

Git教學



Arthur Pai

軟體需求



- ❖ 請先申請 GitHub 帳號：<https://github.com/>
- ❖ 安裝下列軟體及註冊帳號，安裝過程全部用預設
 - ❖ Git: <https://git-scm.com/download/win>
 - ❖ Sourcetree: <https://www.sourcetreeapp.com/>
 - ❖ Visual Studio Code (可選其他文字編輯軟體):
<https://code.visualstudio.com/>

成績計算



- ❖ 作業1：50%
- ❖ 作業2：50%

Git 是什麼？



What is Git?

版本控制系統 (VCS)

Version Control System



❖ 什麼是版本控制？

- ❖ 版本控制是一種記錄一個或若干文件內容變化，以便將來查閱特定版本修訂情況的系統
- ❖ 可以將某個文件回溯到之前的狀態，甚至將整個專案都回退到過去某個時間點的狀態
- ❖ 可以比較文件的變化細節，查出最後是誰修改了哪個地方，從而找出導致怪異問題出現的原因
- ❖ 是誰在何時造成了某個功能缺陷(Bug)等等！
- ❖ 多人協同開發方式下，每個人開發一個小功能

Screen Demo



energym-server - [~/_Repository/Personal/IFP/energym/energym-server] - SmartGit 21.2.3

Pull Sync Push Git-Flow Merge Rebase Commit Stage Index Editor Unstage Discard Save Stash Apply Stash Add Tag Blame Investigate Log Selection Main Review

Repositories

- FantasyImagesUtility (master)
- vue-auth-plugin (master)
- GMEP14
- IFP
 - CityExpo* (master)
 - cloud-gate-live* (master)
 - HistoryMuseum
 - IFPlus_SFU* (main)
 - IFPlus_SFU_Dev_Tool* (main)
 - RenderStreaming* (feature/2.4.0-exp.4)
 - RenderStreamingSFU* (feature/2.4.0-exp.4)
 - UnityRenderStreaming* (develop)
 - AskMorris (master)
 - cloud-gate-live-laravel (master)
 - cloudGate-online-course (master)
 - docker-elk (master)
 - energym-server (develop)**
 - Esteelauder (master)
 - KayBook (master)
- energym-server (develop)

Branches

- HEAD (8e45ee3b)
- master = origin
- Features (1)
 - docker = origin
- Local Branches (1)
 - develop = origin
- origin (3) - git@gitlab.ifp.edu:ifp/energym-server.git
 - develop
 - master
 - feature/docker
- Stashes (1)
- Recyclable Commits

Graph

origin\develop → origin\master 取消驗證悠遊卡格式 2019/12/30 PM 12:30
increase band lock time to 90 2019/12/10 PM 12:00
add remove zasy card feature 2019/12/09 PM 10:43
increase band lock time 2019/12/09 PM 10:16
set server ip 2019/11/26 PM 07:25
Finish docker 2019/11/26 PM 06:52
origin\feature/docker update readme 2019/10/24 PM 09:23
update db backup script 2019/10/24 PM 08:15
fix band_lock_time 2019/10/24 PM 07:38
set db restart policy 2019/10/24 PM 05:03
fix band code length of session 2019/10/24 PM 04:54
update config 2019/10/23 PM 05:20
fix url error 2019/10/23 PM 05:20
update README 2019/10/23 PM 02:31
update README 2019/09/14 PM 07:52
Merge branch 'master' into feature/docker 2019/09/14 PM 07:31
Update fastpass api 2019/09/12 PM 02:19
Disable reservation feature, add new fastpass model and ... 2019/09/12 PM 02:03
設定 Docker 部署機制 2019/09/14 PM 07:30
add database sedding, and signature default images 2019/09/12 PM 12:16
Merge remote-tracking branch 'origin/master' into feature... 2019/09/12 AM 11:34
Fix available bands' query method of band_info 2019/09/11 PM 05:51
update README 2019/09/12 AM 11:33

Commit

Commit 8e45ee3b by Artuhn 2019/12/30 PM 12:30
parent dce3b31e

Files 1,868 files hidden

Name	Modified	Renamed	Path	Relative Directory
views.py	Modified			django/backstage

Changes of views.py (Modified)

```
458     return redirect(request.path_info)
459   else:
460     messages.error(request, '無法變更廣告：請求錯誤')
461     logger.error("Request for creating or updating ad is invalid")
462     return redirect(request.path_info)
463   else:
464     return Response(content, status=status.HTTP_405_METHOD_NOT_ALLOWED)
465
466 @api_view(['POST'])
467 def ajax_manipulate_ads(request):
468   out = {
469     'ret': 'ok',
470     'msg': '',
471     'data': '',
472   }
473
474   if request.method == 'POST':
475     @api_view(['POST'])
476     def ajax_manipulate_ads(request):
477       out = {
478         'ret': 'ok',
479         'msg': '',
480         'data': '',
481       }
482
483       if request.method == 'POST':
484         # Process POST data
485         ...
486
487       else:
488         # Handle other methods
489         ...
490
491       return JsonResponse(out)
492
493   else:
494     # Handle other methods
495     ...
496
497   return JsonResponse(out)
```

Unified Side by Side | Compact

源起



- ❖ Linus Torvalds
(Linux 的作者)
- ❖ 為了管理 Linux Kernel
程式碼而開發
- ❖ 改善當時(2005)，版本控
制系統的缺點



Version Control System



❖ Distributed Version Control Systems

❖ [Git: Github, Gitlab, Bitbucket](#)

❖ [Mercurial](#)

❖ Centralized Version Control System

❖ CVS([Centralized Version Control](#))

❖ SVN([Subversion](#))

❖ [Perforce](#)

採用 Git 的專案



- ❖ [Visual Studio Code](#)
- ❖ Android
- ❖ Linux Kernel
- ❖ Debian, Fedora, openSUSE
- ❖ Git
- ❖ Gnome
- ❖ GIMP
- ❖ GTK+
- ❖ jQuery
- ❖ Perl
- ❖ Qt
- ❖ Ruby on Rails
- ❖ VLC
- ❖ Wine
- ❖ X.org Server
- ❖ x264

Git 的好處



Why Git

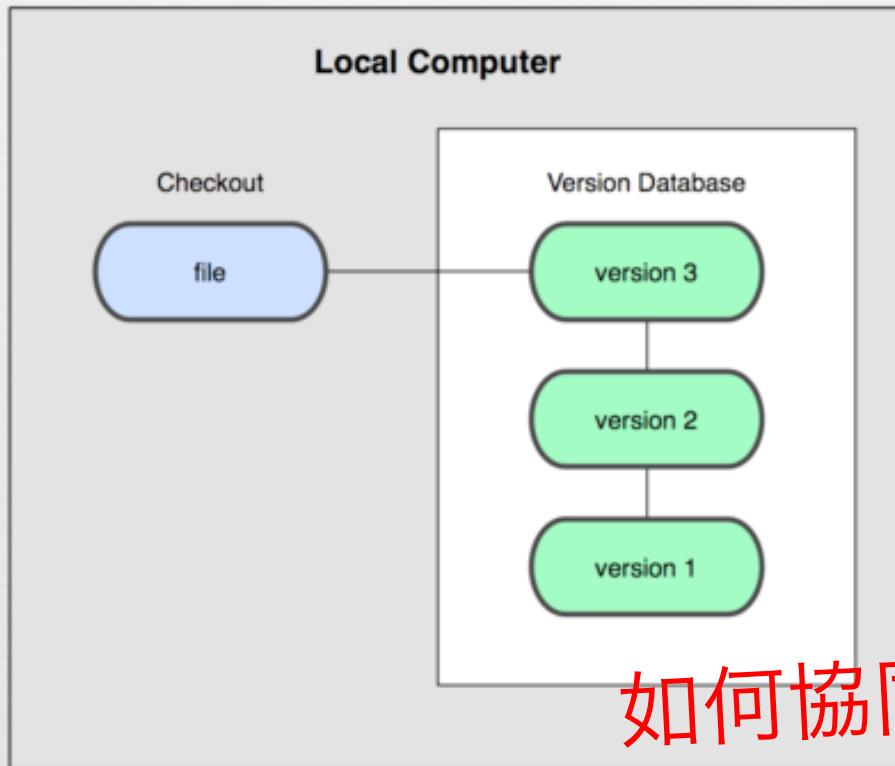


優於其他管理系統

- Local
- Centralized Version Control System
- Distributed Version Control Systems

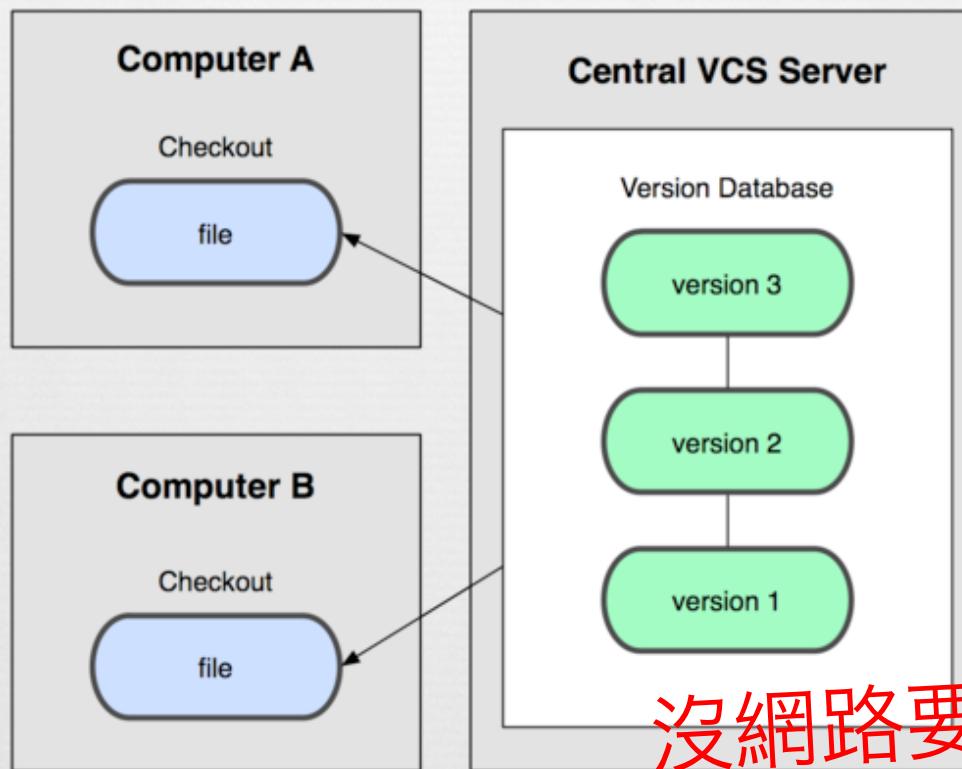
本機管理方式

Local



中央管理

Centralized Version Control System(CVCS)

