

Git & Github

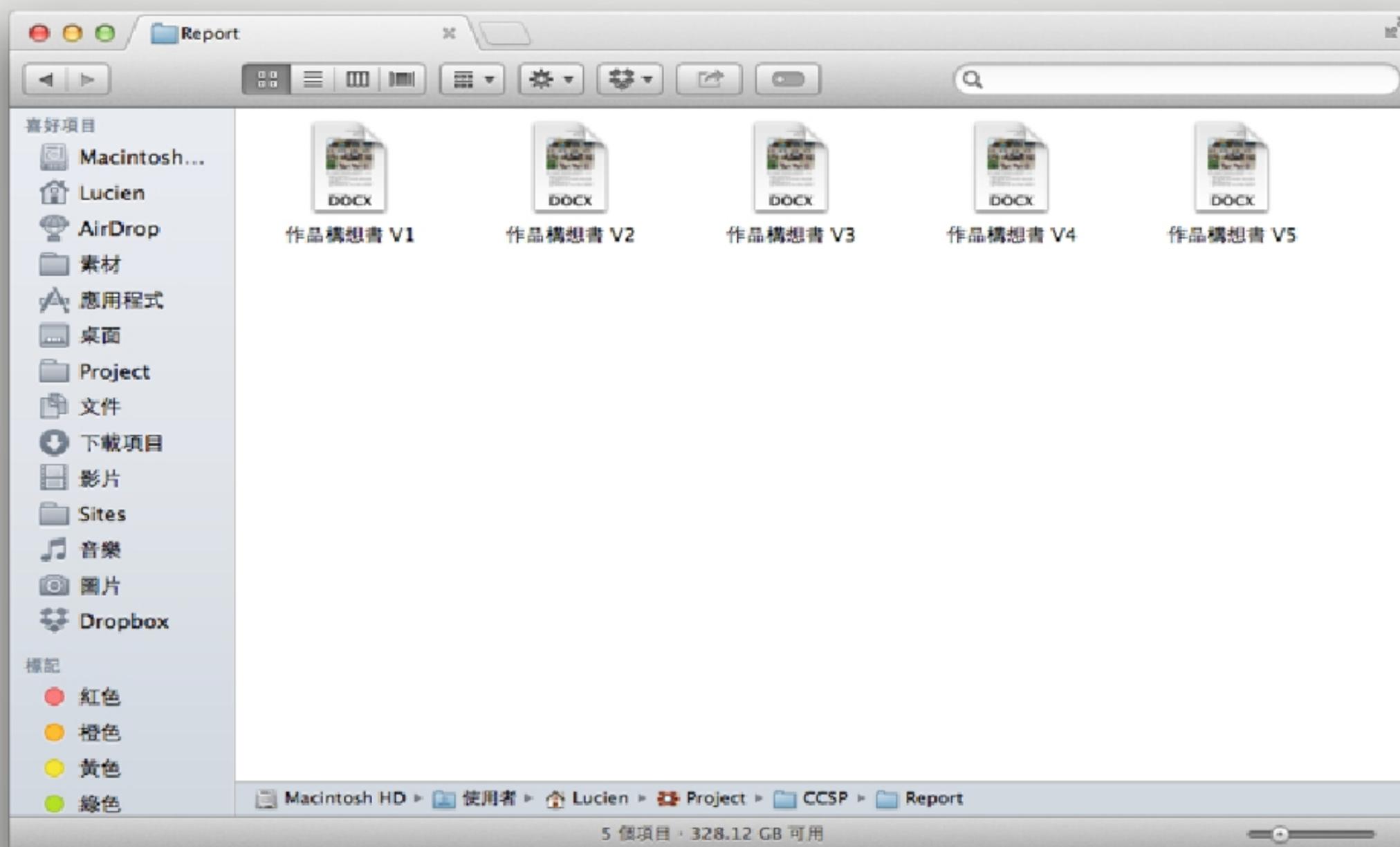
Min Sun

Slides from Lucien Lee

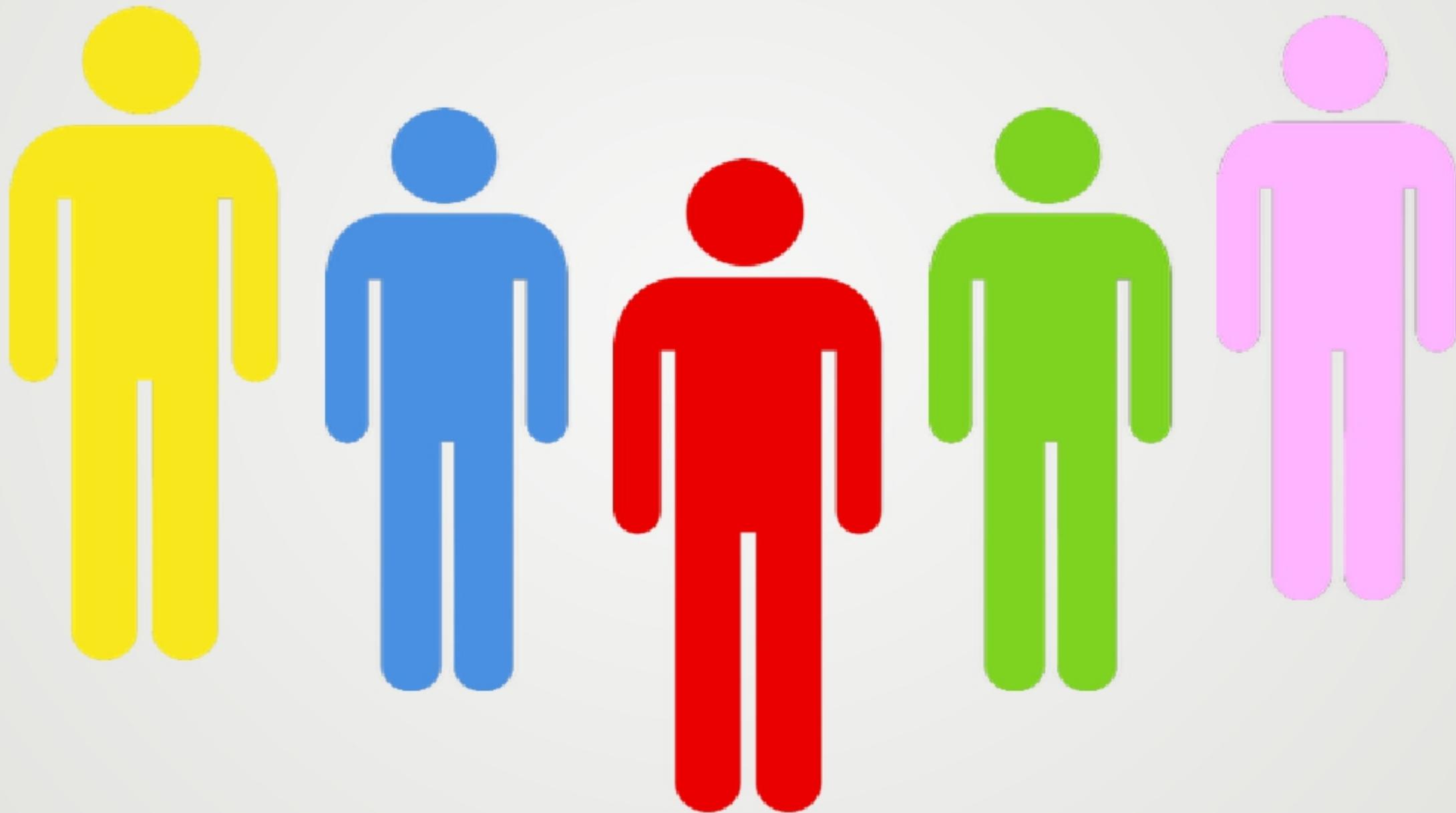
Work alone



