Git

Created By	Ⓜ Mohamed Abdelrahman
Last Edited	@Apr 21, 2020 4:47 PM
Tags	

What is Git ?

Version Control System

- Software designed to record changes to files over time
- Ability to revert back to previous file version or project version
- Compare changes made to files from one version to another
- Version control any plain text file not just source code

Three Stages of a File

▼ Committed

mean that files is stored safely in repo of the project

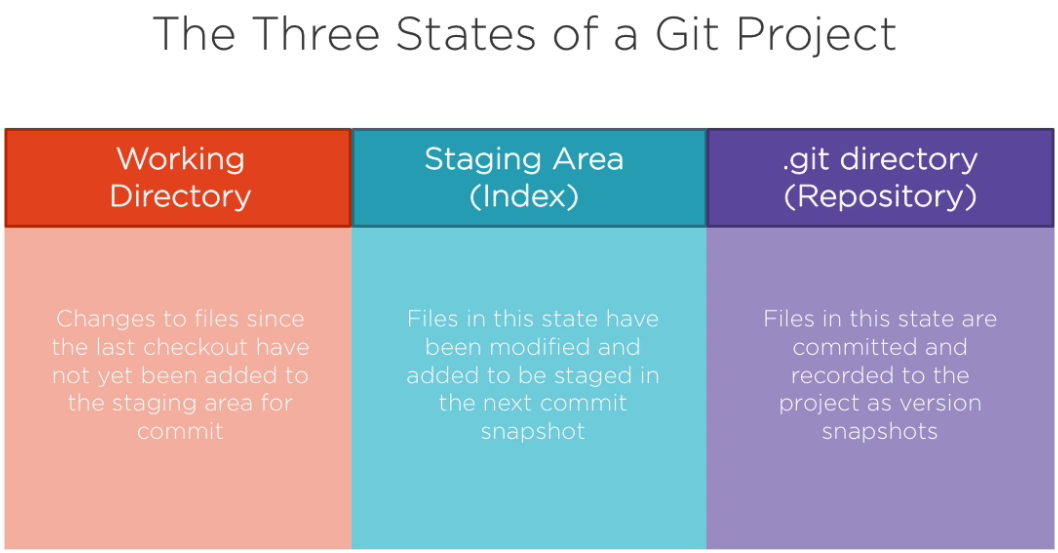
▼ Modified

when make changes in last commit, just introduced new changes but has committed them yet

▼ Staged

when changes finished and ready to commit.

Three States of git project



Command Line

> github-git-cheat-sheet.pdf
> git-docs-all-commends

Tracked files by git has three stages

- Committed

  Unmodified changes from the last commit snapshot, when make changes to these files content then moved to modified stage

- Modified

  Changes made to files since last commit snapshot, when you satisfied with changes and stage them for commit will move to staged area

- Staged

  Changes marked to be added into the next commit snapshot, now can commit it to origin repositories

Untracked Files

mean that git sees a new file that didn't exist in her last commit. and add them to stage are to be ready to commit

```
$ git status
$ git status --short
```

> This commend show the status of tracked and untracked files for your local and remote origin repository

```
$ git diff
$ git dif — staged
```