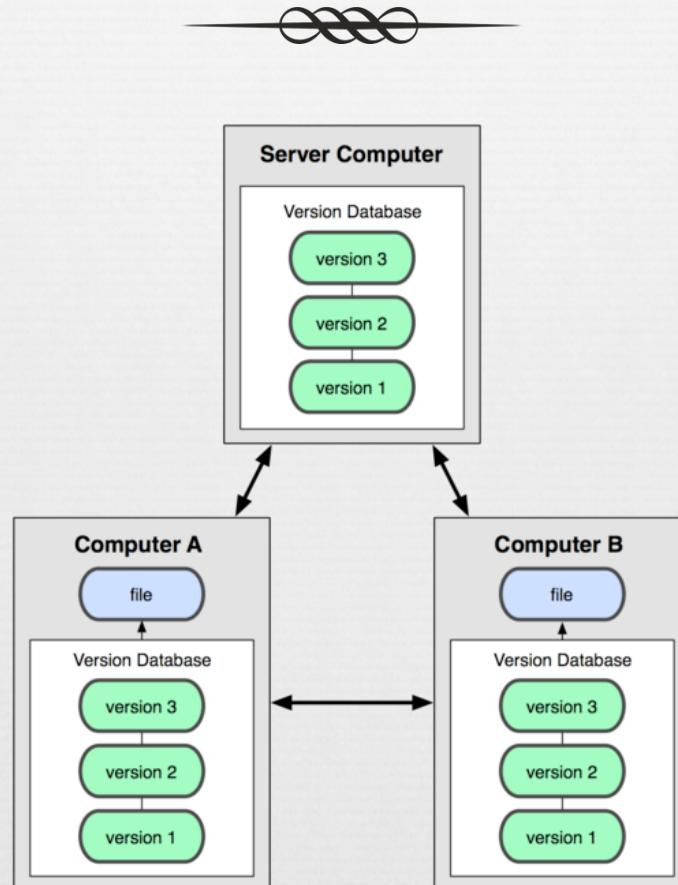


沒網路要怎辦？

Server 掛了怎辦？

SVN(Subversion)
CVS

分散式 Distributed Version Control Systems (DVCS)



適合超大量資料



- ❖ 操作速度超快
- ❖ 儲存的檔案很小
- ❖ 連 Linux Kernel 都 hold 得住 (2700 萬行)

沒有網路也能工作



- ❖ Git once, coding anywhere!
- ❖ Branch 開不用錢
- ❖ 看 log 超快

Branch 不用錢



- ❖ 開 branch 幾乎零成本
- ❖ 開 branch 改東西，不怕改爛主軸
- ❖ 切換 branch 很簡單
- ❖ Git 的 merge 很聰明

支援協定



- ❖ **file://**
本機檔案、例如用 Dropbox 架 repo
- ❖ **https://**
簡單、效率低、可突破防火牆
- ❖ **git://**
安全、效率好、匿名訪問不可、適合寫入

休息_{10分鐘}



跨平台工具



Git Software

工程師的浪漫

Command Line Interface (CLI)



```
- 88e72e8 [docker-compose]
The plugin now checks for the `docker-compose` command instead of trying whether `docker compose` is a valid command. This means that if the old command is still installed it will be used instead. To use `docker compose`, uninstall any old copies of `docker-compose`.

Features:
- bf30396 [aws] Adds the login option for AWS SSO (#9921)
- 46e6334 [branch] Show mercurial bookmarks if used (#9948)
- f0f792f [cli] Add `omz version` command
- 8e5f3db [dotnet] Add alias for `dotnet build` command (#10435)
- 58478d0 [git] Add alias for rebasing to origin/main-branch (#10445)
- 2b379ec [kn] Add plugin for `kn` completion (#8927)
- 60b89cd [ssh-agent] Add `quiet` option to silence plugin (#9659)
- c66fc00 [updater] Show command to update when update skipped (#10465)
- 452ddff [xcode] Support `.swiftpm` as project file in `xc` (#10434)

Bug fixes:
- ff09151 [bgnotify] Avoid permission prompts by checking frontmost app ID (#10318)
- 98b4801 [bundler] Use BUNDLE_JOBS in `b` to avoid config file change
- 88e72e8 [docker-compose] Check for old command instead of calling `docker` (#10409)
- fb12e41 [install] Fix backslash in `print` when showing logo (#10422)
- 0314604 [lib] Don't error if `INSIDE_EMACS` is not defined (#10443)
- b60b3f1 [osx] Deprecate `osx` plugin without symlink (#10428)
- 2b96b7c [updater] Stop update if `$ZSH` is not a git repository (#10448)

Other changes:
- 15fd9c8 [bundler] Style: Simplify `bundled_commands` array operations
- 2c06852 [dirhistory] Style: Remove use of `eval` completely

You can see the changelog with `omz changelog`
```



Hooray! Oh My Zsh has been updated!

To keep up with the latest news and updates, follow us on Twitter: @ohmyzsh
Want to get involved in the community? Join our Discord: [Discord server](#)
Get your Oh My Zsh swag at: [Planet Argon Shop](#)



The screenshot shows the SourceTree application interface. On the left is a sidebar with icons for Commit, Pull, Push, Branch, Merge, Shelf, Show in Finder, Terminal, and Settings. The main area has tabs for WORKSPACE, File status, History (which is selected), Search, BRANCHES, BOOKMARKS, TAGS, REMOTES, SHELVED, and SUBREPOSITORIES. The central part displays a commit history graph on the left and a detailed table of commits on the right.

Graph	Commit	Author	Description	Date
	b7358c7	Rahul Chhab...	[origin/master] [origin/HEAD] Removing ol...	Mar 3, 2016, 11:...
	bdb8bef	Rahul Chhab...	Merged in update-google-verification (pull request #14)	Feb 18, 2016, 1:3...
	dfe975d	Tyler Tadej...	[origin/update-google-verification] Update google verificati...	Feb 11, 2016, 2:2...
	3bc3290	Tyler Tadej...	Replace outdated Atlassian logo in footer with base-64 en...	Feb 11, 2016, 2:1...
	dba47f9	Tyler Tadej...	Add gitignore	Feb 11, 2016, 1:3...
	ff67b45	Mike Minns...	Updated Mac min-spec to 10.10	Feb 15, 2016, 11:...
	72d32a8	Michael Min...	Merged in hero_images (pull request #13)	Feb 15, 2016, 10:...
	246c4ff	Joel Unger...	[origin/hero_images] [hero_images] Used Tinypng to c...	Feb 11, 2016, 3:3...
	9d9438c	Joel Unger...	Replacing hero images with new version of SourceTree	Feb 9, 2016, 2:59...
	ce75b63	Michael Min...	Merged in bug/date-https (pull request #12)	Feb 15, 2016, 10:...
	85367bb	Patrick Tho...	[origin/bug/date-https] fixed date and https errors	Jan 7, 2016, 12:2...
	4f9b557	Joel Unger...	New Favicon	Feb 8, 2016, 3:55...
	384e6d5	Rahul Chhab...	[origin/search-console-access] search console google ver...	Feb 3, 2016, 2:09...
	6fa47a9	Mike Minns...	updated to move supported version to OSX 10.9+	Dec 15, 2015, 2:0...
	8dd87bb	Mike Minns...	remove extra , when a line is skipped due to empty server	Nov 23, 2015, 2:2...
	faa195e	Mike Minns...	Skip records with empty server/user id as gas rejects them	Nov 23, 2015, 2:1...
	0cdfe96	Mike Minns...	corrected paths after merge	Nov 23, 2015, 2:0...
	051ab1b	Mike Minns...	corrected column counting	Nov 23, 2015, 1:5...
	a723bc2	Mike Minns...	Merge branch 'au2gex'	Nov 23, 2015, 1:5...
	65fd580	Mike Minns...	deal with invalid instanceids	Nov 23, 2015, 1:5...
	500a892	Michael Min...	Merged in au2gex (pull request #11)	Nov 23, 2015, 1:0...

SourceTree



<https://www.sourcetreeapp.com/>

Free

The screenshot shows the SmartGit interface for a Joomla CMS repository. The main window displays a commit graph for the '4.0-dev' branch. A specific commit, '4[com_joomlaupdate] untranslated warning strings (#34804)', is selected. The commit details show it was signed by Nicola Galgano on July 16, 2021, at 18:02. The commit message is '[4] [com_joomlaupdate] untranslated warning strings (#34804)'. The commit has a purple 'NG' icon and a child commit '3071cd68'. The commit history shows several other commits related to Joomla updates and bug fixes. Below the commit graph, the 'Changes of HtmlView.php (Modified)' section shows a side-by-side comparison of the file's code. The left pane shows the original code, and the right pane shows the modified code. The code changes involve updating language files and handling layout settings. The interface includes standard Git operations like Pull, Sync, Push, and Commit, along with specific SmartGit features like Graph, Stage, Index Editor, Unstage, Discard, Delete, Save Stash, Apply Stash, Blame, and Investigate.

SmartGit



<https://www.synteko.com/smartgit/>

Cross-Platform, free for non-commercial purpose.

設定 Git 環境



Setup Git

安裝 Git



❖ Git 下載網址

❖ <https://git-scm.com/>

❖ 安裝過程直接套用預設的安裝設定

The image contains two side-by-side screenshots of the Git website (<https://git-scm.com/>) on a Windows 11 browser.

Left Screenshot: Shows the main homepage with a brief introduction, a diagram of a branching workflow, and sections for About, Documentation, Downloads, and Community. A red arrow points to the "Download for Windows" button on a computer monitor icon.

