



# INTRODUCTION TO DATA SCIENCE

TA class IV – Git & GitHub

TA : Lee Chi-Hsuan



# Git & GitHub



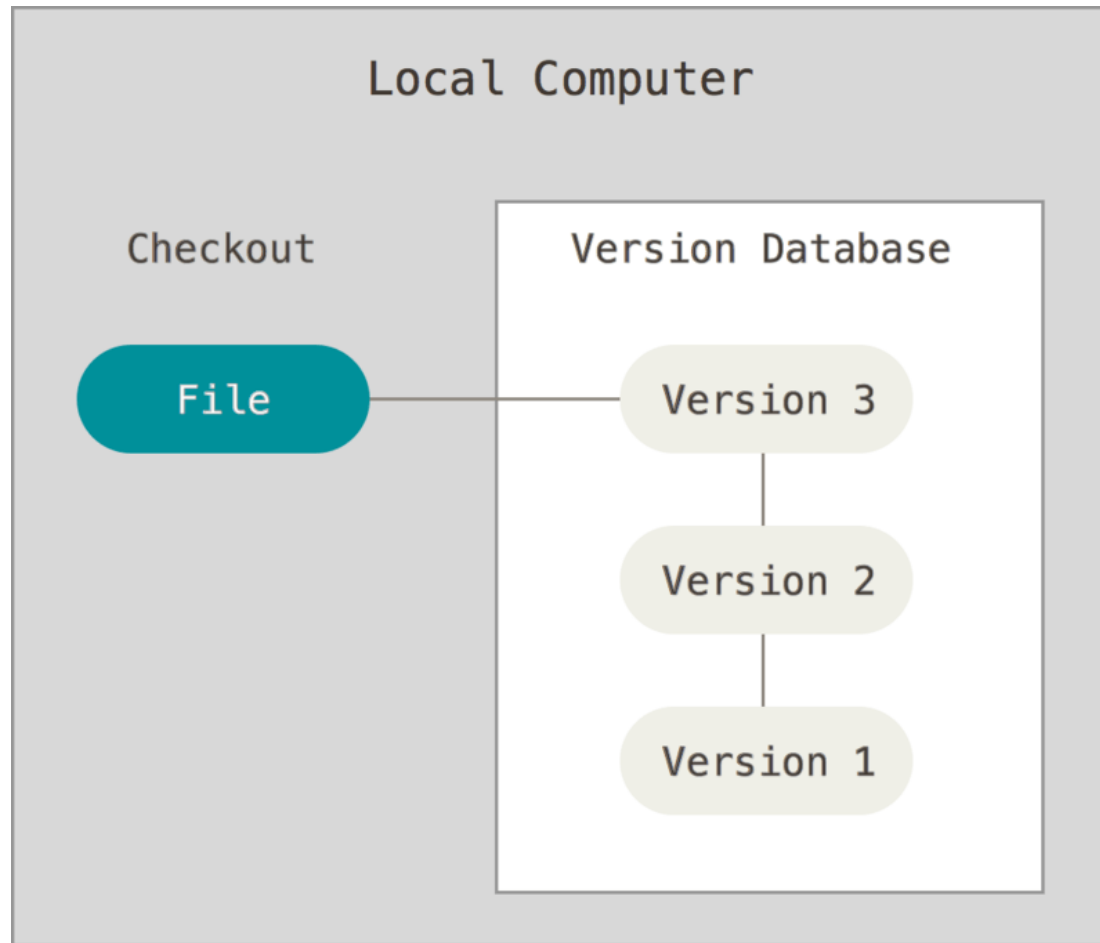
# VCS



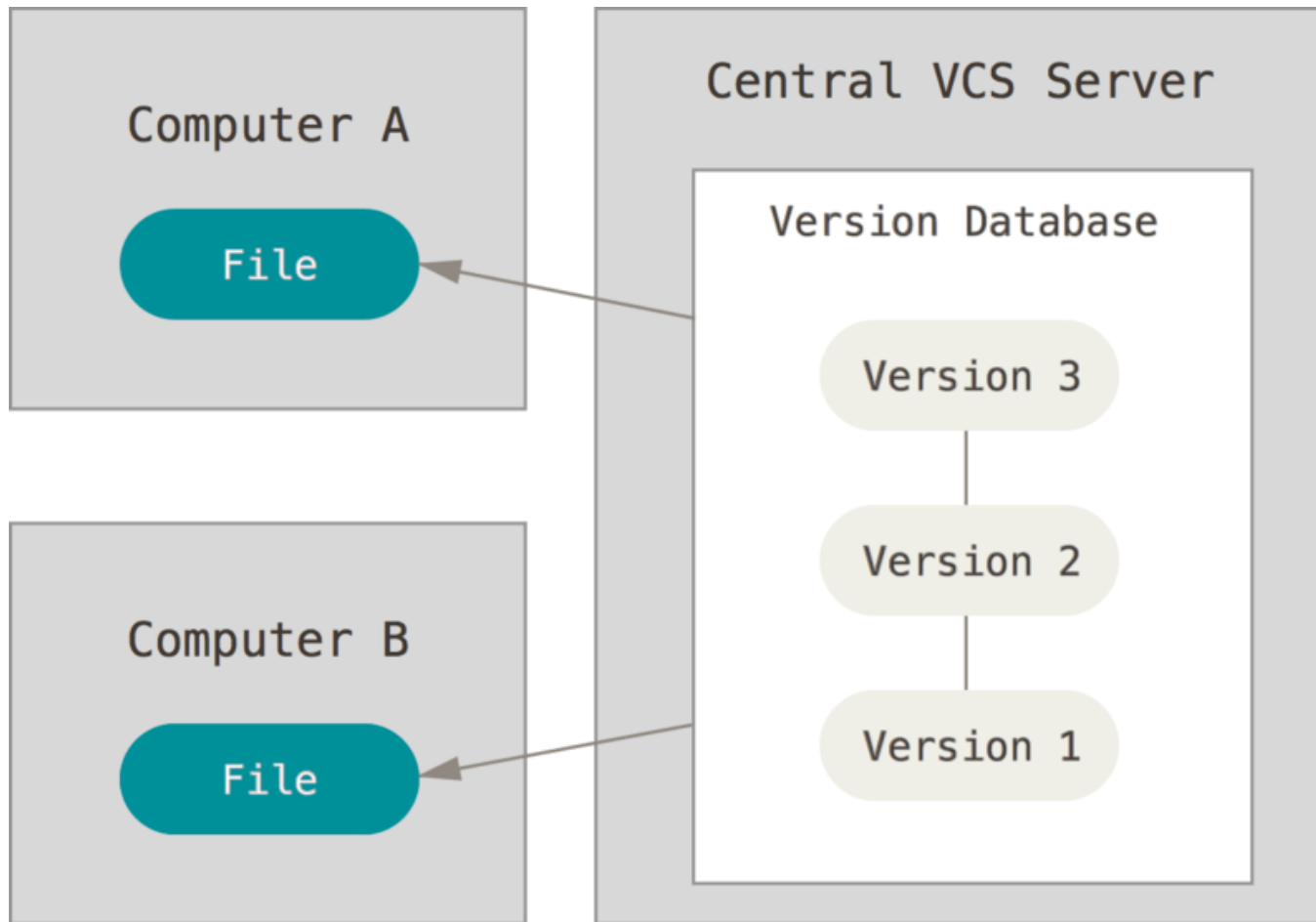
## Version Control System

A system that records changes to files over