

A decorative graphic on the left side of the slide, featuring a blue square, a red square, and a yellow square, with a black vertical line and a horizontal line intersecting them.

GIT chapter 3

- `github`
- `git remote/push`
- `markdown/ .gitignore`

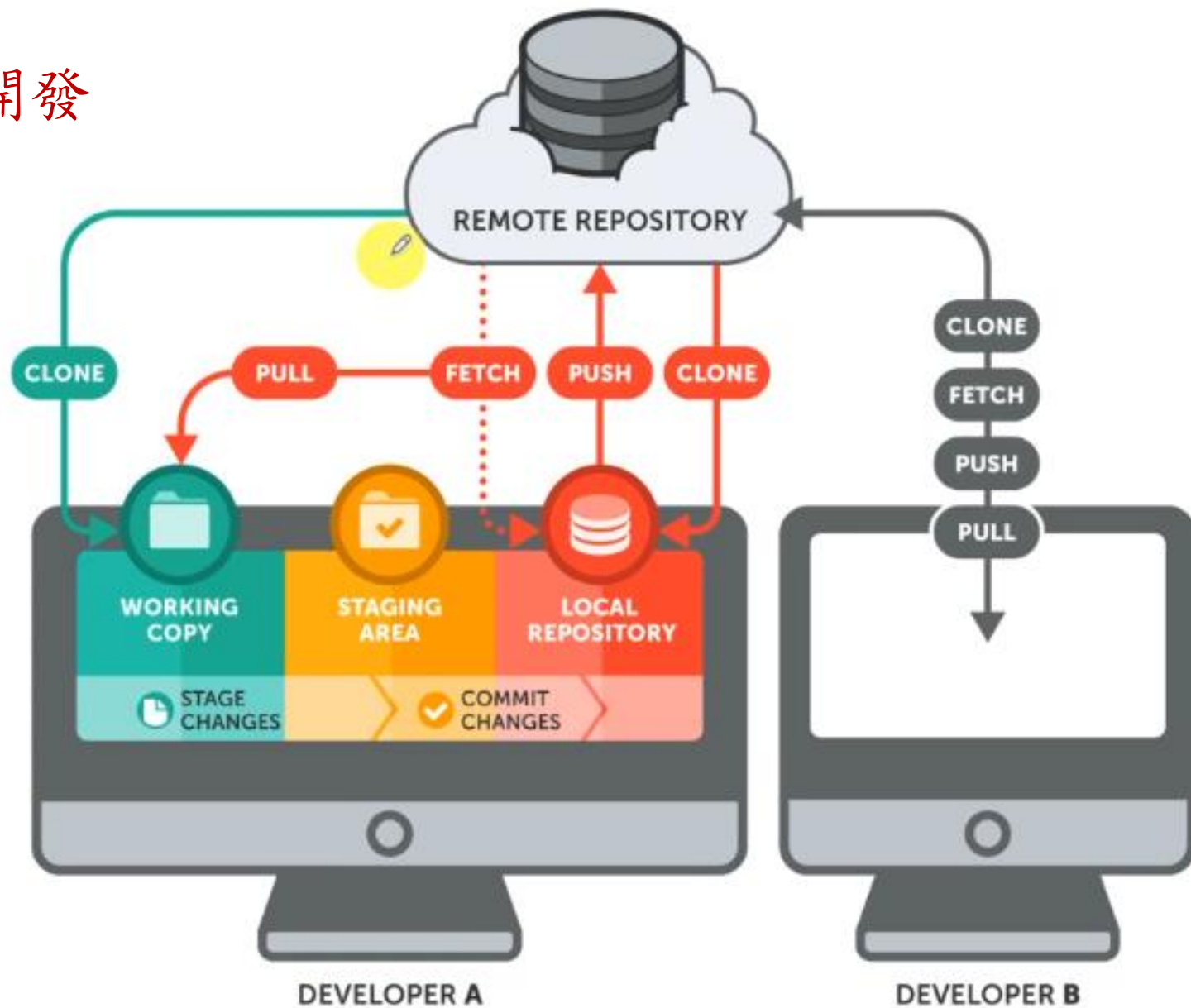


GIT remote

- 遠端版本控管服務器
- 建立遠端倉庫
- 進行同步(push/pull)

