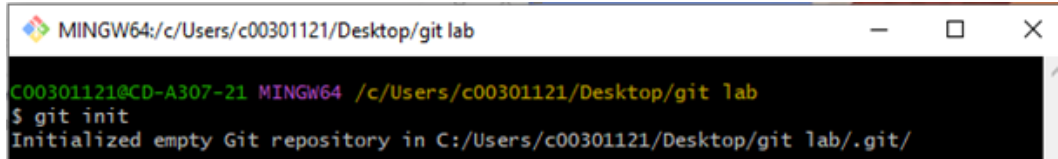


Git Guide

By Tymoteusz Walicnowski

1. Init

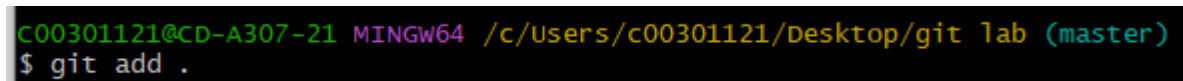
A terminal window titled 'MINGW64:/c/Users/c00301121/Desktop/git lab' showing the command '\$ git init' and its output 'Initialized empty Git repository in C:/Users/c00301121/Desktop/git lab/.git/'.

```
MINGW64:/c/Users/c00301121/Desktop/git lab
C00301121@CD-A307-21 MINGW64 /c/Users/c00301121/Desktop/git lab
$ git init
Initialized empty Git repository in C:/Users/c00301121/Desktop/git lab/.git/
```

Format: `git init`

Creates a new git repository

2. Add

A terminal window showing the command '\$ git add .' and the prompt '(master)' at the end of the line.

```
C00301121@CD-A307-21 MINGW64 /c/Users/c00301121/Desktop/git lab (master)
$ git add .
```

Format: `git add .`

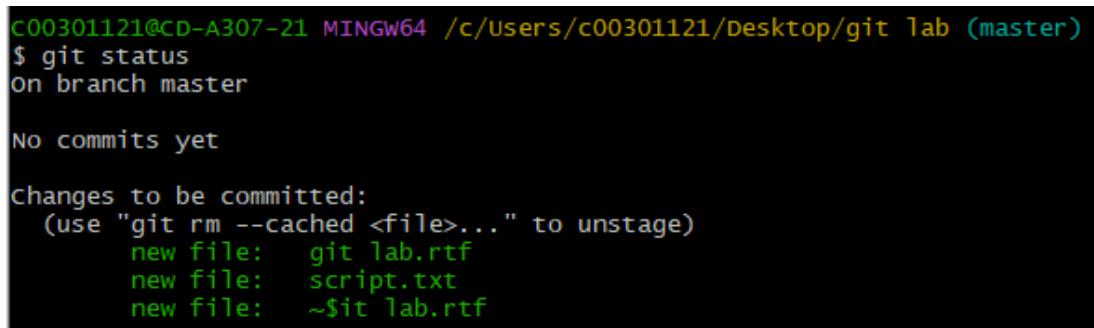
Adds all files (excluding ones included in .gitignore) to the staging area of the next commit

`git add filename/pattern`

Specifies which filename/pattern to add to the next commit, rather than all of them.

The add command adds files to the next commit only. If a file is modified after it is added (but not committed), it will not be updated for the commit and will need to be added again.

3. Status

A terminal window showing the output of the '\$ git status' command, indicating the current branch is master and listing new files to be committed.

```
C00301121@CD-A307-21 MINGW64 /c/Users/c00301121/Desktop/git lab (master)
$ git status
On branch master

No commits yet

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

Format: `git status`

Lists files, modified files in working area, added files in staging area and current branch

4. Commit

```
C00301121@CD-A307-21 MINGW64 /c/Users/c00301121/Desktop/git lab (master)
$ git commit -m "starting report"
[master (root-commit) 7c9349b] starting report
  Committer: (Student C00301121) Tymoteusz walichnowski <C00301121@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

3 files changed, 208 insertions(+)
create mode 100644 git lab.rtf
create mode 100644 script.txt
create mode 100644 ~$it lab.rtf
```

Format: `git commit -m "comment"`

`git commit`

The git command is used to create a new revision point on the local repository. A comment can be entered to detail the changes/added/removed features, this is important for letting other users know the changes made.

5. Remote

6. Push

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 40.60 KiB | 284.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/PeterLowe/git-one-TymekWalichnowski.git
  1e6904a..cb4c1a3 master -> master
```

Format: `git push`

The git push command updates the remote node with the current node. The current node needs to be associated with the remote mode so we use 'git push -u origin master' to push the local master to the origin on the server. The -u means to associate with 'my master' with the remote master branches, and is only needed this once.