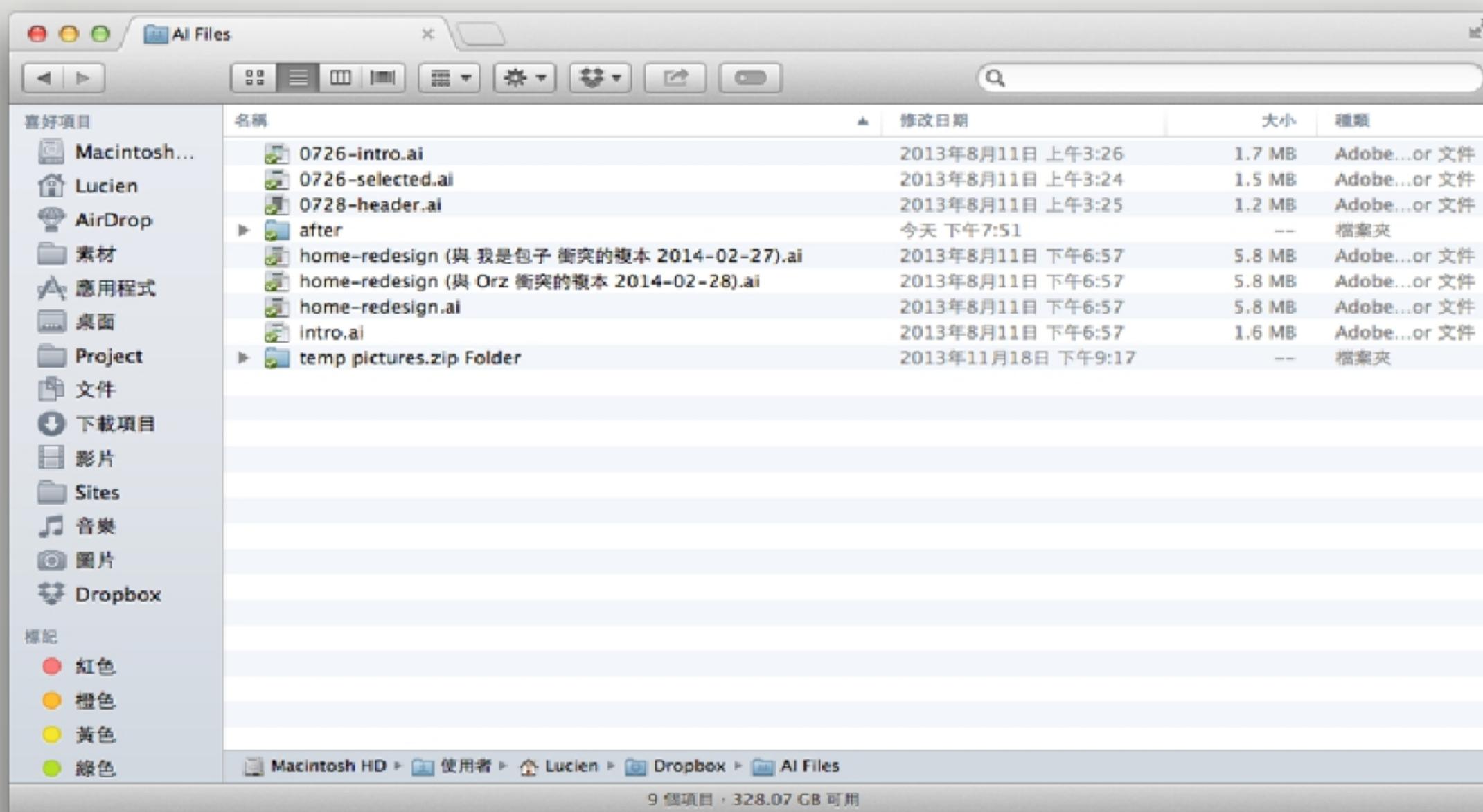
不知不覺分裂增生的報告



Work with others by dropbox, right?



那些年，一起撞牆的夥伴



Does it sounds
happened to you before?