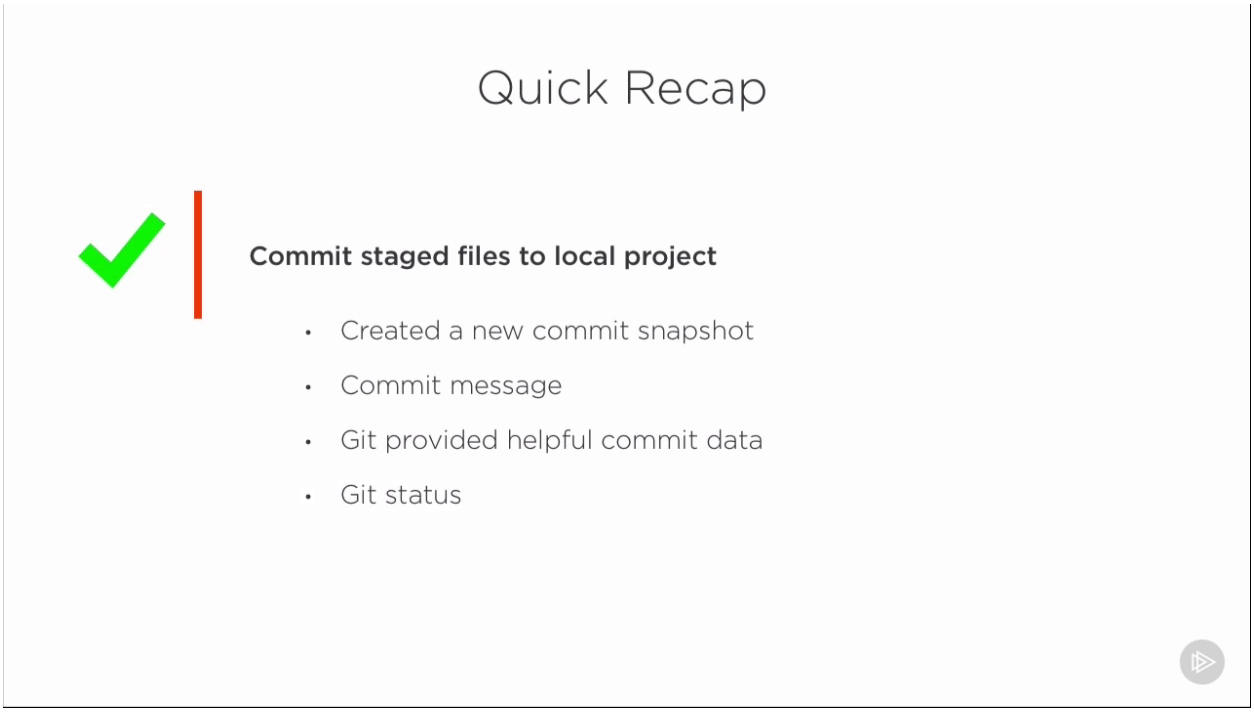
> this commend answer the following two questions
> 1- What changes have I staged that are ready to be committed ?
> 2- What changes have I made but not yet staged ?

```
git diff --staged
diff --git a/file1.txt  b/file1.txt
Index 9863745..f30c839  100644
--- a/file 1.txt
+++ b/file1.txt
@@ -12, 2 +12, 3 @@
Example lines…
- Old content
+ New content
```

◂ Compared Files
◂ File Metadata
◂ Change Markers for File A/B

◂ Chunk Header
◂ Chunk Changes

Push committed changes to Origin Master branch



After Commit push these files from local project to remote origin repository on github

```
$ git push origin master
$ touch <file_name>      # create a new file
```

Track commits and see the history of previous commits

```
$ git log
$ git log -1
$ git log --oneline
$ git log --stat
$ git log --patch
```

Commit Message

There are guidelines are followed to commit message to be useful and helpful to other contributors

there are blog that had a lot of information about these seven rules in this blog

> 1- Separate subject from body with a blank line
> 2- Limit the subject line to 50 characters
> 3- Capitalize the subject line
> 4- Do not end the subject line with a period
> 5- Use the imperative mood in the subject line
> 6- Wrap the body at 72 characters
> 7- Use the body to explain what and why vs. how