Right Screenshot: Shows the "Download for Windows" page. It includes links for 32-bit and 64-bit Windows Setup, Portable versions, and a link to the Pro Git book. A red box highlights the "64-bit Git for Windows Setup" link. Below it, there's information about the current source code release (version 2.36.1) and instructions for using Winget.

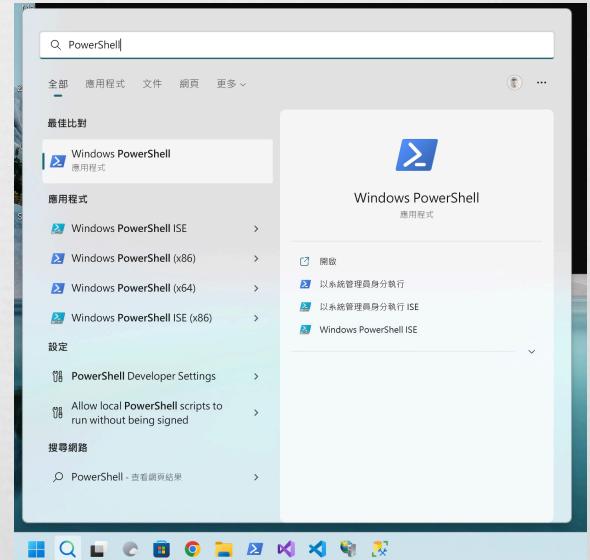
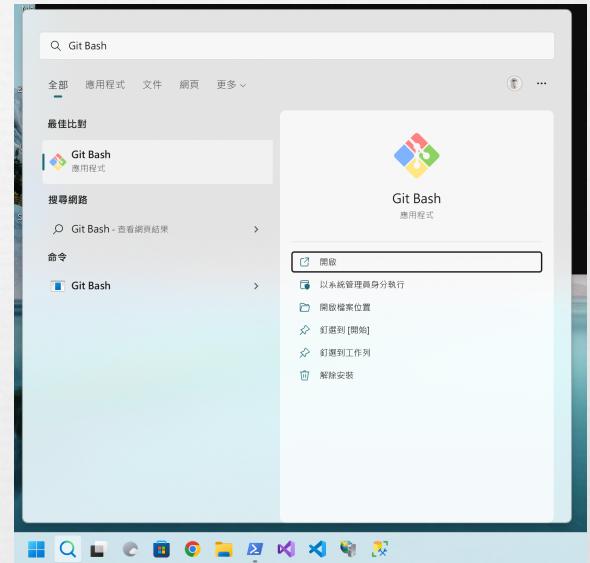
安裝 Git



❖ 測試是否安裝成功

- ❖ 開啟Git bash
或是 Windows PowerShell
- ❖ 輸入 Git version

```
PS C:\Users\Arthur> git version
git version 2.36.1.windows.1
PS C:\Users\Arthur>
```



練習 5分鐘



安裝 SourceTree



- ❖ SourceTree 下載網址
 - ❖ <https://www.sourcetreeapp.com/>
- ❖ 安裝過程直接套用預設的安裝設定

練習 5分鐘



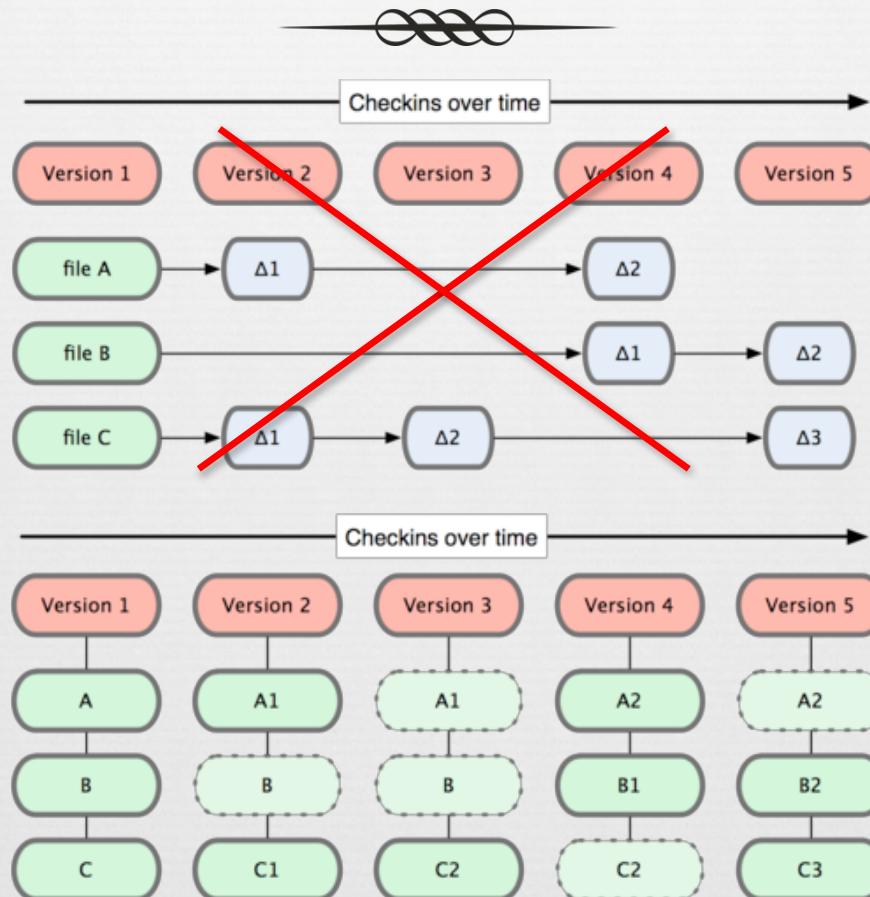
Git 基礎知識



Git Overview

https://iissnan.com/progit/html/zh-tw/ch1_3.html

記錄檔案快照，而不是差異的部份



Git 基礎



- ❖ 大部份的操作皆可在本地端完成
- ❖ 處理速度快
- ❖ 瀏覽歷史快
- ❖ 可離線工作
- ❖ Git 通常只增加資料

Hash



❖ Git 能檢查完整性

❖ 使用查核碼 (checksum) 檢索物件(Commit、BLOB)

❖ 24b9da6552252987aa493b52f8696cd6d3b00373

❖ Hash 算法：

❖ 把任意長度的資料透過演算法轉換增固定長度的字串

❖ MD5(128bit)：不安全，文件校驗

❖ SHA1(160bit)：不安全，Git 採用

❖ 40 * 4bit(16進位)

❖ SHA256(256bit)：安全，Docker image

❖ SHA512(512bit)：安全

Hash



❖ SHA1

❖ 16進位：長度40

24b9da6552252987aa493b52f8696cd6d3b00373

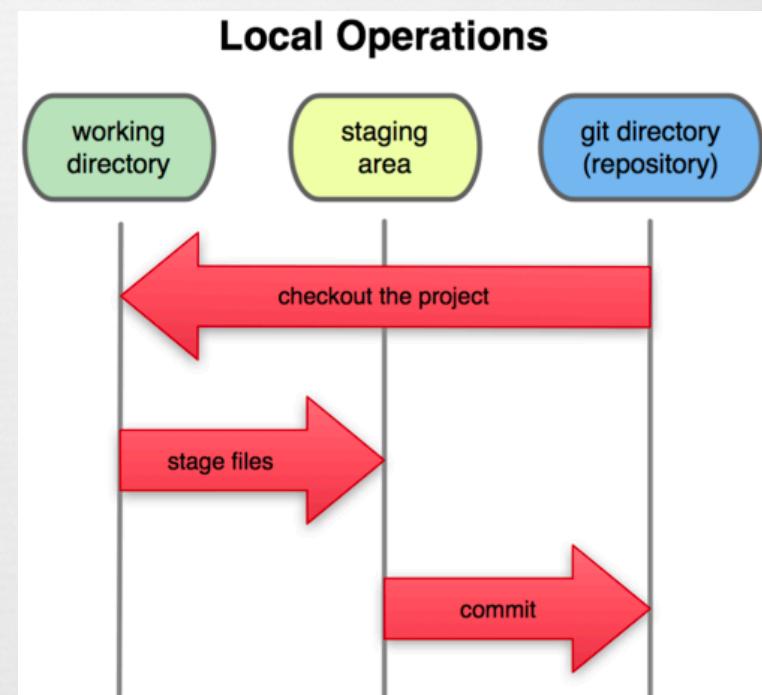
❖ 2進位：長度160

10010010111001110110100110010101010010001001
01001010011000011110101010010010011101101
0100101111100001101001011011001101011011001
1101100000000001101110011

處理流程

❖ 基本Git工作流程大致如下：

- ❖ 修改工作區(working directory)的檔案
- ❖ 暫存區(stage)，將檔案的快照新增到暫存區域(staging area)
- ❖ 做提交(commit)的動作，這會讓存在暫存區的檔案快照永久地儲存在Git目錄(repository)



作業1



作業1



1. 註冊 Github 帳號
2. 將 eMail 及 Github 帳號 填寫到 Google 試算表
3. 接受邀請(Join/Accept Invitation)
4. 確認可以看到 Git-Work Repository
5. 在 最後一個Commit 加入討論 (Comment)

作業1



ArthurPai / **git-test** Private

Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

ArthurPai Update README.md 50b4513 now 3 commits

.gitignore Initial commit 9 months ago

README.md Update README.md now

README.md

資展 Git 課程

- 學號：000
- 姓名：白華勝

About

No description, website, or topics provided.

Readme 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

作業1



Screenshot of a GitHub repository page for "ArthurPai/git-test".

The repository is private.

Watched by 1 user, Forked 0 times, Starred 0 times.

Code, Issues, Pull requests, Actions, Projects, Security, Insights tabs are present.

The main branch is selected.

Commits on Aug 31, 2022:

- Update README.md** (Verified) - ArthurPai committed 1 hour ago. This commit is highlighted with a red box.
- Create README.md** (Verified) - ArthurPai committed 1 hour ago.

Commits on Dec 1, 2021:

- Initial commit** (Verified) - ArthurPai committed on 1 Dec 2021.

Newer Older buttons are at the bottom.