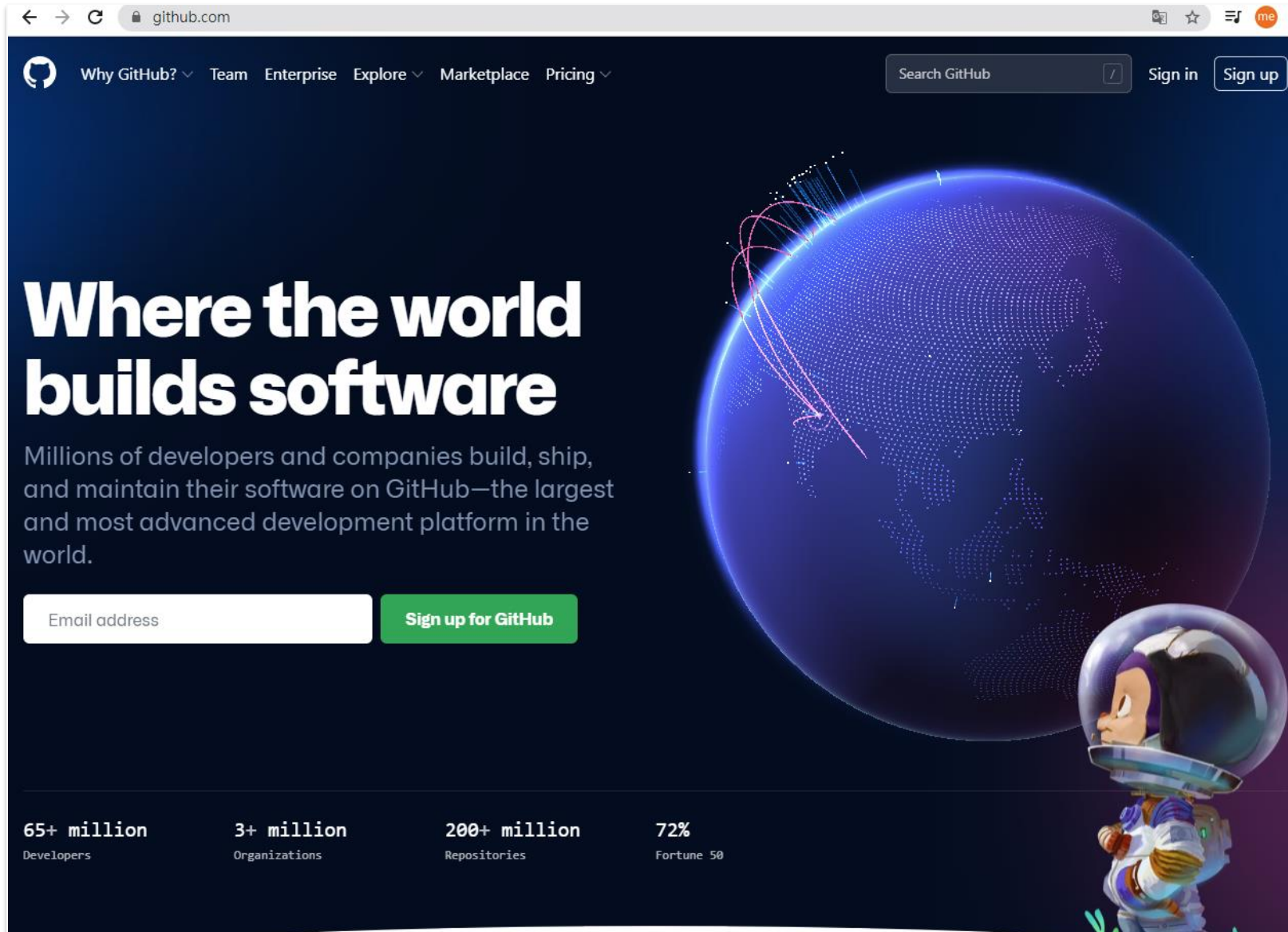
■ 遠端版本控管

■ 進行多人協同開發





■ 申請網址 <https://github.com/>



建立帳號



■ 建立遠端倉庫位置

The screenshot shows the GitHub homepage. In the top navigation bar, there are links for "Pull requests", "Issues", "Marketplace", and "Explore". On the left sidebar, under the "Repositories" section, there is a "Find a repository..." search bar and a list of repositories. A red box highlights a green "New" button with a plus icon, and a yellow arrow points from a yellow box labeled "新增" (Add) to this button. Below the "Repositories" section is the "Recent activity" section. The main content area on the right features a light green box with the heading "Learn without any code!" and a subheading "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." Below this text are two buttons: "Read the guide" and "Start a project". Further down is the "All activity" section, which includes a card titled "Introduce yourself" with instructions on creating a README and a list of prompts for a README file. At the bottom right of the page, there are two buttons: "Dismiss this" and "Continue".

Search or jump to... / Pull requests Issues Marketplace Explore

Repositories

Find a repository...

- codewithjerry / git-demo
- codewithjerry / ten-half-game
- codewithjerry / flask-todolist
- codewithjerry / dream-analyze-web
- codewithjerry / python_google_calendar
- udemy-course / git-test

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Learn without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

All activity

Introduce yourself

The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

```
codewithjerry / README.md
```

```
1 - 👋 Hi, I'm @codewithjerry
2 - 🤖 I'm interested in ...
3 - 🌱 I'm currently learning ...
4 - 🚀 I'm looking to collaborate on ...