All you need is **version control**

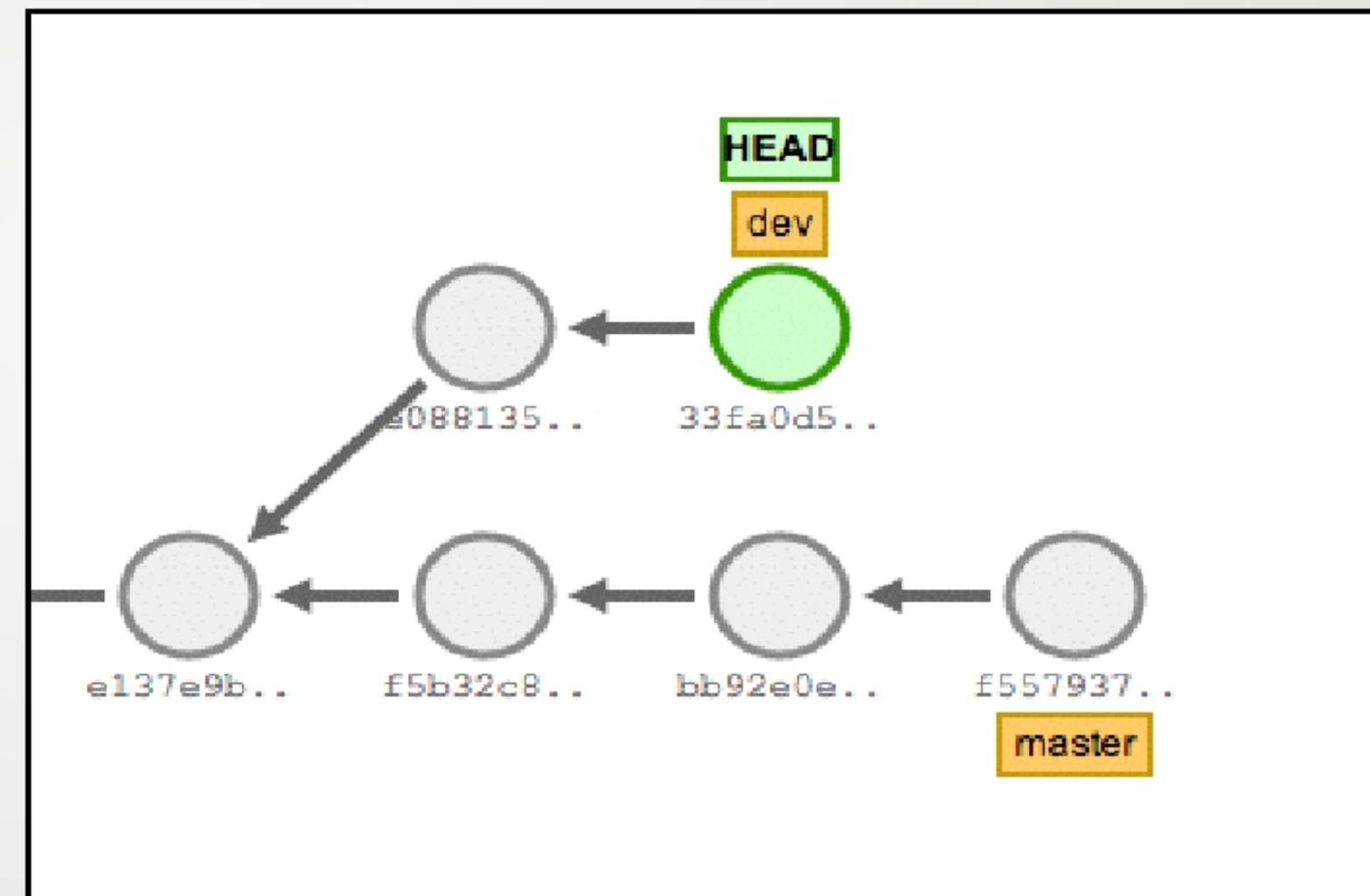
什麼是版本控制

- 儲存進度
- 讀取進度
- 管理不同記錄



什麼是版本控制

- 我們想要：
 - 知道為什麼變更
 - 知道什麼時候變更
 - 知道變更了什麼
 - 知道是誰變更的
 - 切換到過去的進度
 - 切換不同的版本
 - 合併不同的版本





git: Distributed Version Control Systems



OPPAN
GITHUB

GITMAN

- ▶ 安裝
- ▶ 設定
- ▶ 初始

1. SETUP YOUR Git

To-Do B: 安裝



SourceTree

Windows



git for windows

<https://windows.github.com/>

Mac



git for Mac

<https://mac.github.com/>

github app

<https://help.github.com/articles/set-up-git>

Git for Windows

msysgit.github.io

Welcome to the home page of Git for Windows

Git is a powerful version control system aiming to be the fastest decentralized source code management tool on this planet.

Having its root in the Linux development community, Git used to be quite dependent on POSIX features usually only provided by Unix-style Operating Systems. Thanks to the efforts of a few contributors, this project succeeded in providing an almost feature-complete fork of Git on Windows. Being solely driven by volunteers in their spare time, it is nevertheless quit

How to start:

- Startmenu
- Desktop shortcut
- QuickLaunch icon
- Explorer context menu

Something does not work? Have a look at the [Community](#) and how to Report/Fix Bugs.



Links:

- [FAQ](#)
- [Homepage](#)
- [Wiki](#)
- [Downloads](#)
- [Downloads \(build environment\)](#)
- [Repository](#)
- [Repository \(build environment\)](#)
- [Mailing list](#)

GitHub for Windows x

← → C windows.github.com ☆ ≡

GitHub Windows

overview help release notes

+ add ⌂ refresh ⚙

FILTER REPOSITORIES

local

repositories

github

tclm

github

libgit2

libgit2sharp

Introducing GitHub for Windows

Develop on Windows, share on GitHub.

Clone Repositories Browse History Commit Changes Branch Code Share on GitHub.com

libgit2/GitForDelphi

libgit2/libgit2 Clone

libgit2/libgit2-backends

libgit2/libgit2.github.com →

libgit2/libgit2sharp

libgit2/luagit2

libgit2/node-gitteh

libgit2/objective-git

tclem LOG OUT

Windows XP, Vista, 7 & 8

README.md

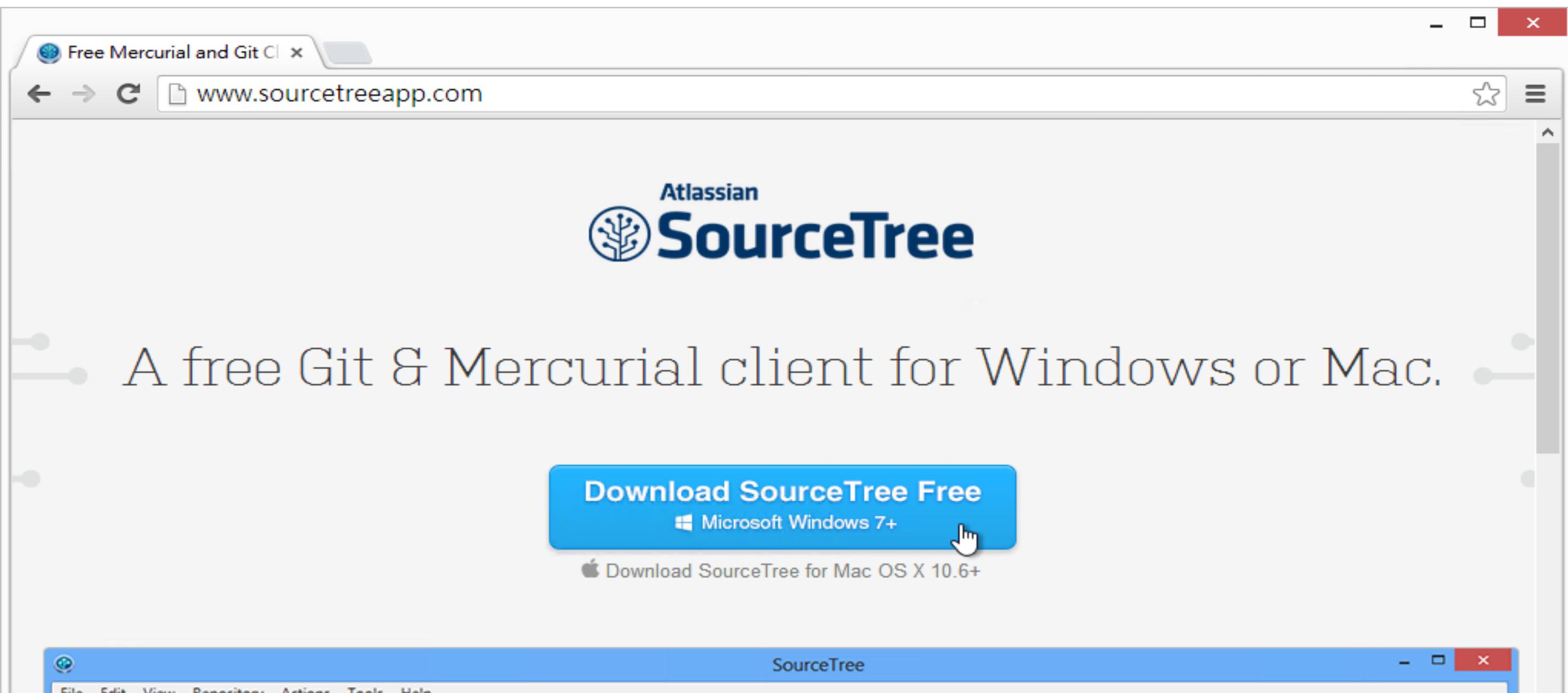
libgit2 - the Git linkable library

build status passing

libgit2 is a portable, pure C implementation of the Git core methods. It is a linkable library with a solid API, allowing you to write native speedups for any language with bindings.

libgit2 is licensed under a **very permissive license** (GPLv2 with a special exception). This basically means that you can link it (unmodified) with any kind of application and release its source code.