7. Branch

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (master)
$ git branch test_branch
```

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (master)
$ git branch -a
* master
  test_branch
  remotes/origin/master
```

Format: `git branch new_branch_name`

The git branch command is used to create a new branch within the local repository. The -a/-l/-r flags can be used to list all branches located within a repository. A branch is a series of nodes, and a separate parallel version of project currently being worked on.

8. Checkout

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (master)
$ git checkout test_branch
Switched to branch 'test_branch'

ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (test_branch)
$ git branch -a
  master
* test_branch
  remotes/origin/master
```

Format: `git checkout branch_name`

The git checkout command allows the user to make the chosen node into the current HEAD. This ensures that the working directory matches that node/commit. This command won't work if there are modifications in the local branch that are different than the one you are switching to. The -f flag can be used to discard non committed files, but this is risky and can end with lost files.

9. Merge

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (test_branch)
$ git merge master
Already up to date.
```

Format: `git merge master`

The git merge master command is used to merge the head with the master branch. Git will automatically apply changes if they are in separate elements of the project. However, it will warn of a conflict and not merge if the two merged branches have the same object but different properties under it. It is advisable to edit the relevant files first and then add them to the commit.

10. Log

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (test_branch)
$ git log
commit cb4c1a361f627140951edd7ad2de55e79346b5e9 (HEAD -> test_branch, origin/master, master)
Author: ARandomLemon <c00301121@setu.ie>
Date: Mon Oct 23 20:14:26 2023 +0100

    test commit

commit 1e6904a1774e12c9818db3ba0c73aafc8729cc5b
Author: (Student C00301121) Tymoteusz Walichnowski <C00301121@setu.ie>
Date: Mon Oct 23 15:40:24 2023 +0100

    updated word document

commit f4345b2abfcf34d3af7130cea418cc957d8aaf7f
Author: (Student C00301121) Tymoteusz Walichnowski <C00301121@setu.ie>
Date: Fri Oct 20 10:08:22 2023 +0100

    added text for rm

commit 7c9349b8f3f5d4383916df0a92ef9ce2af0842c3
Author: (Student C00301121) Tymoteusz Walichnowski <C00301121@setu.ie>
Date: Fri Oct 20 09:44:44 2023 +0100

    starting report
```

Format: git log

Git log can be used to display the entire commit history of a repository. While navigating the history, pressing the space key will bring up more, while pressing the q key will quit the navigation. The --stat flag can be added to show the files altered and what lines were changed, while the --graph flag can be used to draw the paths of branches, adding --oneline will draw the paths with a condensed output. The -n flag while only show the last amount of specified commits, input as -n.

11. Clone

12. Pull

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 802.66 KiB | 1.55 MiB/s, done.
From https://github.com/PeterLowe/git-one-TymekWalichnowski
   f4345b2..1e6904a  master    -> origin/master
Updating f4345b2..1e6904a
Fast-forward
 git lab.rtf | 116648 ++++++
~WRL3104.tmp |    206 +
 2 files changed, 116792 insertions(+), 62 deletions(-)
 create mode 100644 ~WRL3104.tmp
```

Format: git pull

Pulls the current online repository into the local directory.

13. Stash

```
ADMIN@DESKTOP-64V6F7D MINGW64 ~/Desktop/git lab (test_branch)
$ git stash
Saved working directory and index state WIP on test_branch: cb4c1a3 test commit
Unlink of file 'git lab.rtf' failed. Should I try again? (y/n) y
Unlink of file 'git lab.rtf' failed. Should I try again? (y/n) n
error: unable to unlink old 'git lab.rtf': Invalid argument
fatal: Could not reset index file to revision 'HEAD'.
```

Format: **git stash**

The git stash command creates a local copy of the current working directory for the user. The --list flag can be used to list previously pushed stashes. While the --pop flag can be used to retrieve the stash{0} and copy the files into a working directory. The git apply stash{1} command will retrieve stash{1} and copy it into the current directory.

14. Rm

Format: **git rm <filename>**

The git rm command is used to remove files from a github repository. It is the opposite of the add command. By default it deletes file from both repository and from the file system, the --cached flag can be used so only the file in the repository is deleted. The -r flag is used to recursively remove folders. The -n flag is used if the user wishes to do a dry run, listing the files that would be deleted by the command, but not actually deleting them.