time so that you can recall specific versions later.

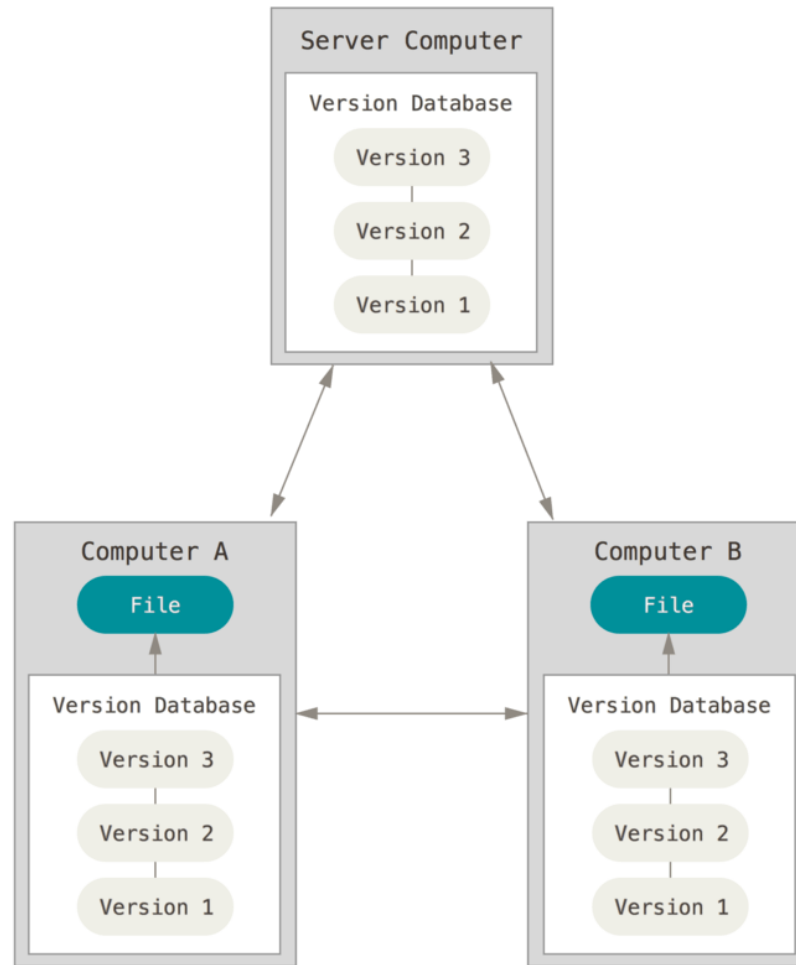
# Local VCS




# Centralized VCS



# Distributed Version VCS





# Git


 **git** --local-branching-on-the-cheap


Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.


Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.




**About**  
The advantages of Git compared to other source control systems.

**Documentation**  
Command reference pages, Pro Git book content, videos and other material.

**Downloads**  
GUI clients and binary releases for all major platforms.

**Community**  
Get involved! Bug reporting, mailing list, chat, development and more.



Latest source Release  
**2.28.0**  
[Release Notes \(2020-07-27\)](#)  
[Download 2.28.0 for Windows](#)

TRIVIAL



# Three state of tracking files :

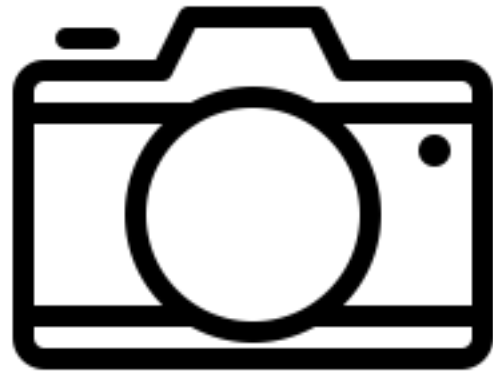
- modified

- staged

- committed

} state





- Snapshotting your repository

- `git checkout <sha1>`  
to check out a particular commit  
(detached HEAD !)

- `git revert <sha1>`  
to revert a particular commit



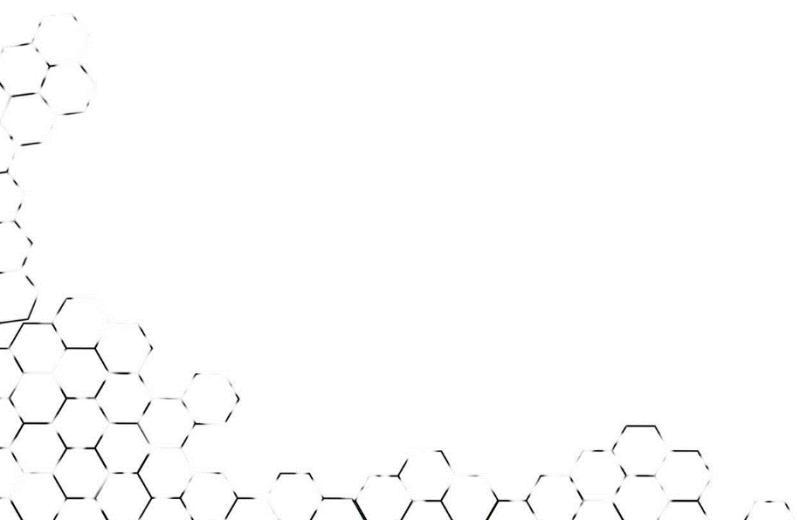
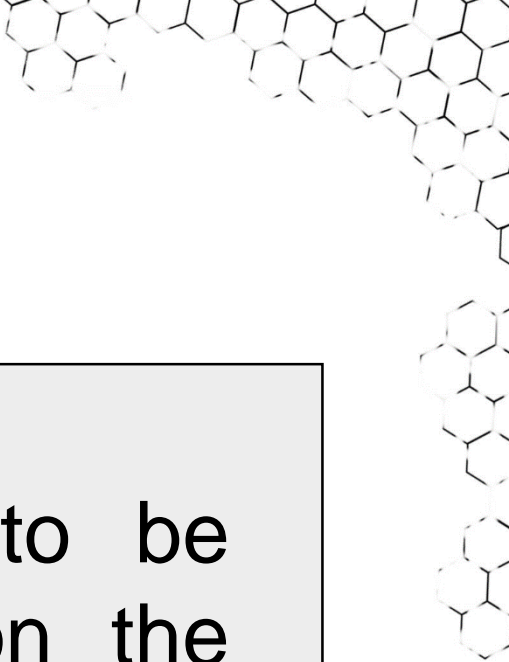
<sha1>

```
commit 6881570d510eb1a6c35484b3c88952d455285f98
Author: CY <cyclonelee34@gmail.com>
Date: Sat Sep 26 21:38:03 2020 +0800
```

Add the .gitignor file in the folder.

# branch

The default branch git create for you when a new git repo is initialized is called “master”.