Remove and Move Files

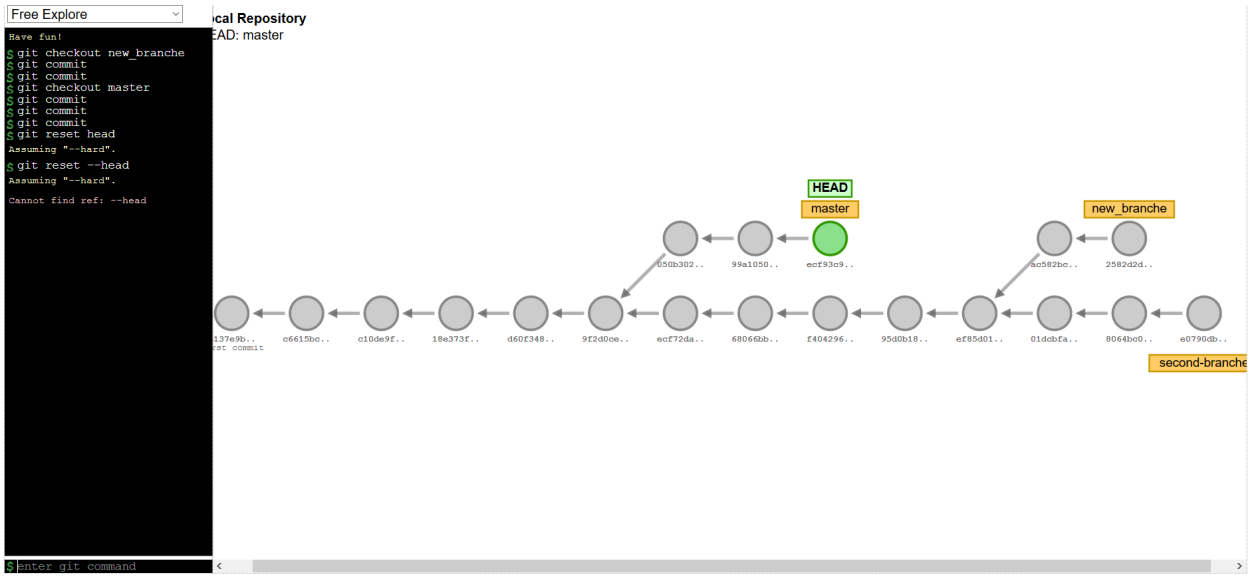
```
$ git rm <file-name> # remove file from project and git untracks it
$ git rm --cached <file-name> # only git untracks it but keeping in local repo
$ git mv <old-file-name> <new-name> # to rename file
$ git reset head <file-name> # to stop tracking this file
```

Branches

branches to organize your code to adding new features or working separately from base code and merge this after finishing or creating pull request. especially working with multiple collaborators to get a copy of version and work in it and merge it

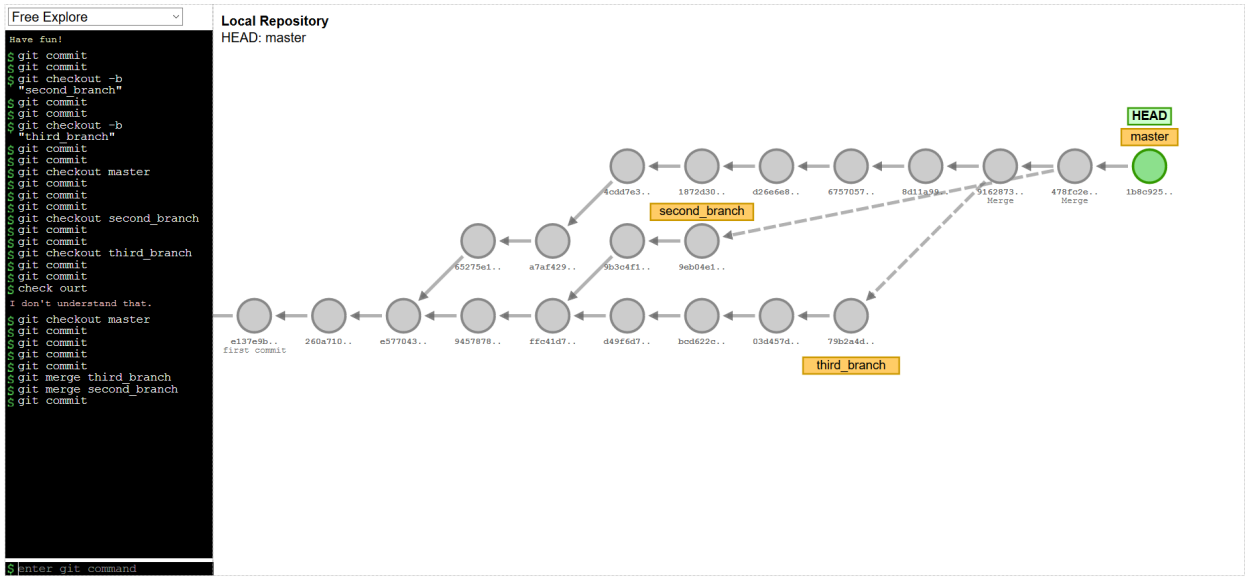
This tool is animate to understanding branching in git

```
$ git checkout <name-of-new-branche> # working/switching on new_branch and commit to it
$ git checkout -b <name-of-new-branch> # creating and working/switching on new_branch and commit to it
$ git checkout master # working/switching in master brance
$ git branch # list of branches do you have
$ git branch <name-of-branche> # create new branch but don't switch to it (checkout)
$ git branch -m <old-name> <new-name> # rename a branch
$ git branch -d <branch-name> # delete a branch
$ git branch -D <branch-name> # Force delete a branch if this branch have commits do not merged yet
```