5 - 📫 How to reach me ...
6
```

[Dismiss this](#) [Continue](#)

■ create a new repository



repository

選擇公開或私有

建立倉庫

Create a new repository

Owner *

17app001

Repository name *

cv

Great repository names are short and memorable. Need inspiration? How about [improved-succotash](#)?

Description (optional)

我的個人履歷網頁

☒



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.](#)

This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository



■ 上傳用等資訊

remote url
遠端倉庫位址

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

Set up in Desktop

or

HTTPS

SSH



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 "# git-test1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/codewithjerry/git-test1.git
git push -u origin main
```



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

```
git remote add origin https://github.com/codewithjerry/git-test1.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.



- 加入遠端倉庫位置
- `git remote add origin`
 - <https://github.com/>你的帳號名稱/xxxx.git

遠端倉庫地址

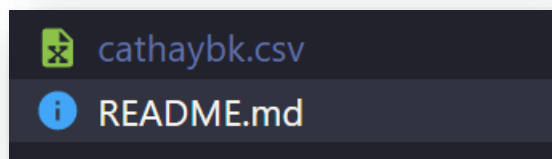
```
[core]
  repositoryformatversion = 0
  filemode = false
  bare = false
  logallrefupdates = true
  symlinks = false
  ignorecase = true
[remote "origin"]
  url = https://github.com/codewithjerry/git-test1.git
  fetch = +refs/heads/*:refs/remotes/origin/*
```




git push



- push之前
 - 新增README.md
 - 說明文件檔案



```
README.md > [abc] # 繪製表格 > [abc] ### 繪製表格 Tables
  > # GIT DEMO
  > 擺放學習GIT用得各項資源檔案，熟係使用指令。
  >
  > ![GIT圖片](https://encrypted-tbn0.gstatic.com/images?q=tbn:AND9GcRrRPMBBBa3Z3VH00Bx6aCk8aOKzp0TmfjOmBG2jf1mOW5mG4Ri_AVZA1JybkrjXyE1HBc&usqp=CAU )
  >
  > ### 創建日期
  > - 2021/12/8
  >
  > ### 使用工具
  > - [visual code](https://code.visualstudio.com/)
  > - [GIT](https://git-scm.com/)
  >
  > ### Markdown
  > - 參考資料
```