Branches enable changes to be  
worked on without disruption the  
most current working state,

- git branch branch\_name

```
user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/TA_GitHub (master)  
$ git branch working_on_py
```

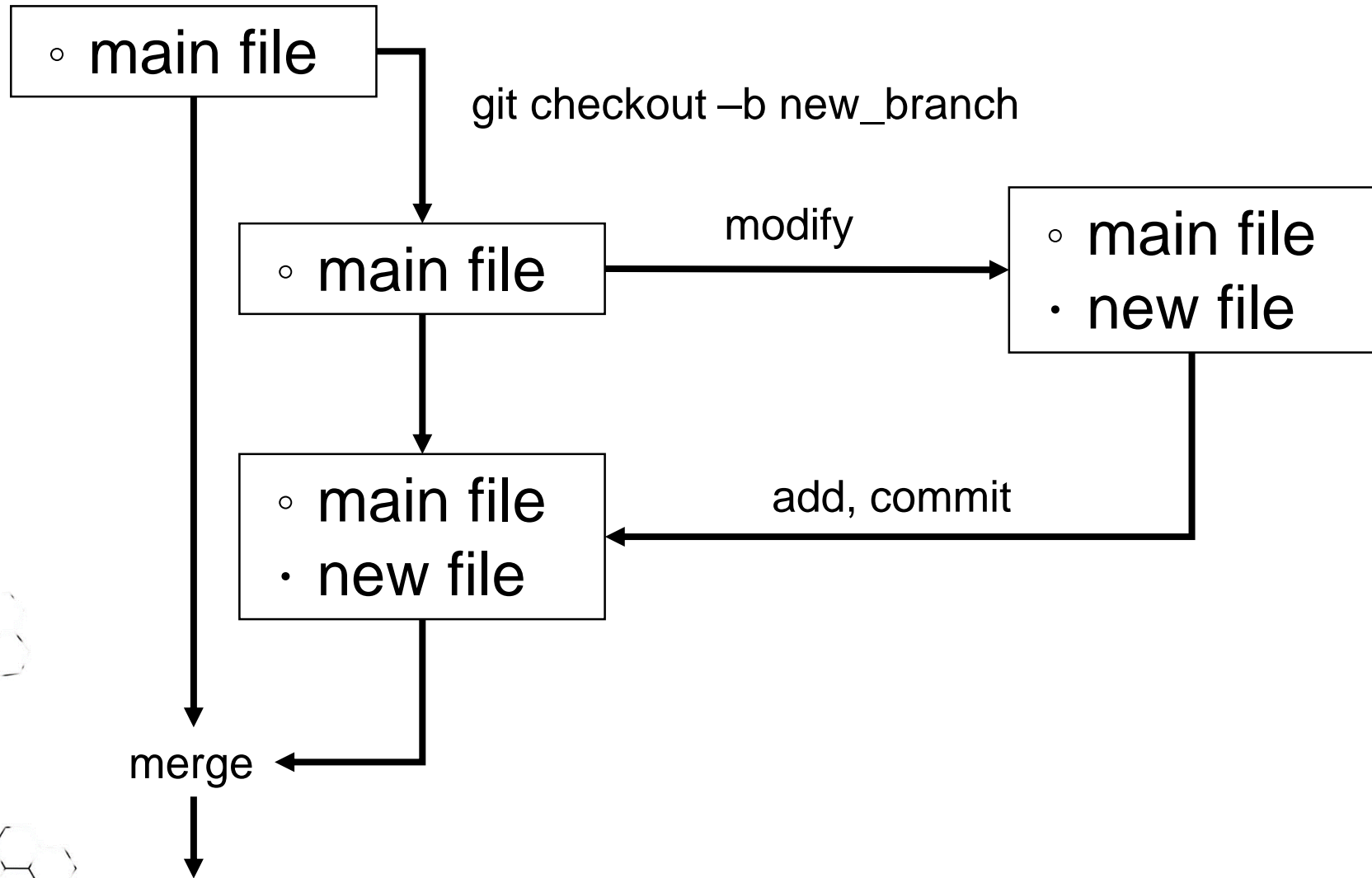
## • git branch

```
user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/TA_GitHub (master)
$ git branch
* master
  working_on_py

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/TA_GitHub (master)
$ git checkout working_on_py
Switched to branch 'working_on_py'

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/TA_GitHub (working_on_py)
$ git branch
  master
* working_on_py
```