- Mailing list: libgit2@librelist.org
- Archives: <http://librelist.com/browser/libgit2/>
- Website: <http://libgit2.github.com>



<http://ithelp.ithome.com.tw/question/10132333>

設定身分

```
$ git config --global user.name <名字>  
$ git config --global user.email <email>  
$ git config --global color.ui true
```

初始化

```
// under your project  
$ git init
```

幫你在專案裡生成 git 所需檔案: .git

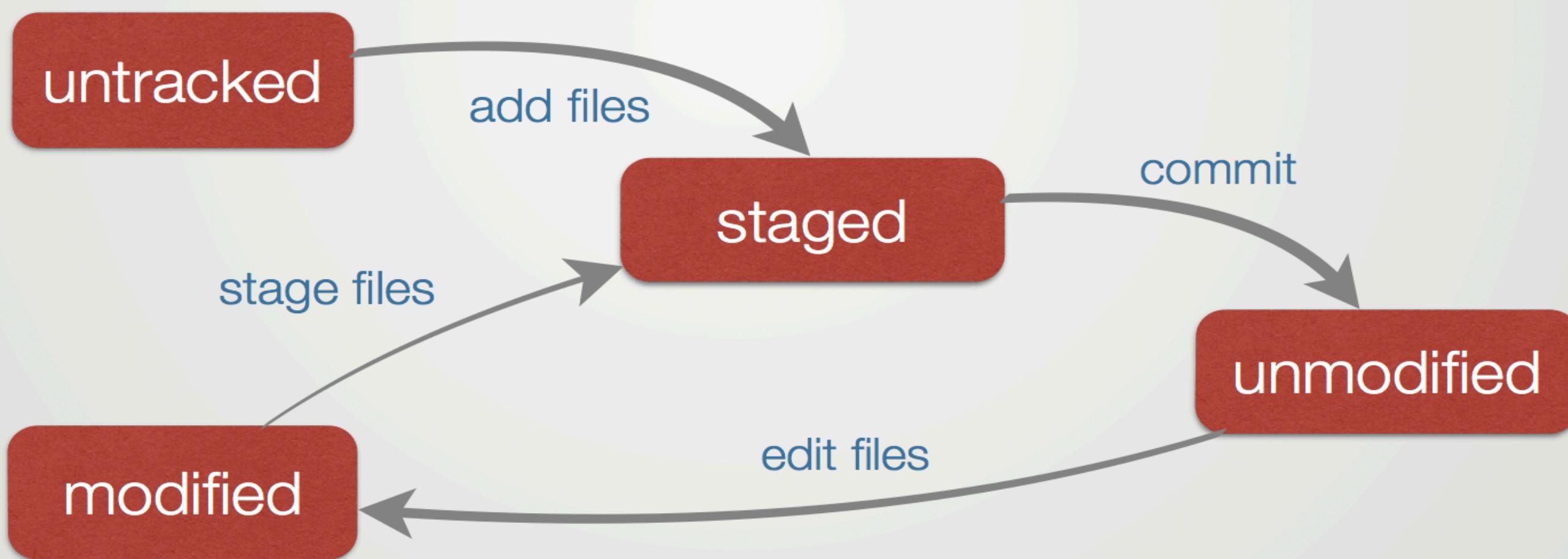
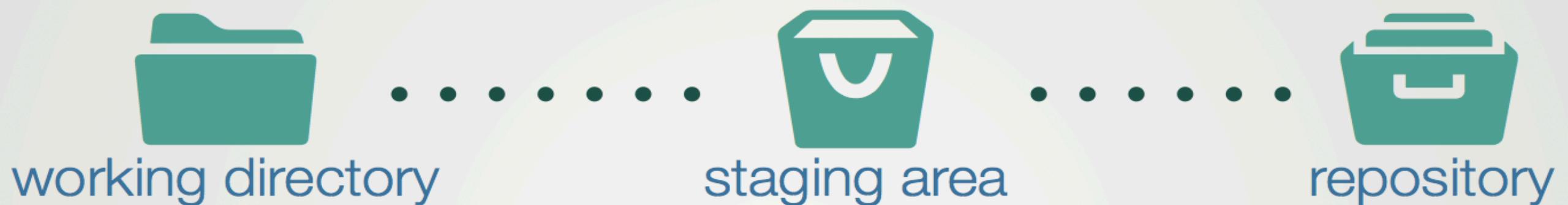
►基礎概念

►儲存流程

►.gitignore

Basic git in local

基礎概念 – 檔案狀態



檢視git的當前狀態

```
$ git status  
On branch master  
nothing to commit, working directory clean
```

儲存流程



加到暫存區

```
// From working directory to staging area  
$ git add <檔案>  
!  
// From staging area to working directory  
$ git rm --cached <檔案>
```