作業1



ArthurPai / git-test (Private)

Code Issues Pull requests Actions Projects Security Insights

Update README.md

main ArthurPai committed 1 hour ago (Verified)

Showing 1 changed file with 3 additions and 0 deletions.

3 README.md @@ -1 +1,4 @@

1	1	# 資展 Git 課程
2	+	
3	+	- 學號 : 000
4	+	- 姓名 : 白華勝

1 comment on commit 50b4513

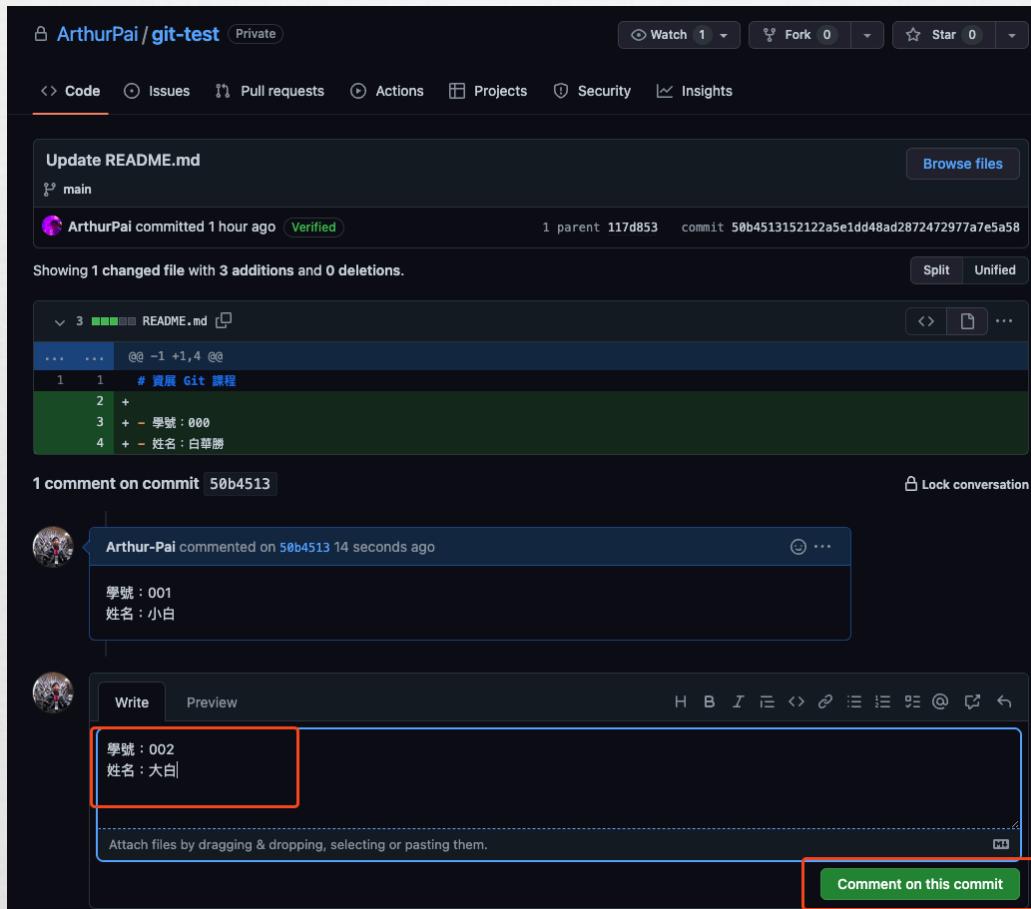
Arthur-Pai commented on 50b4513 14 seconds ago

學號 : 001
姓名 : 小白

Write Preview

學號 : 002
姓名 : 大白

Comment on this commit



Command Line

基礎操作



Command Line Basic

建議使用 **Git Bash**

目錄相關指令



❖ 目前目錄

`$ pwd`

❖ 列出目前目錄下的檔案及子目錄

`$ ls`

`$ ls -l`

❖ 切換目錄

`$ cd my-data`

`$ cd c:\my-data`

❖ 回上一層目錄

`$ cd ..`

練習 5分鐘



Git

基礎操作



Git Basic

設定 Config



❖ 設定用戶資訊

\$ *git config --global user.name "John Doe"*

\$ *git config --global user.email johndoe@example.com*

❖ 查看目前配置資訊

\$ *git config --global --list*

Git Repostory



❖ 初始化倉庫

❖ 在專案目錄下執行

\$ git init

❖ 初始後，會產生一個 .git 的子目錄，所有 git 資訊都在裡面。

❖ 複製倉庫 (Clone)

git clone git@github.com:ispan-umvr09/git-test.git my-git-test

練習

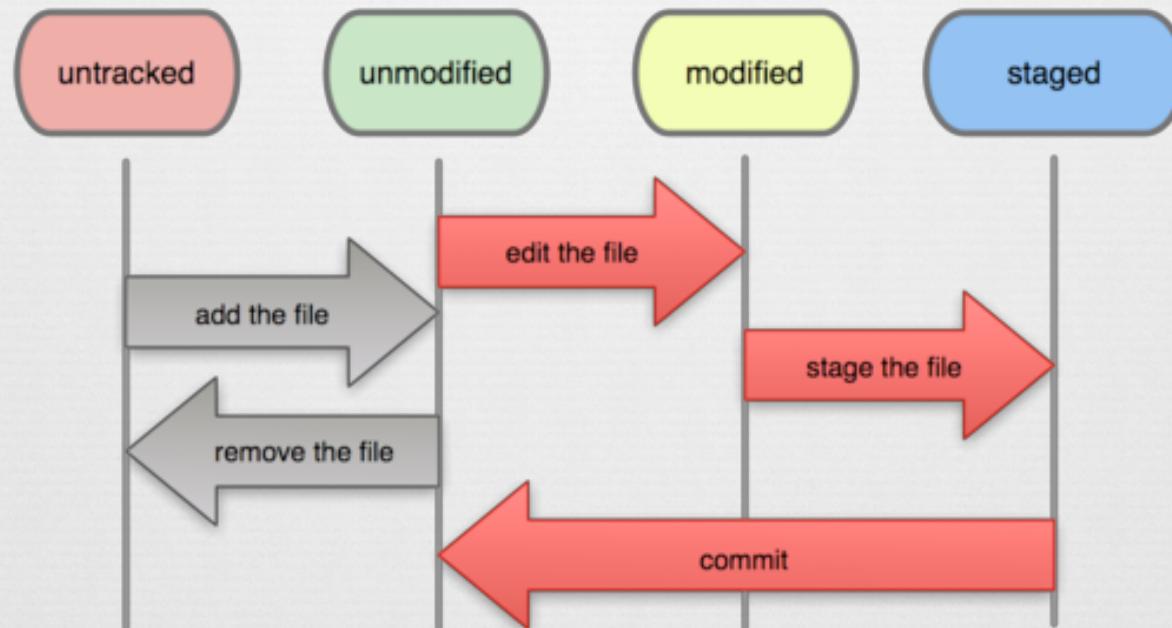
5分鐘



檔案狀態



File Status Lifecycle



<https://codecademy.launchacademy.com/learn-git/git-lifecycle>

Commit



❖ 檢查文件狀態

git status

❖ 跟蹤新文件

git add <file>

❖ 暫存已修改文件

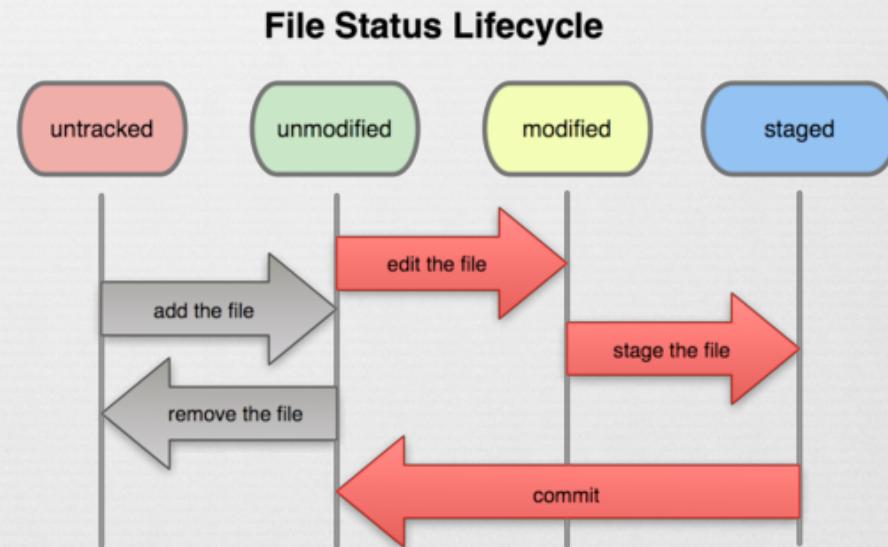
git add <file>

❖ 提交更新(*commit*)

git commit -m <註解>

❖ 跳過暫存區(*staging*)

git commit -a -m <註解>



練習 5分鐘



Diff



❖ 查看未暫存的修改

git diff

❖ 查看已暫存的修改

git diff –cached

練習 5分鐘



Commit History



❖ 查看提交歷史

git log

git log -1

❖ 查看提交詳細歷史

git log -p

git log -p -2

練習 5分鐘



Remove File



❖ 移除文件

git rm <file>

❖ 移動文件

git mv <current location> <new location>

撤消操作



❖ 修改最后一次提交

git commit --amend

❖ 取消已暫存的文件

git restore --staged <file>

❖ 取消對文件的修改(剛寫的程式會消失哦！！)

git restore <file>

❖ 取消最後一次提交

git reset master^

練習 5分鐘



忽略某些文件



❖ .gitignore 檔案

```
# 此为注释 - 将被 Git 忽略
# 忽略所有 .a 结尾的文件
*.a
# 但 lib.a 除外
!lib.a
# 仅仅忽略项目根目录下的 TODO 文件，不包括 subdir/TODO
/TODO
# 忽略 build/ 目录下的所有文件
build/
# 会忽略 doc/notes.txt 但不包括 doc/server/arch.txt
doc/*.txt
# ignore all .txt files in the doc/ directory
doc/**/*.txt
```