\* master



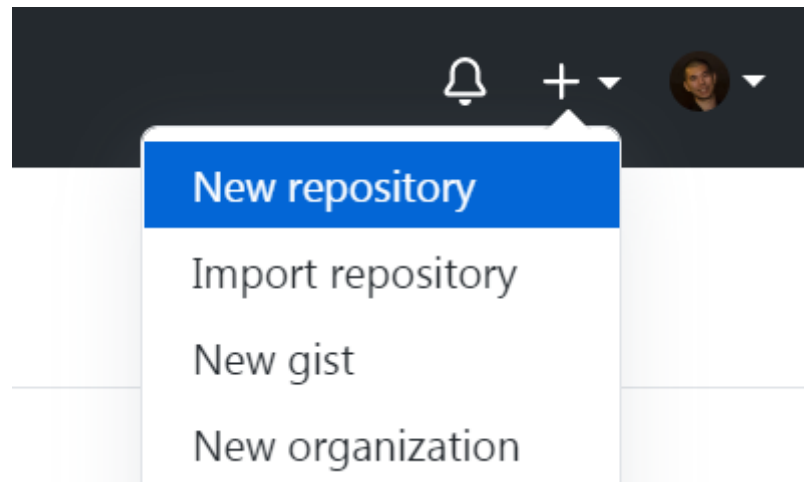
- history of diff. branches are diff.
- `git branch -d branch_name`
- `git merge branch_name`
- fast-forward merge (when no diverge)
- three-way merge (diverge but no conflict)
- conflict ? Fix it, add, and commit



# GitHub



- push your repository to GitHub



## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner \*



cyclonelee24 ▾

Repository name \*

/

Great repository names are short and memorable. Need inspiration? How about **fluffy-spork**?

Description (optional)



**Public**

Anyone on the internet can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.



**Add a README file**

This is where you can write a long description for your project. [Learn more.](#)



**Add .gitignore**

Choose which files not to track from a list of templates. [Learn more.](#)



**Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

## Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

git@github.com:cyclonelee24/test.git



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

## ...or create a new repository on the command line

```
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin git@github.com:cyclonelee24/test.git
git push -u origin master
```



## ...or push an existing repository from the command line

```
git remote add origin git@github.com:cyclonelee24/test.git
git branch -M master
git push -u origin master
```

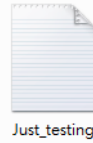




- master vs. main ? !

# • push

HDD文件 媒體櫃  
Draft Code



```
MINGW64:/b/Documents/2020五上/TA/2020 BDA/Draft Code

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git log --graph --oneline
* 9507168 (HEAD -> master, origin/master) First commit for showing how to push stuff to GitHub

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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:   Just_testing.txt

no changes added to commit (use "git add" and/or "git commit -a")

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git commit -a -m 'Second commit'
[master 9da77e9] Second commit
 1 file changed, 1 insertion(+)

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git log --graph --oneline
* 9da77e9 (HEAD -> master) Second commit
* 9507168 (origin/master) First commit for showing how to push stuff to GitHub

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 254 bytes | 254.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:cyclonelee24/test.git
   9507168..9da77e9  master -> master

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.


nothing to commit, working tree clean

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code (master)
$
```



 cyclonelee24 / test


 Unwatch 1

 Star 0


 Fork 0

 Code

 Issues


 Pull requests

 Actions


 Projects

 Wiki

 Security

 Insights

 Settings


 master

 1 branch

 0 tags

Go to file

Add file

 Code



cyclonelee24 Second commit

9da77e9 3 minutes ago 2 commits



Just\_testing.txt

Second commit

3 minutes ago

Help people interested in this repository understand your project by adding a README.