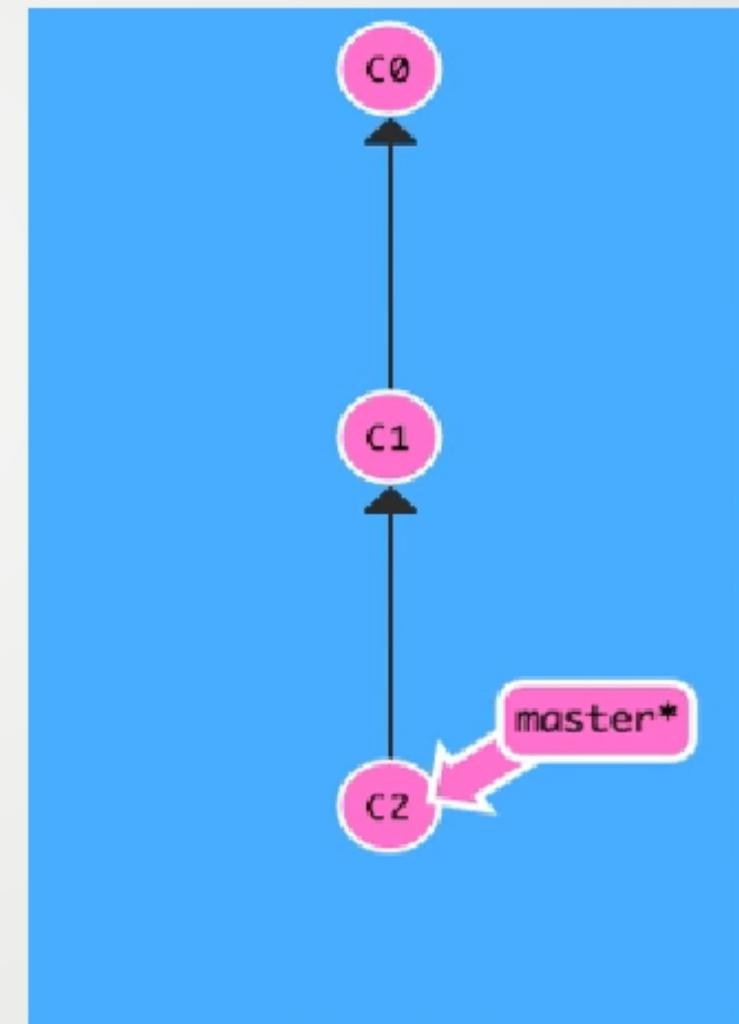
送進檔案庫

```
// commit to repo  
$ git commit <檔案>  
$ git commit -m “commit message”  
!
```

Take a snapshot

- Commit 會在目前的狀態，建立一個新的記錄點，之後可以回復到過去的紀錄點。

git checkout [revision] .



git log

```
$ git log
```

檢視過去提交的歷史紀錄，看看過去有哪些存檔

git diff

```
$ git diff
```

讓你知道哪些檔案被修改，並且比較修改了什麼內容

git rm

```
$ git rm <檔案>
```

刪除檔案，同時從 repo 的歷史紀錄移除

.gitignore

- 有些檔案永遠不想 add
 - 編輯器的暫存檔
 - C/C++ 的 binary 檔、物件檔
- 把你不要被 add 的檔案，寫進 .gitignore
 - 可以使用 *，表示全部檔案
 - 用正規表示式的語法

情境小幫手

不小心把 code 弄壞了，我想全部重來，那要怎麼回到上一個存檔點的樣子？

```
$ git reset --hard
```

清除所有與最近一次 commit 不同的修改

情境小幫手

只有一個檔案，想改回之前的狀態，要怎麼做？

```
$ git checkout -- <檔案>
```

將修改過的檔案回復到最近一次 commit 的狀態

- 遠端概念
- 設定連線
- Push and Pull

git with remote repository

基礎概念



基礎概念



Remote Repository

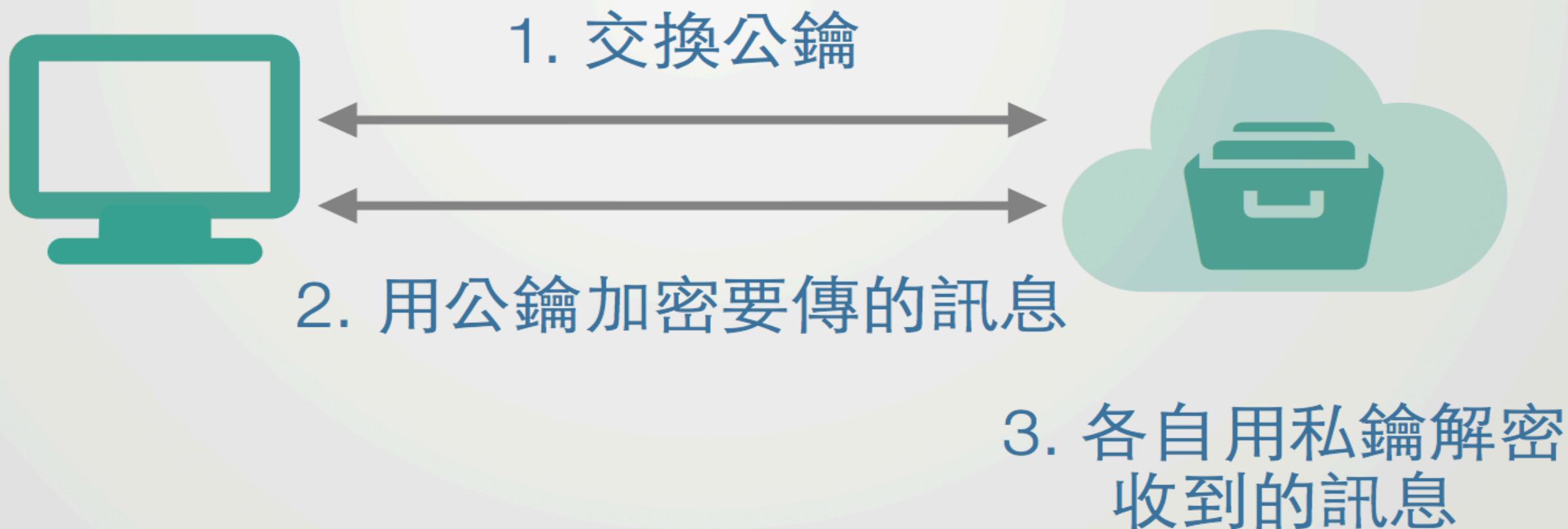


- Cooperate with others, private used or open source
- Issue tracking, Wiki feature
- Free education discount for .edu email users
- In this course, we use github to submit homework

SSH: SECURE SHELL

每台電腦會有：

公鑰 (public key) – 用於加密
私鑰 (private key) – 用於解密



Setup ssh key with github

Reference: <https://help.github.com/articles/generating-ssh-keys>

設定連線

1.  

連結遠端 repo

現有 repo

remote repo
2.  

下載遠端 repo

建立repo

remote repo

在github建立新repo

The screenshot shows a GitHub repository page for 'aliensunmin / test'. The top navigation bar includes 'Unwatch' (1), 'Star' (0), and other repository details. On the left, there's a sidebar with links to 'Code', 'Issues' (0), 'Pull Requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area has three sections: 'Quick setup — if you've done this kind of thing before', '...or create a new repository on the command line', and '...or push an existing repository from the command line'. Each section contains code snippets and copy/paste icons.