Merge

merge command it to merge branches with all it's commits and working files to master branch. if you are contributor you can make branches and working at them and merge them as pull request to your master branch and send it to origin master branch as also pull request to be merged



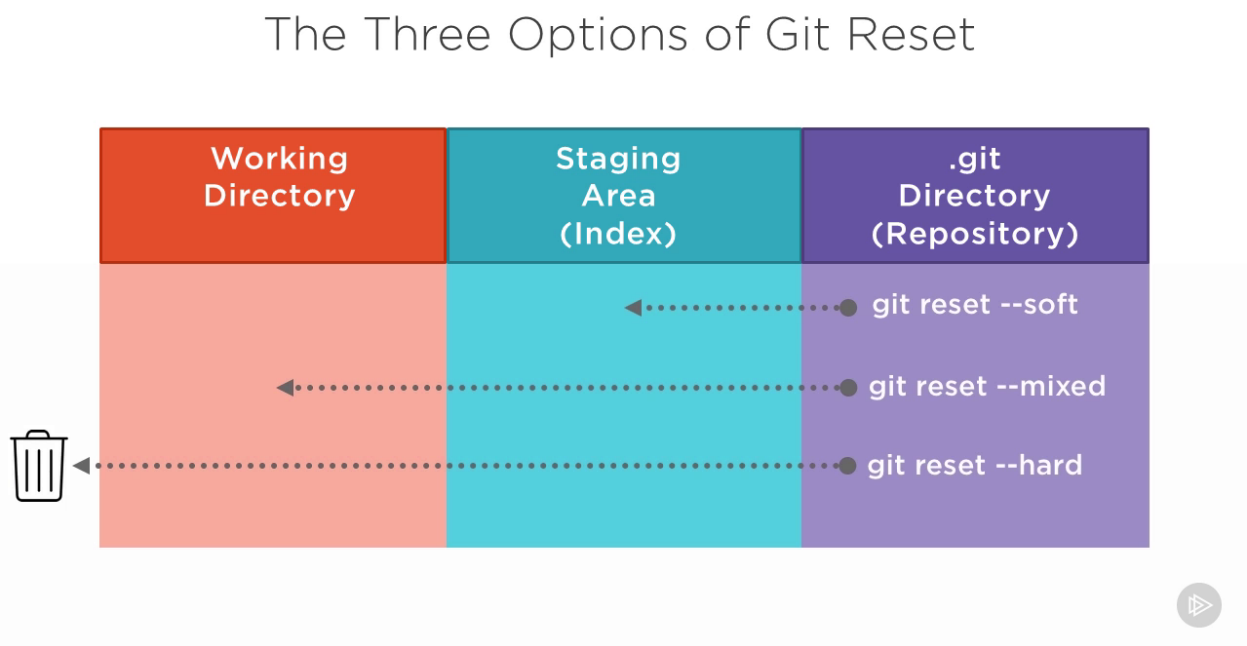
Git stash

git stash temporarily shelves (or stashes) changes you've made to your working copy. The git stash command takes your uncommitted changes (both staged and unstaged), saves them away for later use, and then reverts them from your working copy

```
$ git stash   # saving our working directory and staging area as a secert box
$ git stash list  # get list a stashes or progress chages that we've stashed
$ git stash pop   # get the chages/files from stash to dropped back into his working directory
```

Git reset

git reset allow us move commits from history back into our working or staging area




```
$ git reset --soft <head-of-commit>
$ git reset --mixed <head-of-commit>
$ git reset --hard <head-of-commit>
```