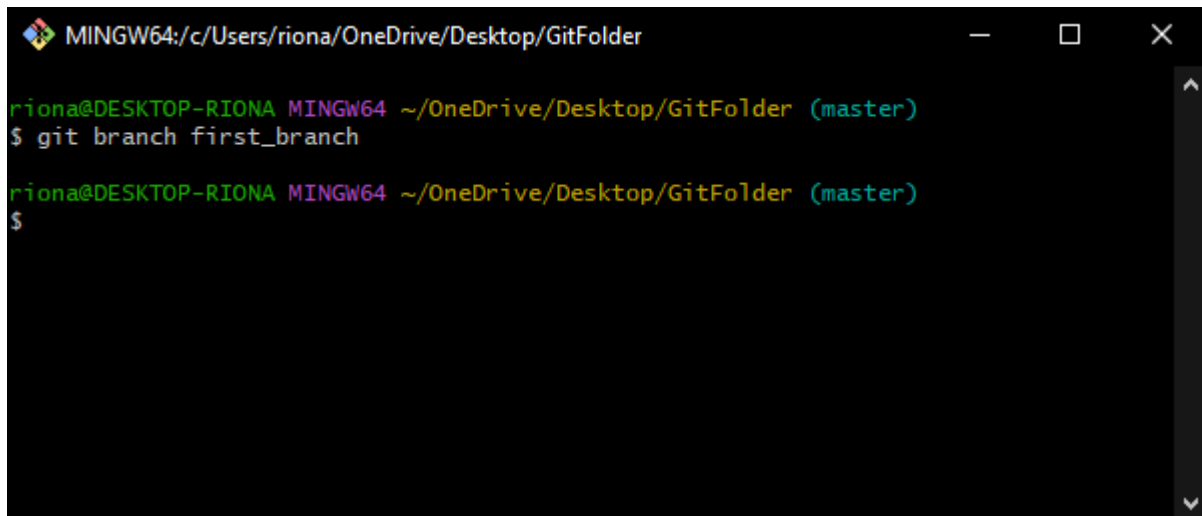
Push

git push updates the remote node with the current node

this command didn't work for me for issues that I am not sure of unfortunately

Branch

git branch creates a new branch in the local repository, the term after branch will be the branch name

A screenshot of a Windows terminal window with a black background and white text. The title bar at the top reads 'MINGW64:/c/Users/riona/OneDrive/Desktop/GitFolder'. The terminal shows a user prompt 'riona@DESKTOP-RIONA MINGW64 ~/OneDrive/Desktop/GitFolder (master)' followed by the command '\$ git branch first_branch'. The prompt appears again on the next line, indicating the command has been executed successfully.

```
MINGW64:/c/Users/riona/OneDrive/Desktop/GitFolder

riona@DESKTOP-RIONA MINGW64 ~/OneDrive/Desktop/GitFolder (master)
$ git branch first_branch

riona@DESKTOP-RIONA MINGW64 ~/OneDrive/Desktop/GitFolder (master)
$
```

git branch --list will list all the branches in the repository

Merge

git merge master/main merges the head with the master, git will automatically apply the changes if separate and warn if there is a conflict before merging

the main should be merged into your branch, only merge your branch into the main when everyone is ready and aware

Log

`git log` displays the entire commit history, you can press <space> for more or <q> to quit

`git log --stat` shows the files altered and the lines that have been changed

`git log --graph` will draw the branch paths

`git log -n` will only display the last (n) commits

Stash

git stash makes a local copy of the current working directory

git stash list lists all the previously pushed stashes

git pop retrieves stash {0} (the last stash of the edited files) and copies the files into a working directory, numbers other than {0} can be retrieved with git apply stash {(n)}