Add a README

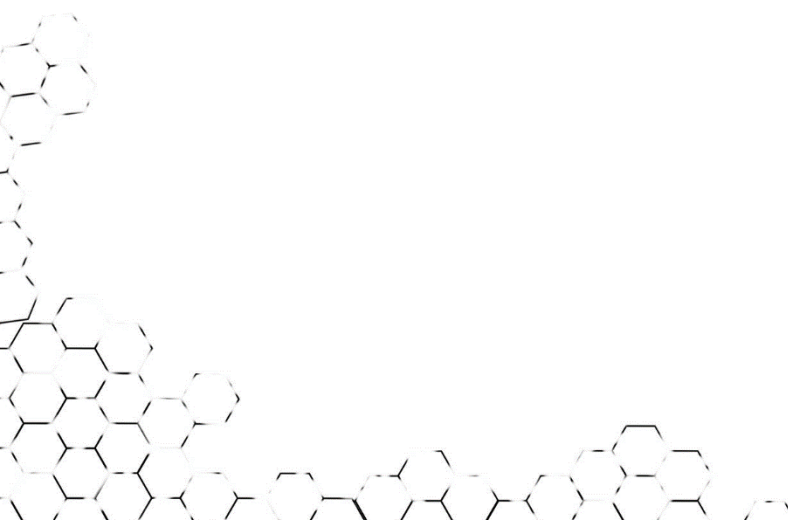
## About

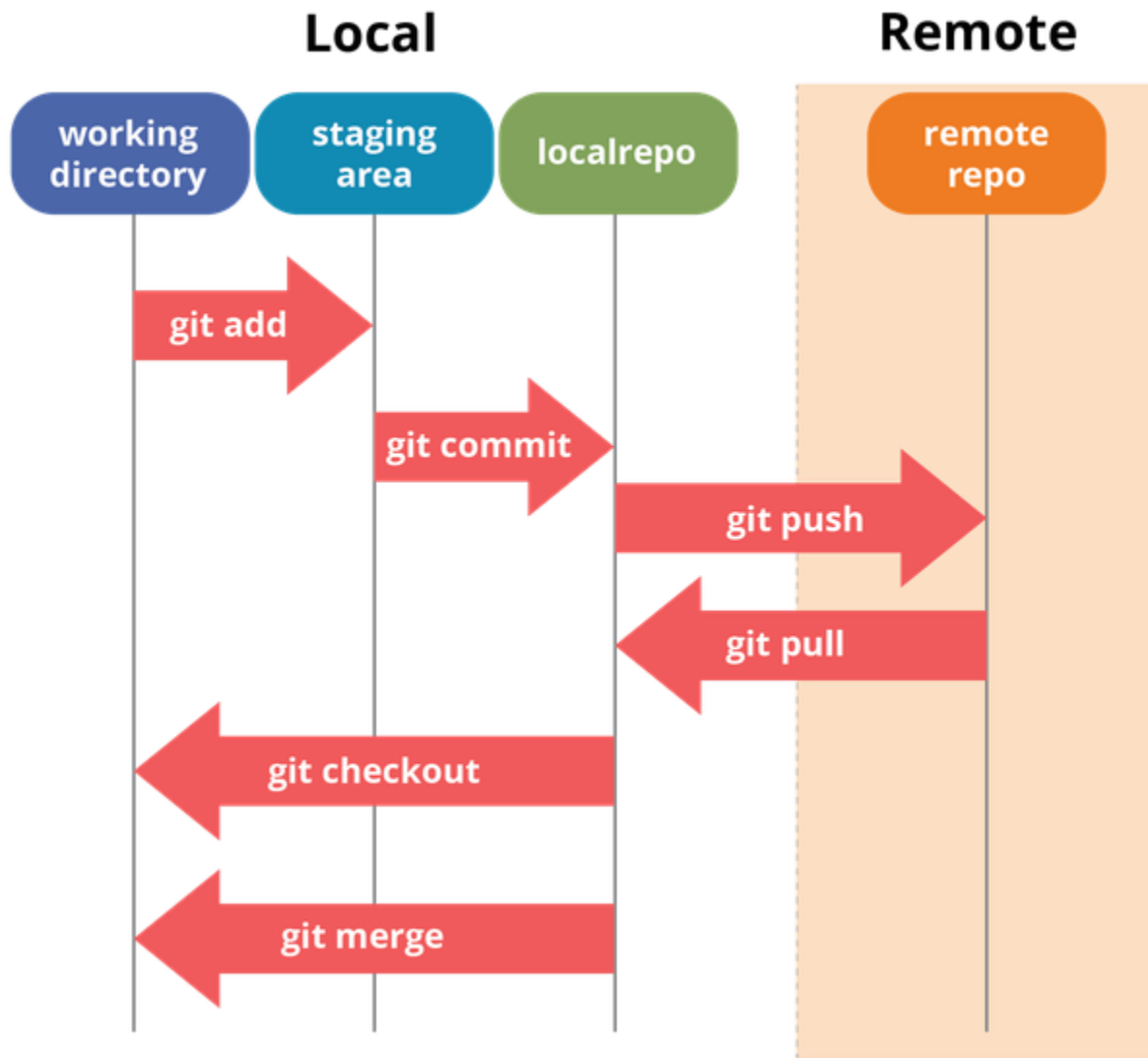


No description, website, or topics provided.

## Releases

No releases published  
[Create a new release](#)