忽略規則



- ❖ 空目錄會被忽略
 - ❖ 習慣放個 .gitkeep 在裡頭避免忽略
- ❖ 用 .gitignore 排除不需要的檔案
 - ❖ <https://github.com/github/gitignore>
 - ❖ 放在 Working tree 的根目錄
 - ❖ 記得也要 commit
 - ❖ 能自動產生的東西都該排除
 - ❖ 全域設定放在 ~/.gitignore

練習

5分鐘



連結遠端 Git 伺服器



Connect Remote Git Server

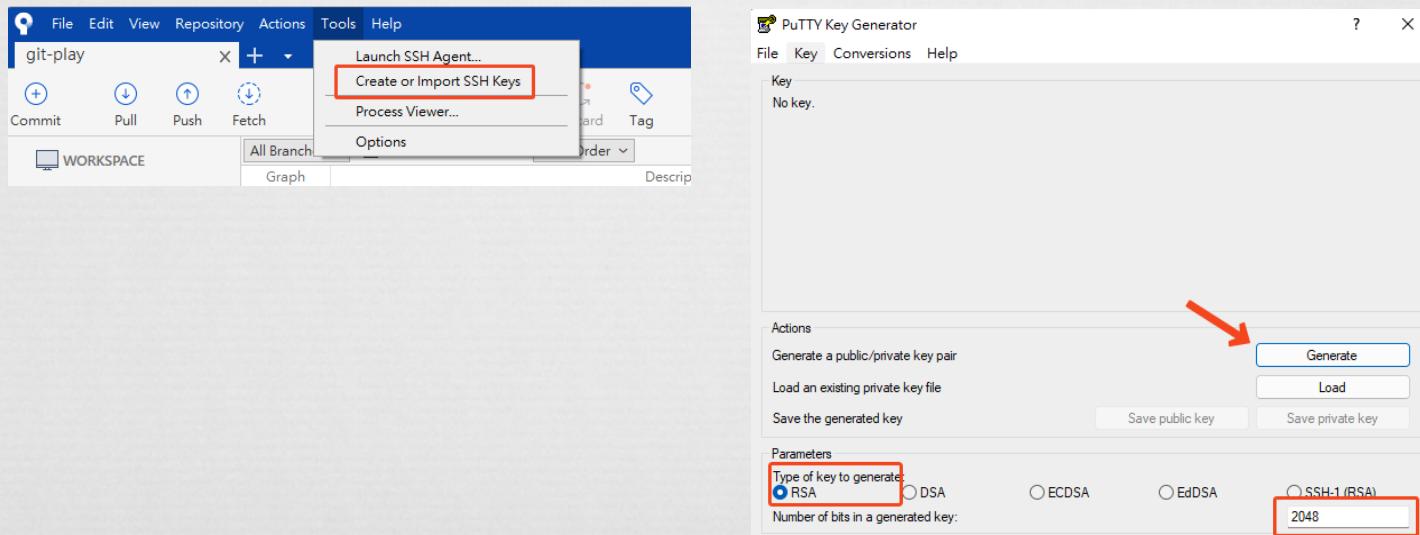
- 產生 SSH Keys
- 註冊 SSH Key 到 Github

產生 SSH Keys



開啟 Source Tree

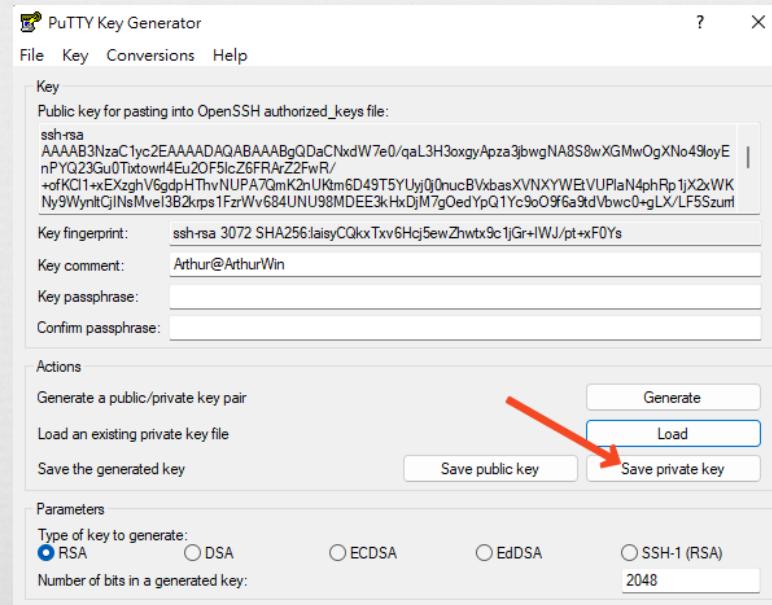
- ❖ 選單列 Tools > Create or Import SSH Keys
- ❖ 點擊「Generate」來產生Key
 - ❖ Parameters 預設值 RSA / 2048 bits 不需調整。



產生 SSH Keys



- ❖ 儲存 PuTTY Private Key (PPK File)
- ❖ 在你的用戶目錄下(C:/Users/[username])，建立.ssh目錄，用來儲存 private 及 public keys。
- ❖ 點擊「Save private key」將私鑰儲存到 .ssh 目錄。

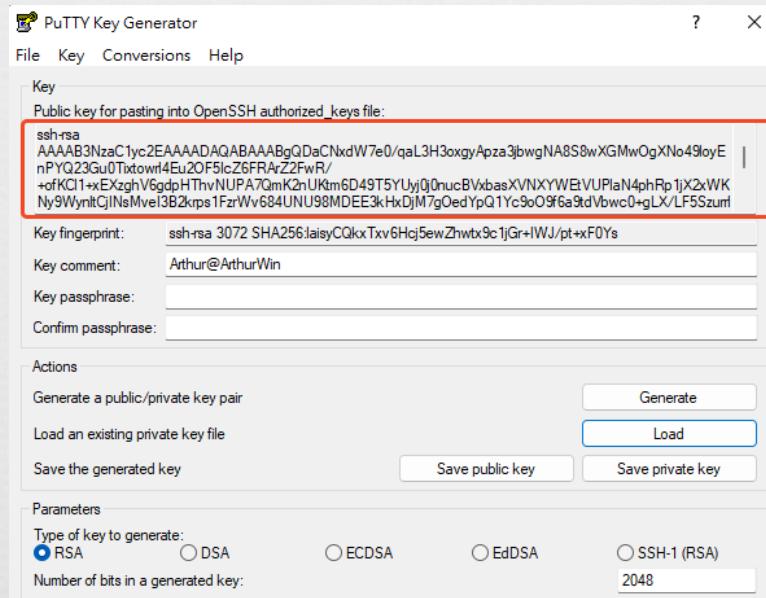


產生 SSH Keys



❖ 儲存 OpenSSH Public Key (id_rsa.pub File)

- ❖ 在.ssh目錄下建立一個 id_rsa.pub的文字檔，並且將 public key 區塊裡的內容複製進去。

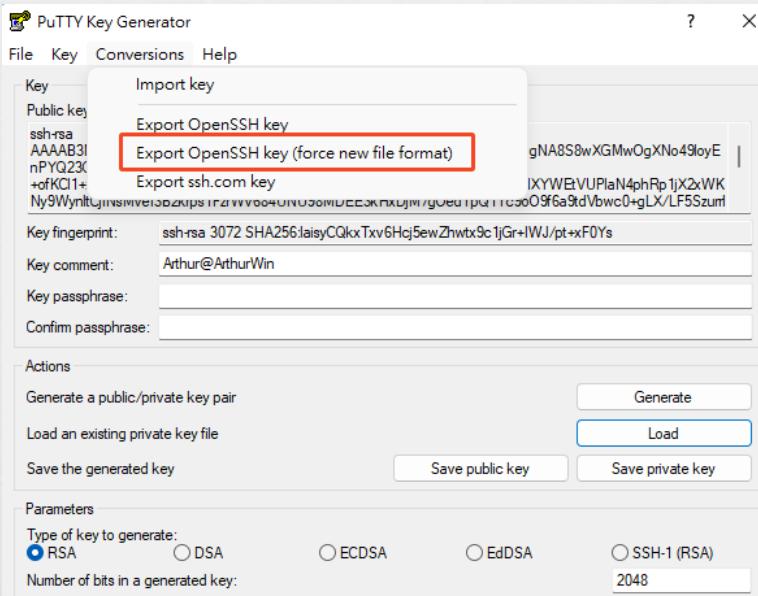


產生 SSH Keys



❖ 儲存 OpenSSH Private Key (id_rsa File)

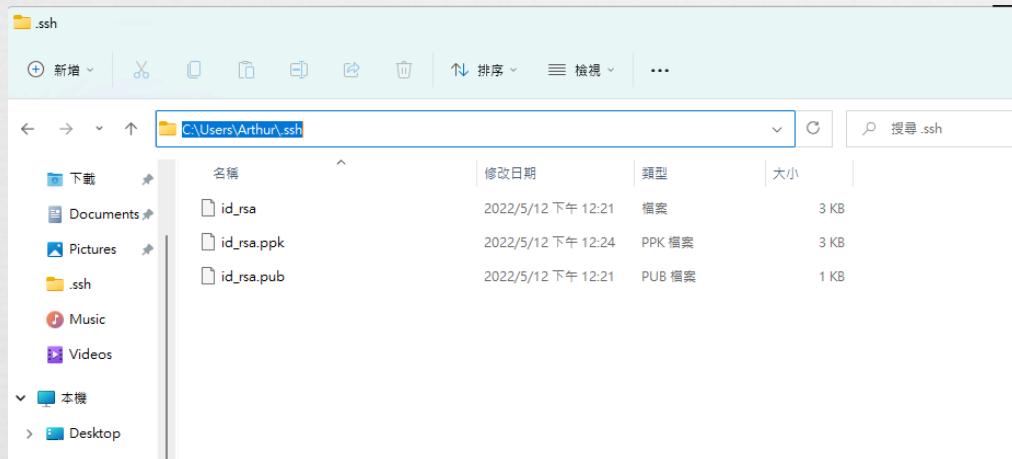
- ❖ 點擊 Conversions / Export OpenSSH key (force new file format)，將 OpenSSH 格式的私鑰儲存到 .ssh 目錄下，並且取名為 id_rsa