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

Set up in Desktop or **HTTP** **SSH** `git@github.com:aliensunmin/test.git`

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
touch README.md  
git init  
git add README.md  
git commit -m "first commit"  
git remote add origin git@github.com:aliensunmin/test.git  
git push -u origin master
```

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

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

連結遠端repo

```
$ git remote add origin git@HOSTPATH
```

設定你的本地 repo 跟遠端連結

複製遠端repo

```
$ git clone git@HOSTPATH:directory name
```

從遠端複製一份 repo 回來，在當前的資料夾建立新資料夾 <directory name>，並自動連結。

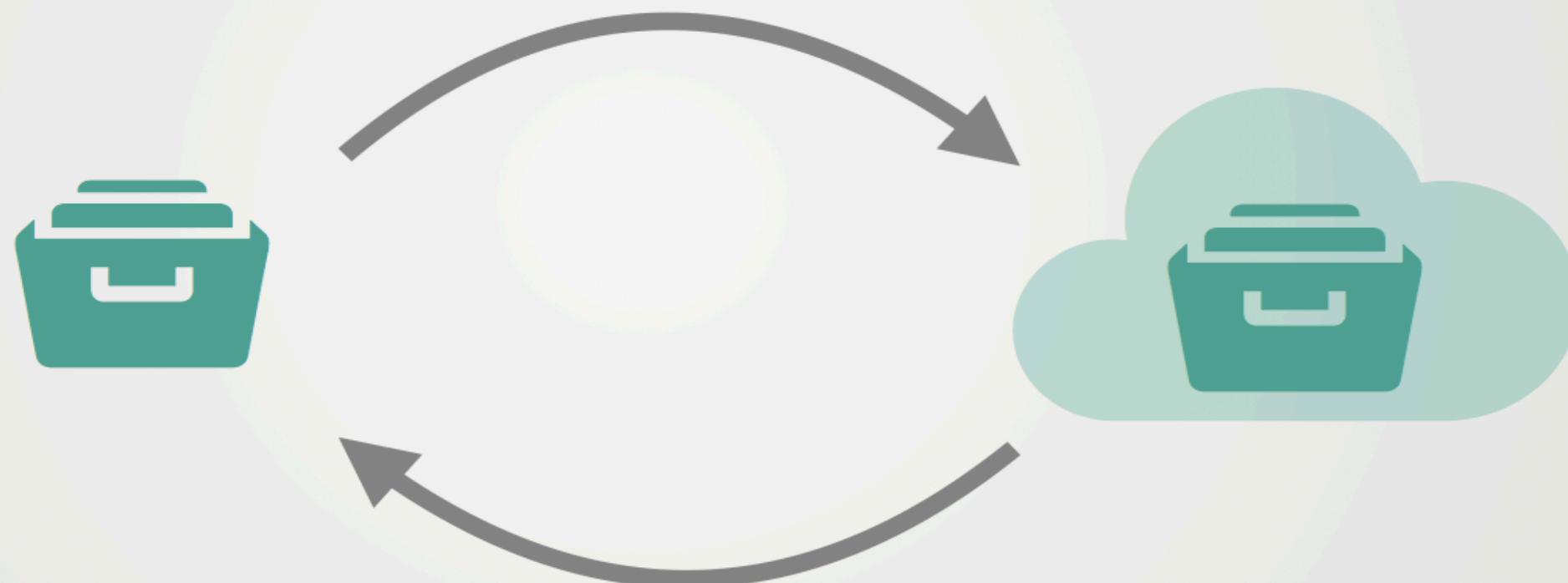
git remote

```
$ git remote -v
```

檢視你遠端 repo 的資訊

PUSH AND PULL

push: 將本地版本推上遠端



pull: 從遠端拉回本地並且合併到本地端

git push

```
$ git push -u origin master
```

本地端預設(分支)叫做 master，我們把本地 master
推送到遠端

git pull