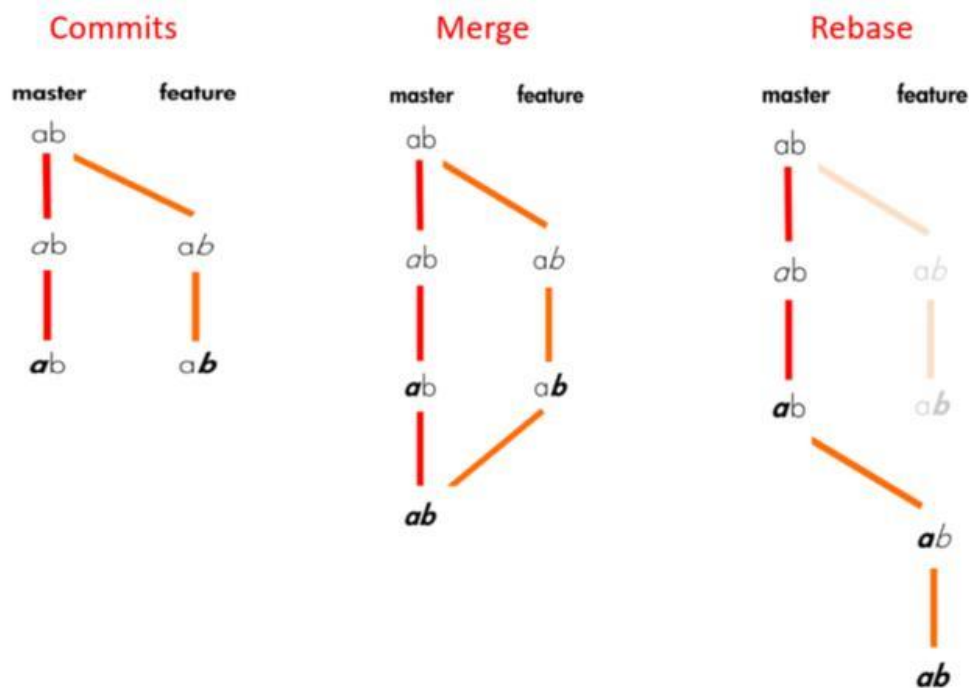



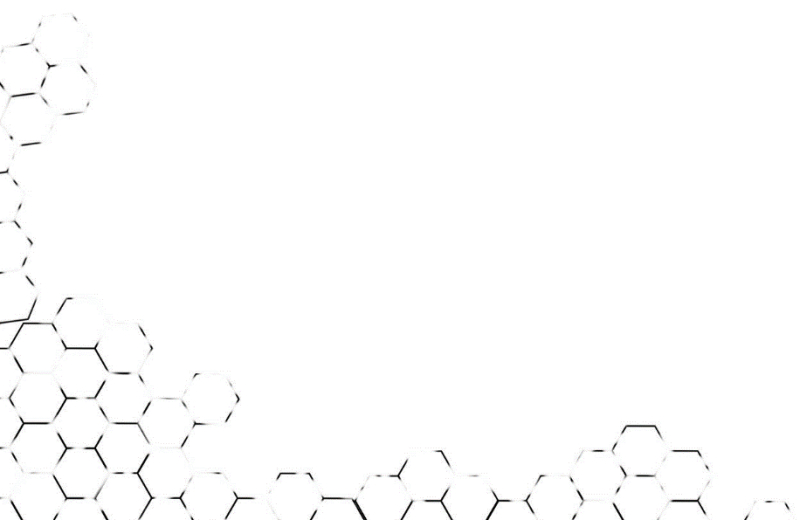


- `git fetch` : copy the commit done with the remote repository to your branch
- `git merge`
- `git pull` = `git fetch` + `git merge`

- if remote repository has changes local repository does not have :  
cannot push (can't fast-forward merge)  
→ sync your local repository first

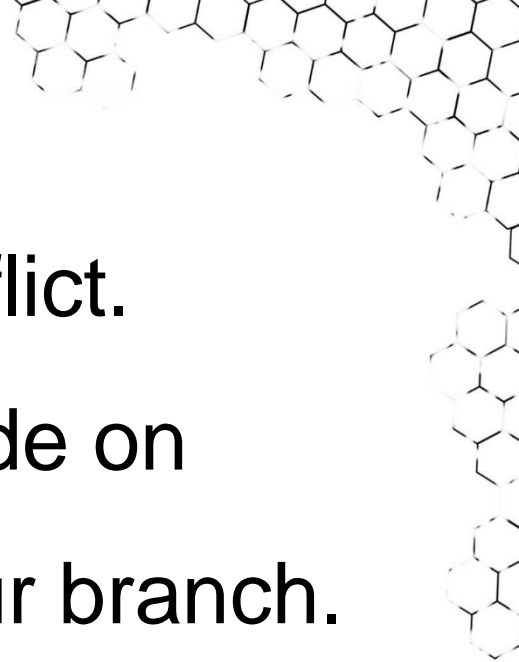

- git rebase : move current branch on top of the rebased branch

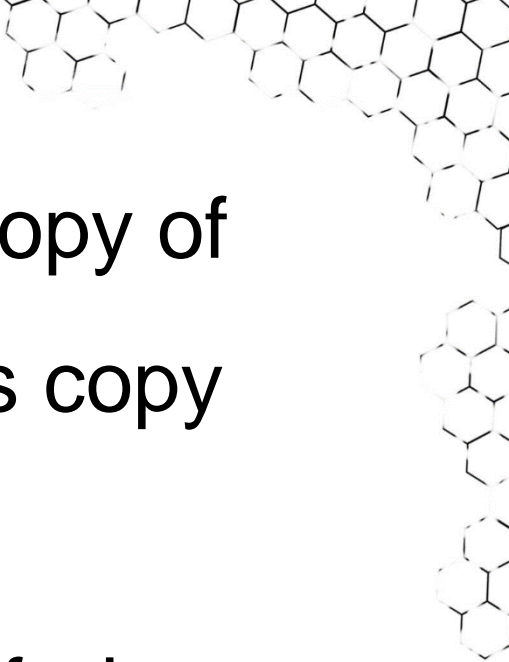
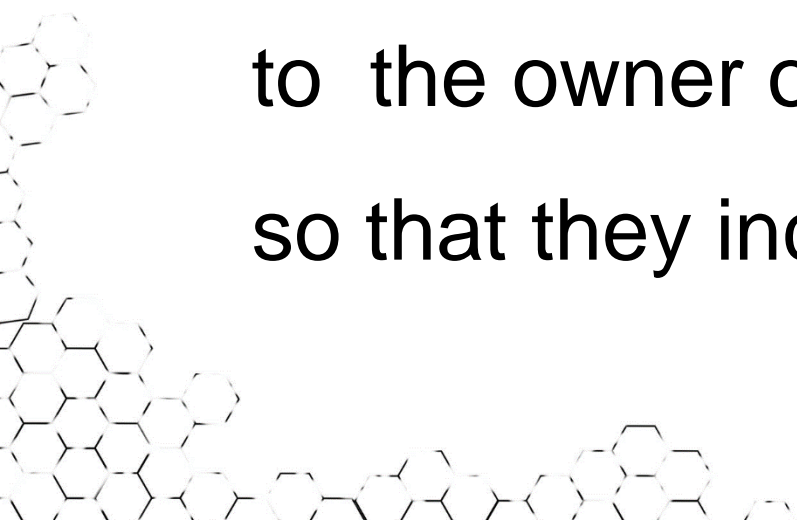