產生 SSH Keys



❖ 最後在 .ssh 目錄下，應該要有以下三個檔案



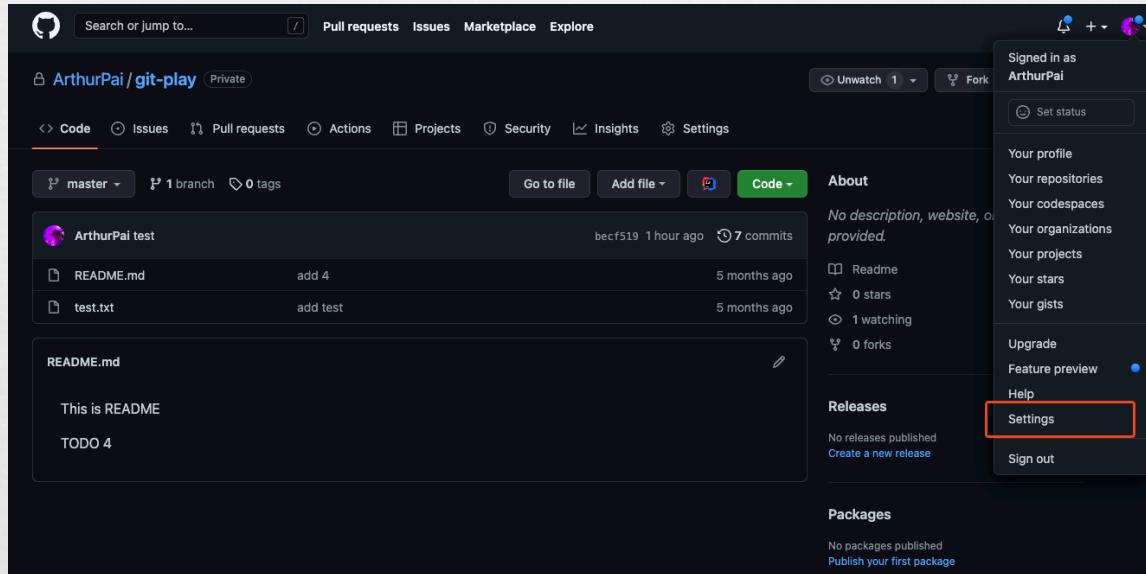
練習 10分鐘



註冊 SSH Key 到 Github



- ❖ 開啟 Github 頁面右上角的頭像，點選 Settings 開啟 設定頁面



註冊 SSH Key 到 Github



❖ 點擊「New SSH Key」按鈕來添加 SSH Key

The screenshot shows the GitHub user profile page for 'Artuhr'. The left sidebar has a dark theme with white text. The 'SSH and GPG keys' section is highlighted with a blue border. The main content area is titled 'SSH keys' and contains two entries:

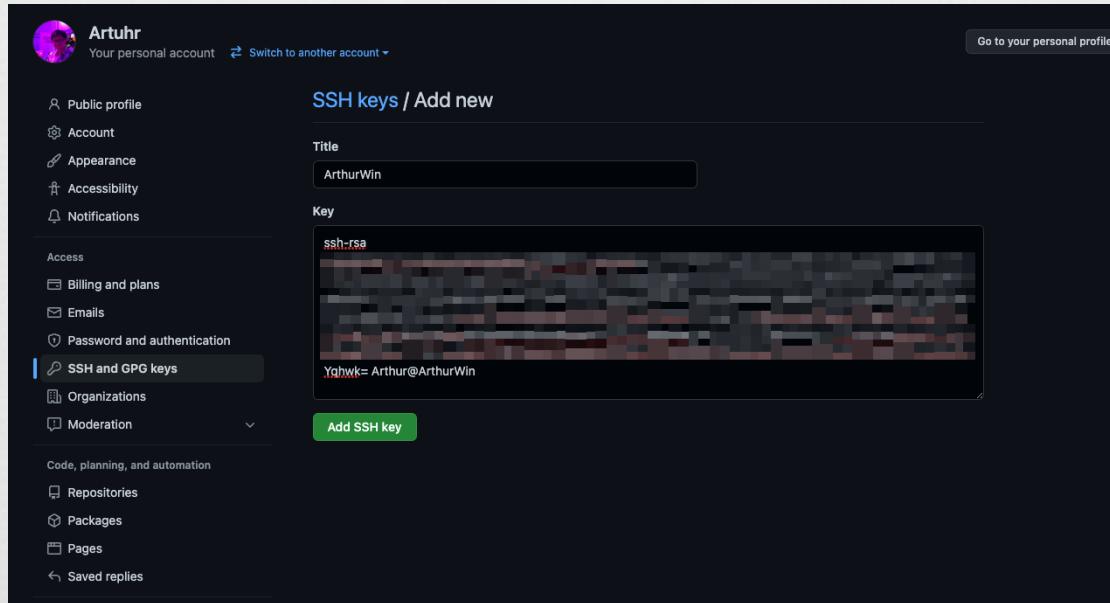
- Arthur**: Added on 22 Jun 2014. Last used within the last week — Read/write. Includes a green key icon and a 'Delete' button.
- ArthurWin**: Added on 12 May 2022. Last used within the last week — Read/write. Includes a green key icon and a 'Delete' button.

Below the list is a note: "Check out our guide to [generating SSH keys](#) or troubleshoot common SSH problems." At the bottom of the page, there's a 'Vigilant mode' toggle switch.

註冊 SSH Key 到 Github



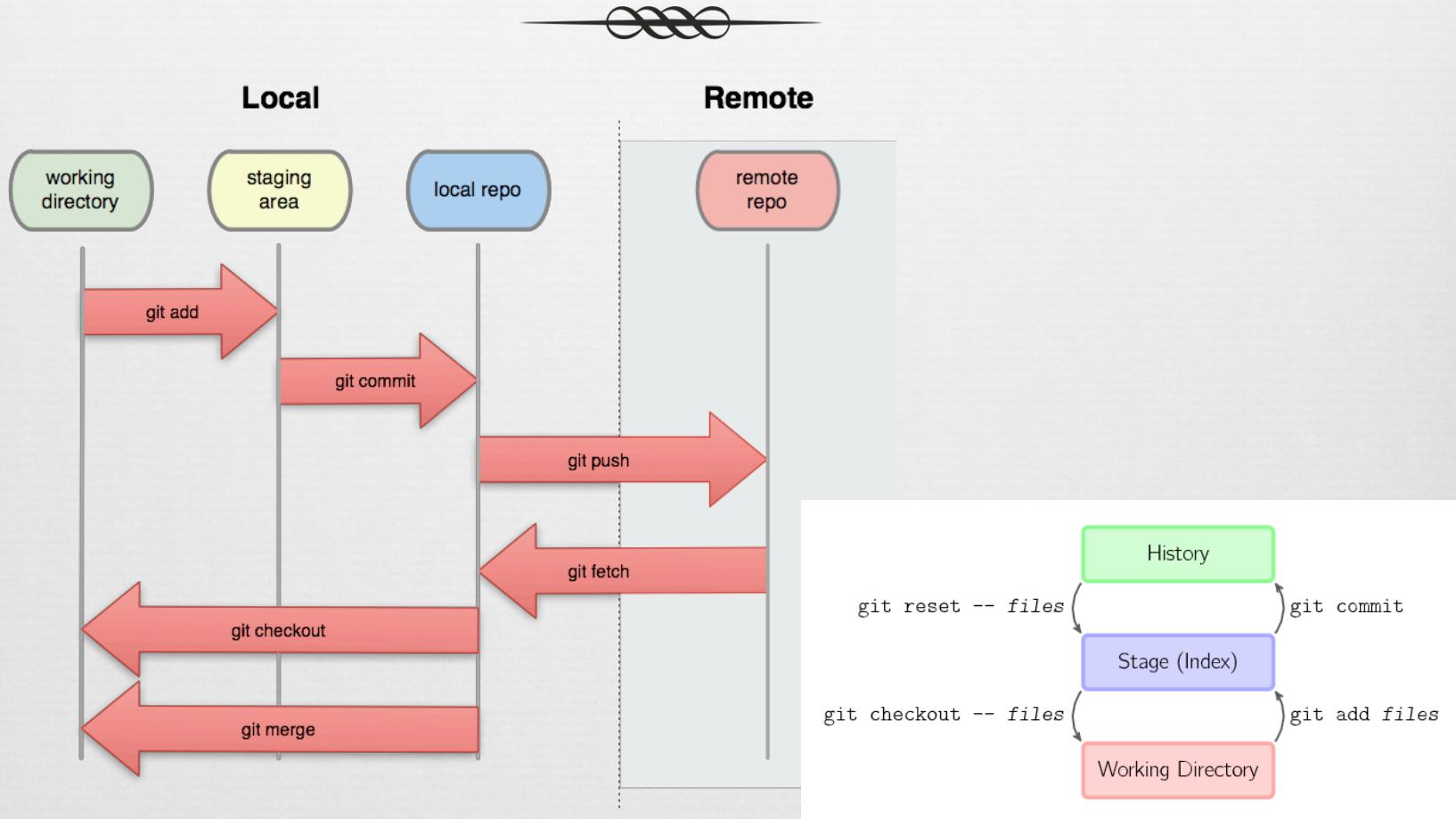
- ❖ 將之前 id_rsa.pub 的內容複製到 key 的欄位，Title 欄位可以隨意填寫，最後按下「Add SSH Key」將此 key 註冊到 Github



練習 5分鐘



流程(遠端)



連結遠端倉庫



❖ 查看目前已連接的遠端倉庫

git remote

❖ 添加遠端倉庫

git remote add <remote-name> <git url>

練習

5分鐘



推送 / 抓取 commits



❖ 推送資料到遠端倉庫

git push origin master

git push [remote-name] [branch-name]

❖ 從遠端倉庫抓取資料並且合併(fetch & merge)

git pull [remote-name] [branch-name]

❖ 從遠端倉庫只抓取資料(fetch) , 而不合併(merge)

git fetch [remote-name] [branch-name]