```
$ git pull origin master
```

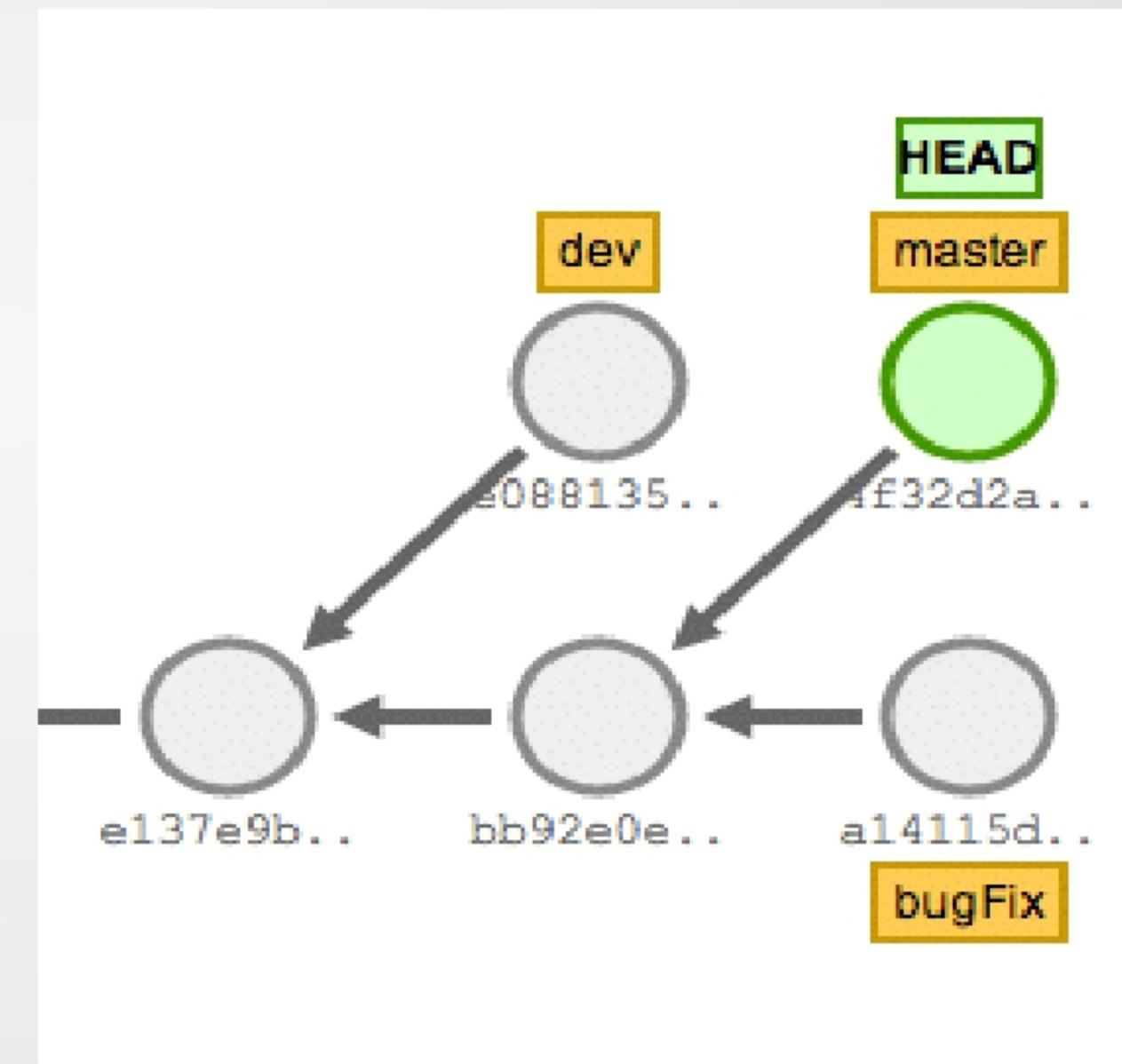
- ▶ 分支概念
- ▶ 切換分支
- ▶ 合併分支
- ▶ 與他人合作

branch



Branch

- 建立一條新的歷史紀錄
- 多種版本獨立開發
- 分工合作時，幫助大家不會互相衝突。
- 最終可以合併在一起



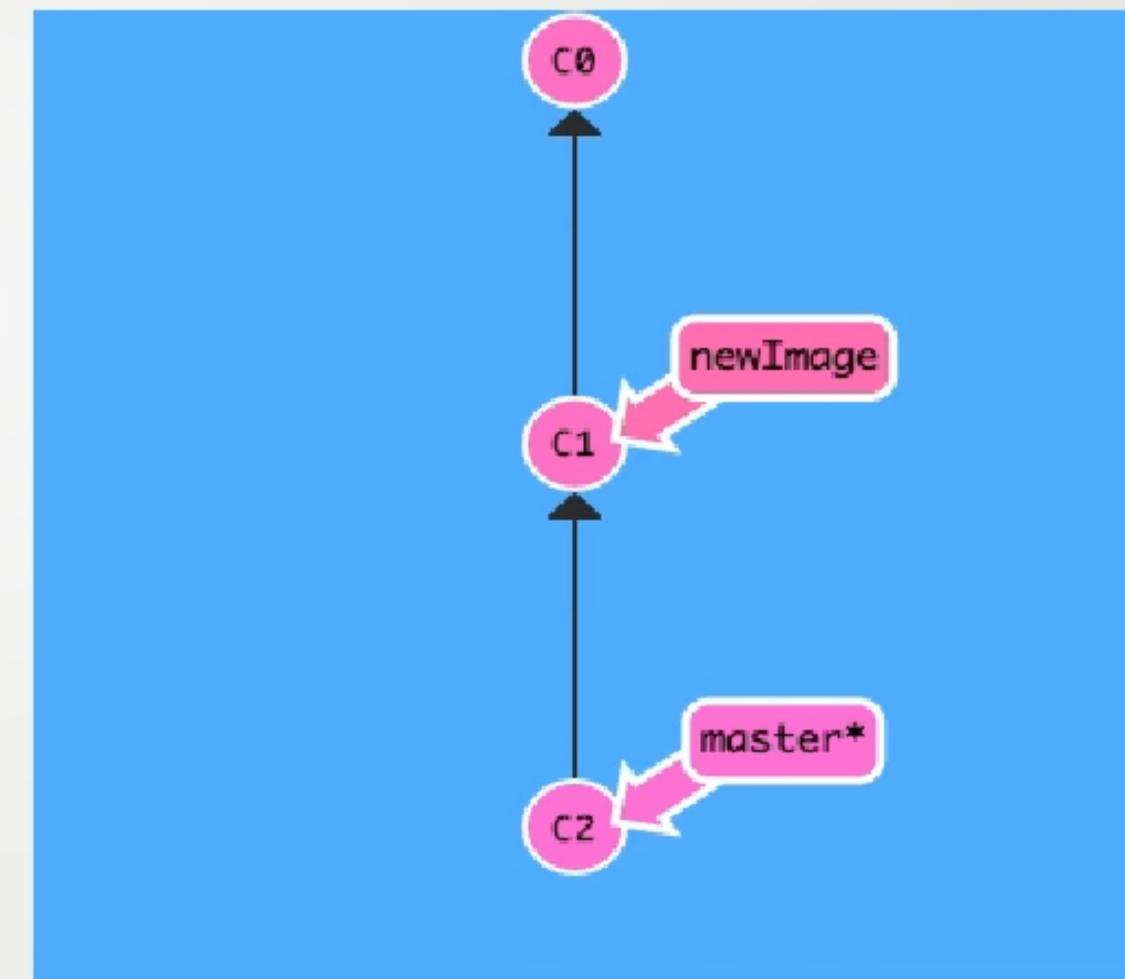
git branch

```
$ git branch <new branch>
```

從當前分支上，新增一條名為 `<new branch>`
的分支

What branch look like

- 生成新 branch 時，並不會切換到新 branch。



Manipulate branch

```
#list local branch  
$ git branch  
  
#list local and remote branch  
$ git branch -a  
  
#delete the branch  
$ git branch -D <branch>
```

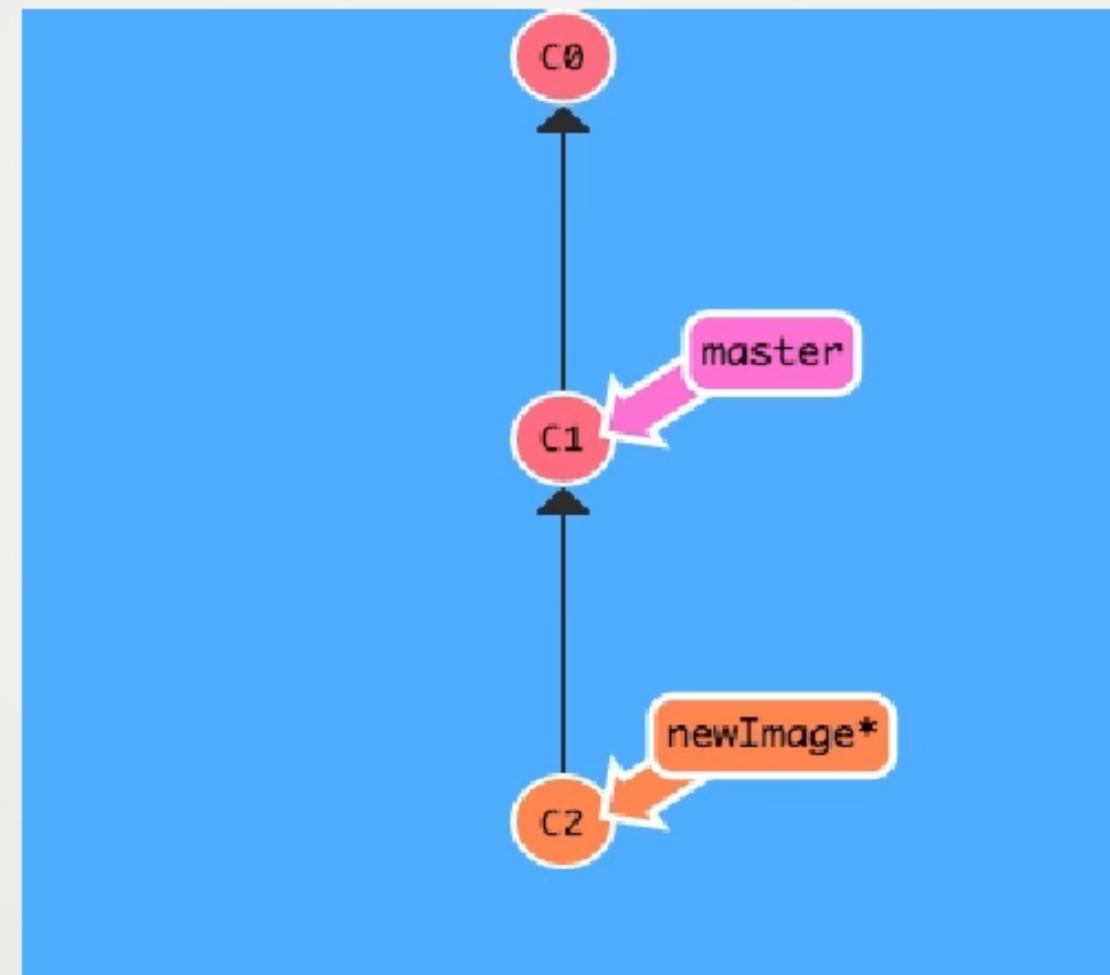
Change your branch