- 
- 
- rebase keep the history linear
  - use it when the changes are minor
  - be careful when using rebase
  - rebase = rewrite the history of commits

# Some remarks

- Use remote repository to incorporate with others.
- Always synchronize your branch before starting any work on you own.
- Avoid large change that modify a lot of different things.

- 
- 
- push/pull often to reduce conflict.
  - Regularly merge changes made on the master branch back to your branch.
  - Should not rebase changes that have been pushed to remote repository.

- 
- 
- Forking : A way of creating a copy of the given repository so that this copy belongs to you.
  - you can push changes to your forks
  - pull request : commit(s) that you send to the owner of the original repository so that they incorporate it into their tree

- proposed file change : commit on your fork → then create pull request
- same branch → same pull request  
different branch → different pull request
- different projects may have different pull request guidelines



CryoliteZ / KMF-Unblock-Extension

Watch 1 Star 51 Fork 2

Code Issues Pull requests Actions Projects Wiki Security Insights



cyclonelee24 / KMF-Unblock-Extension

forked from CryoliteZ/KMF-Unblock-Extension

Watch 0 Star 0 Fork 2

Code Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is even with CryoliteZ:master.

CryoliteZ Add new feature screenshots and contributors section

icons	init commit
manifest.json	Change const to let for initializing observers
popup.html	Add GMAT support
readme.md	Add new feature screenshots and contributors section 6 months ago
script.js	Change const to let for initializing observers for more general use c... 6 months ago

Go to file Add file Code

Clone ?

HTTPS SSH GitHub CLI

git@github.com:cyclonelee24/KMF-Unblock-Extension

Use a password-protected SSH key.

Open with GitHub Desktop

Download ZIP

## About

解除『考满分』区域限制的浏览器扩充功能 A Chrome and Firefox extension that bypasses KMF's region restriction

Readme

## Releases

No releases published  
[Create a new release](#)

## Packages

# • fork vs. clone ?

HDD文件 媒體櫃

Draft Code




KMF-Unblock-E  
xtension

```
MINGW64:/b/Documents/2020五上/TA/2020 BDA/Draft Code

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code
$ git clone git@github.com:cyclonelee24/KMF-Unblock-Extension.git
Cloning into 'KMF-Unblock-Extension'...
remote: Enumerating objects: 30, done.
remote: Counting objects: 100% (30/30), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 71 (delta 14), reused 20 (delta 8), pack-reused 41
Receiving objects: 100% (71/71), 16.42 KiB | 731.00 KiB/s, done.
Resolving deltas: 100% (33/33), done.

user@user-PC MINGW64 /b/Documents/2020五上/TA/2020 BDA/Draft Code
$
```

# More questions ?


 **git** --local-branching-on-the-cheap


Search entire site...


**About**  
**Documentation**  
Reference  
Book  
Videos  
External Links  
**Downloads**  
**Community**


## Reference

Quick reference guides: [GitHub Cheat Sheet](#) | [Visual Git Cheat Sheet](#)


 [Complete list of all commands](#)

 **Setup and Config**  
[git](#)  
[config](#)  
[help](#)  
[bugreport](#)

 **Getting and Creating Projects**  
[init](#)  
[clone](#)

 **Basic Snapshotting**  
[add](#)  
[status](#)  
[diff](#)  
[commit](#)

**Guides**  
[gitattributes](#)  
[Command-line interface conventions](#)  
[Everyday Git](#)  
[Frequently Asked Questions \(FAQ\)](#)  
[Glossary](#)  
[githooks](#)  
[gitignore](#)  
[gitmodules](#)  
[Revisions](#)  
[Submodules](#)  
[Tutorial](#)  
[Workflows](#)

 **Email**