練習

5分鐘



其他



❖ 重命名遠端倉庫

git remote rename <remote-name> <new-name>

❖ 移除遠端倉庫

git remote rm <remote-name>

練習 5分鐘



Git

分支操作



Git Basic

建立 branch



- ❖ branch name 強烈建議用 [0-9A-Za-z_-.]
- ❖ 建立 branch
`git branch <branch_name>`
- ❖ 列出 branch
`git branch`

切換 branch



❖ 切換到其他 branch

```
git checkout <branch_name>
```

❖ 建立並切換到新支

```
git checkout -b <branch_name>
```

= [git branch <branch_name>](#) + [git checkout <branch_name>](#)

練習 5分鐘



建議工作規範



Proposal

git ignore



- ❖ 一定要設定 .gitignore 檔案
- ❖ 不要把不需要的檔案放進Git Repo來浪費空間以及你和隊友的生命！

時常同步



- ❖ 每天至少一次
- ❖ 有備份才安心
- ❖ 習慣 Git 的操作
- ❖ 降低 merge 困難度

關於檔案



- ❖ 避免不同平台產生亂碼
 - ❖ 檔名只用 ASCII 字元（少用中文）
 - ❖ 檔名不得使用可能非法字元，如 / ? < > * \ : | ”
 - ❖ 文字檔編碼一律用 UTF-8
- ❖ 避免不同軟體產生亂碼
 - ❖ 檔案內容一律用 UTF-8

詳細 commit



- ❖ 記得移除行尾空白
- ❖ 一次只 commit 一個主題
- ❖ commit message 寫清楚
 - ❖ 第一行：主旨，少於 50 字元
 - ❖ 第二行：空白
 - ❖ 第三行：詳情，每行少於 74 字元
- ❖ 推薦閱讀：
 - ❖ [Git Commit Message](#) (來源：[WadeHuang的學習迷航記](#))

愛用 branch



- ❖ 開 branch 不用錢，所以任何變動都開 branch
- ❖ 不要在 master 或 develop 修改程式

Git 工作流程

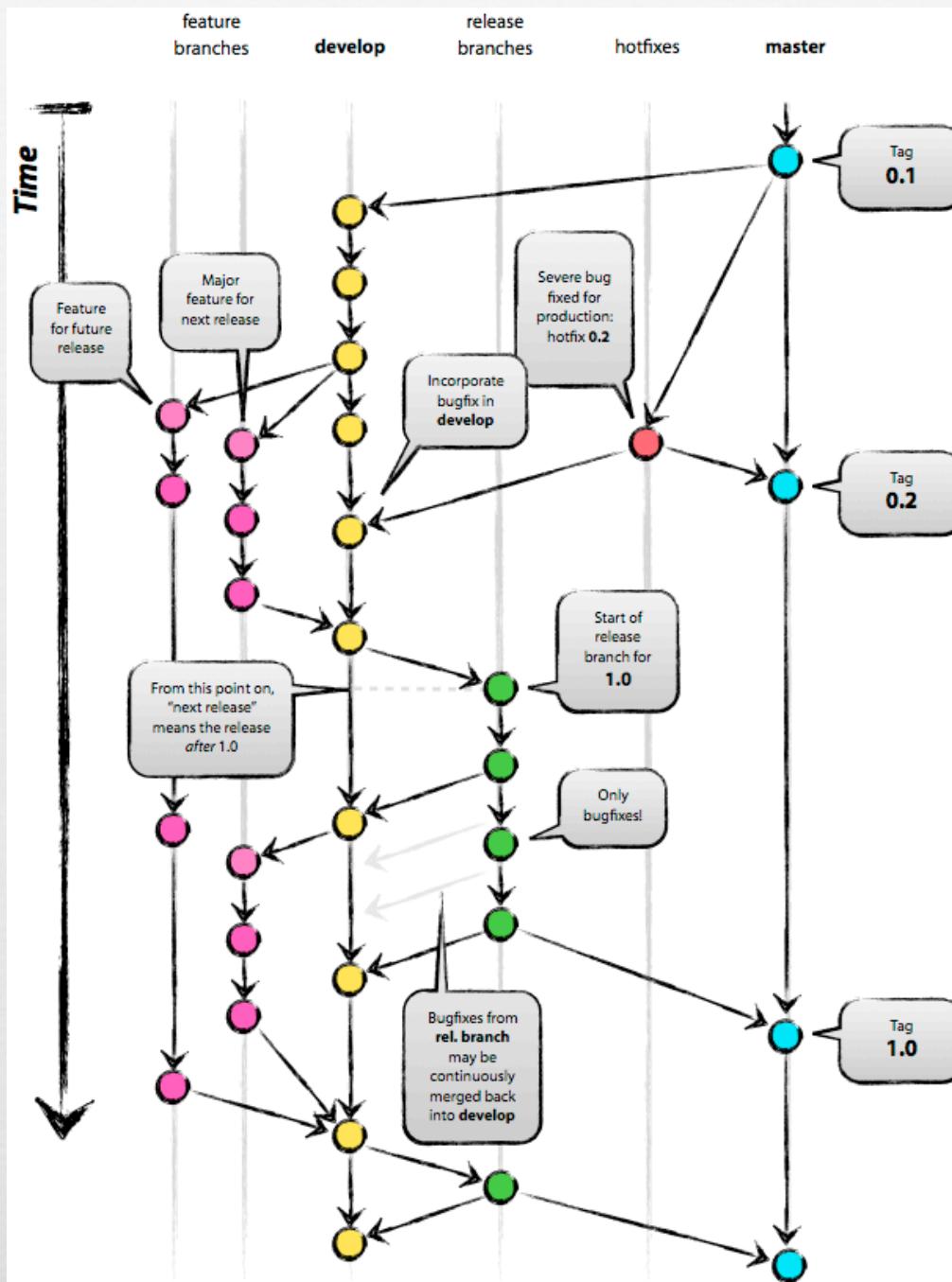


Work with Git



分支模型

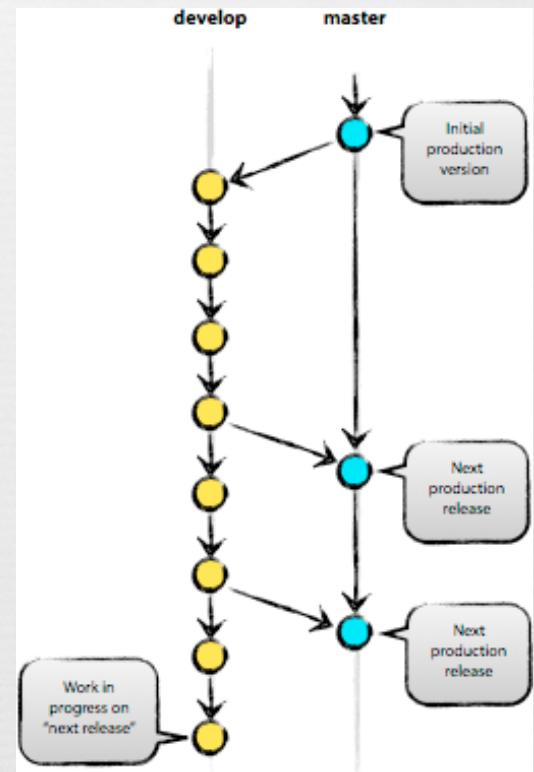
<https://nvie.com/posts/a-successful-git-branching-model/>



master



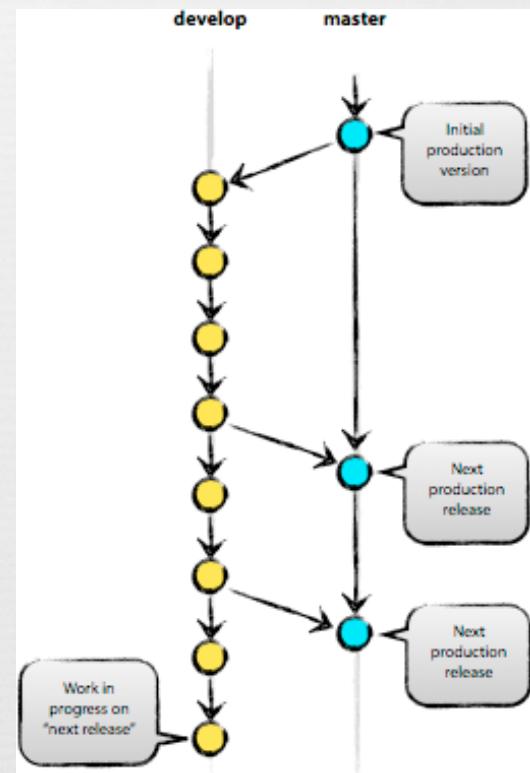
- ❖ 一直存在的分支
- ❖ 記錄每個上線的版本
- ❖ 只有在推出新版時才會變動
- ❖ 所謂的穩定版（stable version）



develop



- ❖ 一直存在的分支
- ❖ 必須總是 build 得過
- ❖ new feature 跟 enhancement 都在這
- ❖ 所謂的測試版（beta version）