```
$ git checkout <branch>
```

New and change branch simultaneously

```
$ git checkout -b <branch>
```

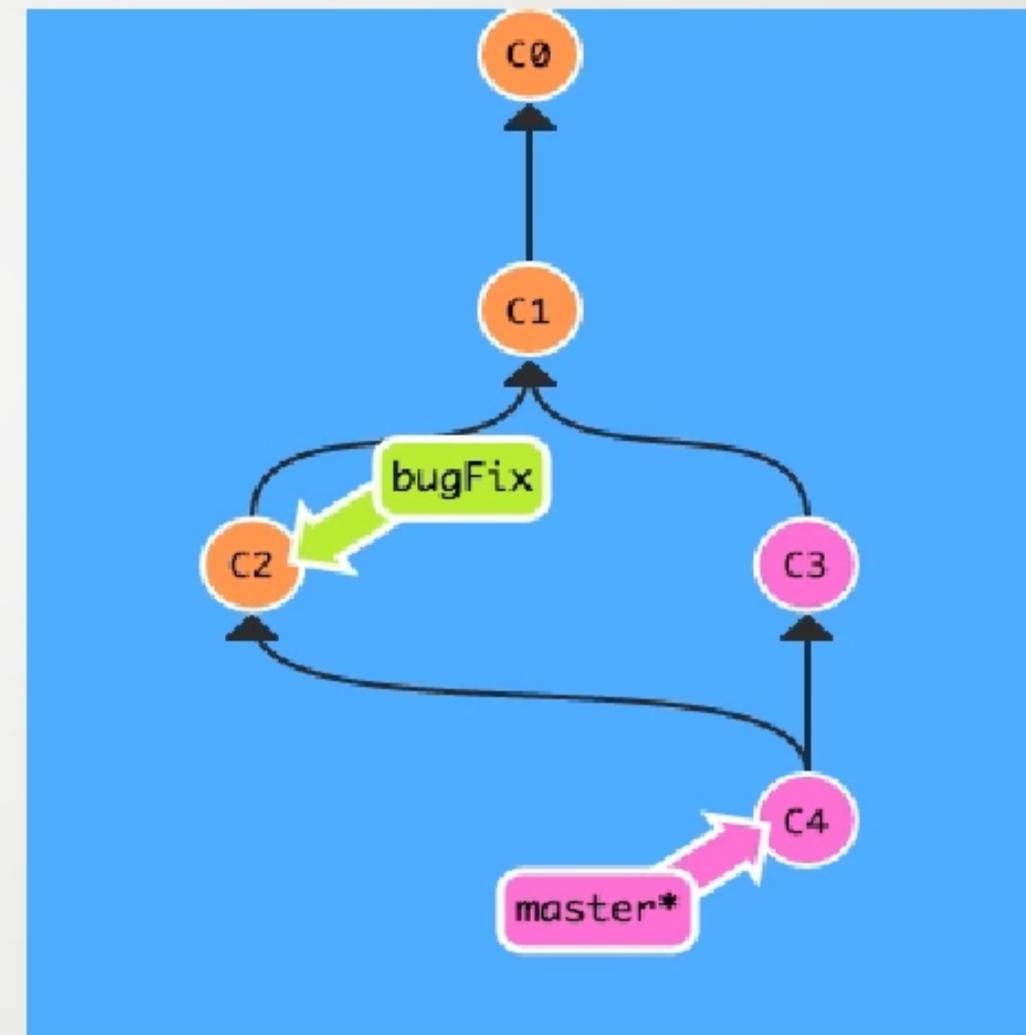
What branch look like



merge

merge

- 當兩個人分別開發在不同分支時，最終會合併為完整版。
- 改動不同檔案、同一檔案不同區域
 - auto merge
- 改到同一行，GG。
 - conflict
 - 人工處理



合併分支支

```
$ git merge <branch>
```

把另外一支 <branch> 合併到自己身上來。

當發生生 conflict

- 首先 git status ，了解狀況。
- 衝突的檔案會標示兩個版本的資訊，手動合併存檔
- git add 改好的檔案
- git commit

conflict

```
$ git status
On branch master
You have unmerged paths.

  (fix conflicts and run "git commit")

!
Unmerged paths:
  (use "git add <file>..." to mark resolution)

!
      both modified: index.html
```

衝突的地方會標示出來

```
<<<<< HEAD  
<div id="footer">contact :  
email.support@github.com</div>  
  
=====  
<div id="footer">  
    please contact us at  
    support@github.com  
</div>  
>>>>> bugFix
```

最後手動commit

```
$ git commit -am "Merge bugFix"
```

情境小幫手

Merge 時發生 conflict，想取消 merge

```
$ git reset --hard
```

將修改過的檔案回復到最近一次 commit 的狀態

情境小幫手

Merge 時發生生 conflict，想把 conflict 的地方，直接用某支 branch 的

```
$ git checkout --ours <file>  
$ git checkout --theirs <file>
```

ours 表示衝突部分使用目前分支的，theirs 則反過來。

多人協作

git 合作方方式

- 有一支 branch 放完整可以動的程式，比比如說 master 或是 dev。
- 開發功能時，大家從主線分支出來，獨立開發。
- 開發完後，合併回去主線。
- 自己的分支自己合。
- 或是 git fetch origin ，把所有遠端分支搬回家，再自己合併。

To-Do C: Homework 0

- <https://github.com/NTHU-EE-CV-2014-Fall/homework0>

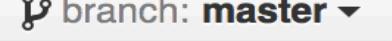
 sunnthy / **homework0**
forked from NTHU-EE-CV-2014-Fall/homework0

Description

Short description of this repository Website for this repository (optional) or [Cancel](#)

 5 commits  1 branch  0 releases  1 contributor

  **homework0** / + 

This branch is even with NTHU-EE-CV-2014-Fall:master  

add  aliensunmin authored on Jul 28 latest commit 84fd04a6cb 

 README.md add a month ago

 **README.md**

Homework 0 **homework0**

This homework is simply used to check if all students know how to work with git and github (no credit). At the sametime, it also helps me to know you a little bit better :)

Website

  0    

HTTPS clone URL
<https://github.com/> 

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#). 