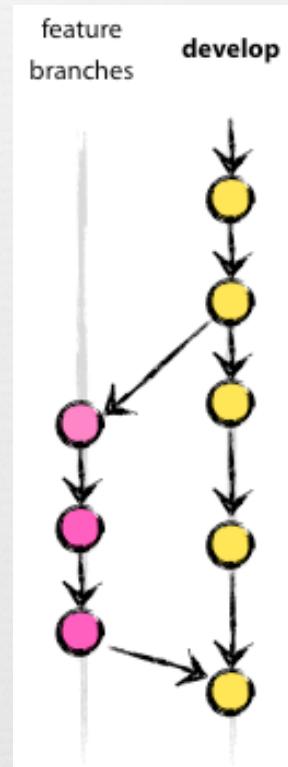
練習 10分鐘



feature



- ❖ 從 develop 分支出來
- ❖ 要 merge 回 develop
- ❖ 新增或修改功能時才會出現
- ❖ 功能完成就砍掉



release



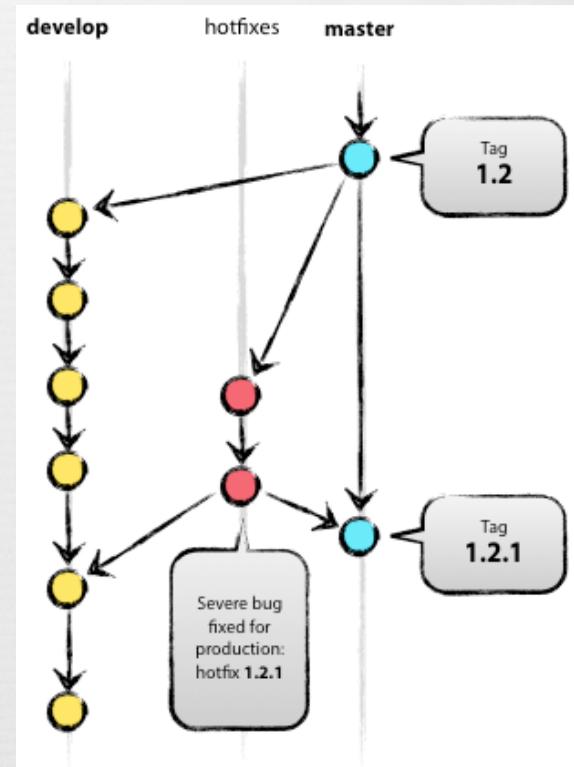
- ❖ 從 develop 分支出來
- ❖ 要 merge 回 master，並且上 tag
- ❖ 要 merge 回 develop
- ❖ 管理者決定要 release 新版時才會建立
- ❖ release 之後就砍掉
- ❖ 修改版本號、Release Note、ReadMe
- ❖ 最後測試並修正測試發現的 bug



hotfix



- ❖ 從 master 分支出來
- ❖ 要 merge 回 master，並且上 tag
- ❖ 要 merge 回 develop 或 release
- ❖ 緊急修正問題專用
- ❖ 修正之後就砍掉
- ❖ 修改版本號、Release Note、ReadMe



總結

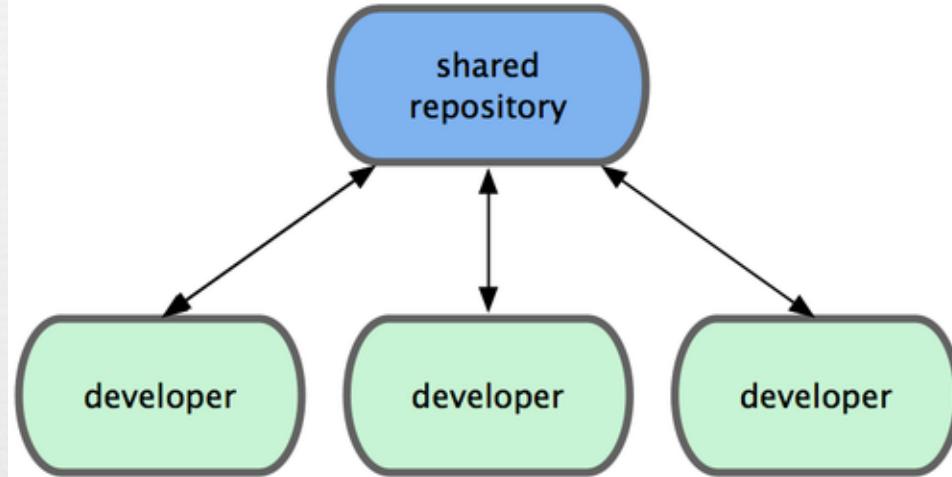


- ❖ 不同 branch 有不同用途，不得混用
- ❖ 任何更動都要開 branch
- ❖ 除了 master 跟 develop 之外，完成任務就砍掉 branch



工作流程

https://iissnan.com/progit/html/zh/ch5_1.html

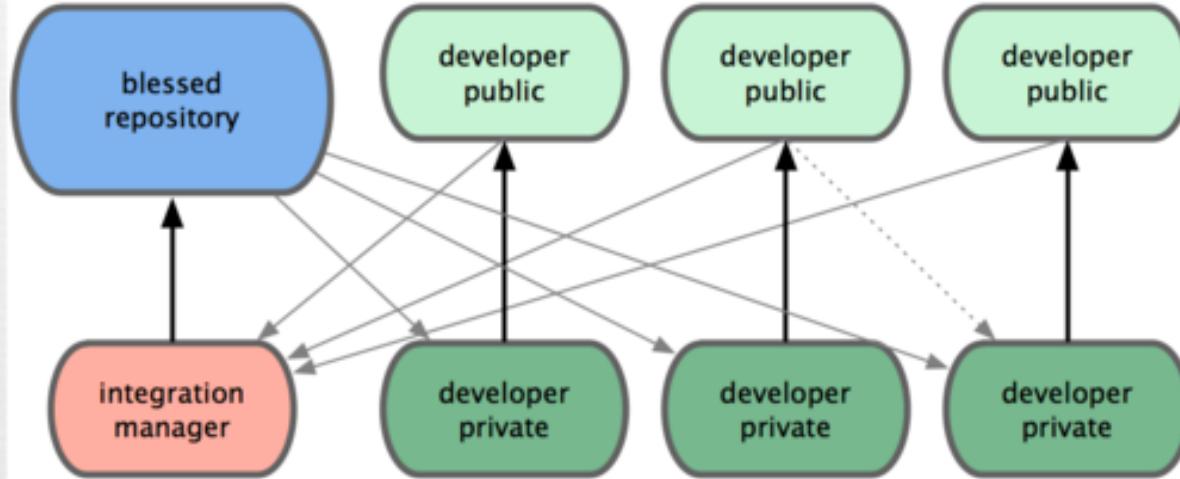


Centralized Workflow



集中式工作流

適合小型團隊

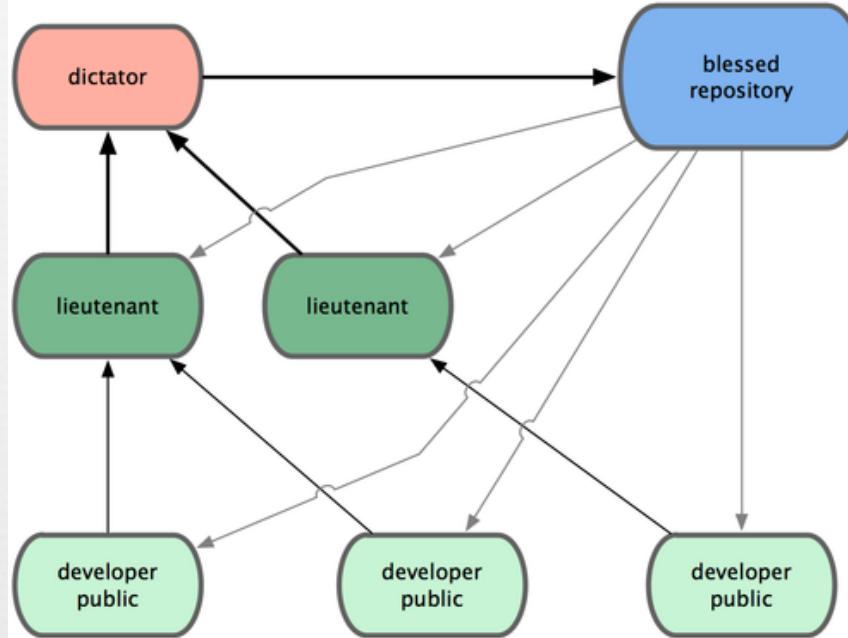


Integration-Manager Workflow



集成管理員工作流

適合網路多人協同專案，例如 GitHub.com



Dictator and Lieutenants Workflow



司令官與副官工作流

適合超大專案，例如 Linux Kernel

作業2



作業2



1. 建立一個新的 Unity 專案(Project) (20%)
2. 設定成 Git 倉庫(Repository)
3. 加入 3個 commit
 1. Initial Commit : 新增 Unity Project 適用的 .gitignore 檔 (30%)
 2. Create Readme : 新增 README.md 檔案，並且填入標題及你的學號 (20%)
 3. Update Readme : 修改 README.md 檔案，加入你的姓名 (20%)
4. 在 Github 上建立新的 私有倉庫 (Private Repository) - 命名為 **Git-Work2** (10%)
5. 上傳 Unity 專案到 Git-Work2 倉庫
6. 邀請我 (ArthurPai) 加入 Git-Work2 倉庫 (100%)
7. 確認我有在你的最後一個 commit (Update Readme)上，加上評論 (Comment)

作業2



Screenshot of a GitHub repository page showing commit history.

The repository is named **ArthurPai / git-test** (Private).

The main tab is selected, showing the **Code** view.

The current branch is **main**.

Commits on Aug 31, 2022:

- Update README.md** (Verified) by ArthurPai committed 2 minutes ago. Hash: 50b4513.
- Create README.md** (Verified) by ArthurPai committed 2 minutes ago. Hash: 117d853.

Commits on Dec 1, 2021:

- Initial commit** (Verified) by ArthurPai committed on 1 Dec 2021. Hash: 25616ea.

Navigation buttons at the bottom: **Newer** and **Older**.

Page footer:

- © 2022 GitHub, Inc.
- Terms
- Privacy
- Security
- Status
- Docs
- Contact GitHub
- Pricing
- API
- Training
- Blog
- About

小技巧



Tips

Stash



- ❖ Stash可以暫時把現在修改的部份先儲存起來。
- ❖ 主要是用在當開發過程時，遇到緊急bug需要修改，可是目前的工作寫到一半，就可以用Save Stash先存起來，這時Stage及Working directory的內容會回到HEAD所指的commit。
- ❖ 改好以後可以用Apply Stash拿回之前寫到一半的程式。
- ❖ 最後可以用Drop Stash指令把不用的Stash刪除。

Git 參考資料



References

❖ 簡介

- ❖ <https://git-scm.com/>
- ❖ <https://www.slideshare.net/littlebtc/git-5528339>

❖ 教學

- ❖ <https://iissnan.com/progit/index.zh-tw.html>
- ❖ <https://git-scm.com/book/en/v2>
- ❖ <https://marklodato.github.io/visual-git-guide/index-zh-tw.html>
- ❖ <https://ihower.tw/blog/archives/category/git>

❖ 其他

- ❖ <https://nvie.com/posts/a-successful-git-branching-model/>
- ❖ <https://github.com/github/gitignore>

提問



Q & A