■ git push遠端推送

■ git push -u origin master

- origin (起源)倉儲
- master 主分支

推送到origin
的master分支

第二次後可以使
用git push即可

本地端master跟
遠端同步

```
PS D:\GoogleDrive\教學文件\Git\chapter4\sourcecode\merge> git log
commit 24ad0acc90686f058cfd91ad3f8e40d4913485f6 (HEAD -> master, origin/master)
Author: codewithjerry <codewithjerry1@gmail.com>
Date:   Fri Sep 17 13:28:16 2021 +0800

    modify test1

commit db216df3bd15aaf90aa2fb8c39dd645e07f1bc1e
Author: Jerry Chen <codewithjerry1@gmail.com>
Date:   Fri Sep 17 09:17:01 2021 +0800

    add test2.txt

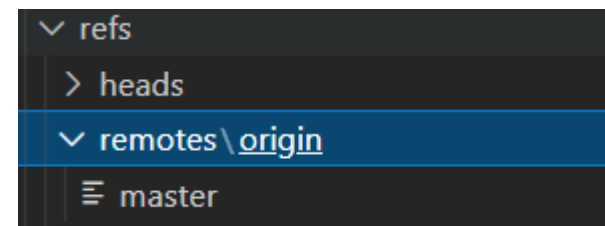
commit fe07ba776a4690aceec031a3ec09afeec0849575
Author: Jerry Chen <codewithjerry1@gmail.com>
Date:   Fri Sep 17 09:17:28 2021 +0800

    add test3.txt
```



- push成功後
 - 新增remote相關訊息

```
config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
[gui]
    wmsstate = normal
    geometry = 1053x539+96+96 228 244
[remote "origin"]
    url = https://github.com/codewithjerry/git-test1.git
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
    remote = origin
    merge = refs/heads/master
```



遠端 master &
local master 一致



■ 同步指向最新的commit

```
PS D:\GoogleDrive\教學文件\Git\chapter4\sourcecode\merge> git log  
commit 24ad0acc90686f058cf91ad3f8e40d4913485f6 (HEAD -> master, origin/master)
```

```
Author: codewithjerry <codewithjerry1@gmail.com>  
Date: Fri Sep 17 13:28:16 2021 +0800
```

```
    modify test1
```

```
commit db216df3bd15aaf90aa2fb8c39dd645e07f1bc1e
```

```
Author: Jerry Chen <codewithjerry1@gmail.com>  
Date: Fri Sep 17 09:17:01 2021 +0800
```

```
    add test2.txt
```

```
commit fe07ba776a4690aceec031a3ec09afeec0849575
```

```
Author: Jerry Chen <codewithjerry1@gmail.com>  
Date: Fri Sep 17 09:17:28 2021 +0800
```

```
    add test3.txt
```

本地端master跟
遠端同步完成

master 1 branch 0 tags

Go to file

Add file ▾

Code ▾



codewithjerry modify test1

24ad0ac 10 minutes ago 4 commits



test1.txt

modify test1

10 minutes ago



test2.txt

add test2.txt

4 hours ago



test3.txt

add test3.txt

4 hours ago



使用.gitignore

加入不需要控管的檔案

新增.gitignore



■ 使用.gitignore

- 排除不要加入控管的檔案(*.jpg , *.png images/..)

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project structure with a 'MERGE' folder containing files like .git, class, .gitignore, test1.jpg, test1.txt, test2.txt, test3.txt, and test4.txt. The main editor area shows the content of the .gitignore file, which is highlighted with a red box. The file contains two lines: `1 *.jpg` and `2 class/`. A yellow arrow points from a yellow box containing the text '檔案跟目錄' (Files and directories) to the red box. Below the editor, the TERMINAL panel shows the output of several git commands: `git rm -f test2.png`, `git status`, and `git add .` followed by `git commit -m "add gitignore"`. The output indicates that the files were untracked, removed, and then committed to the repository.

```
.gitignore
1 *.jpg
2 class/
```

檔案跟目錄

```
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
new file:   test2.png

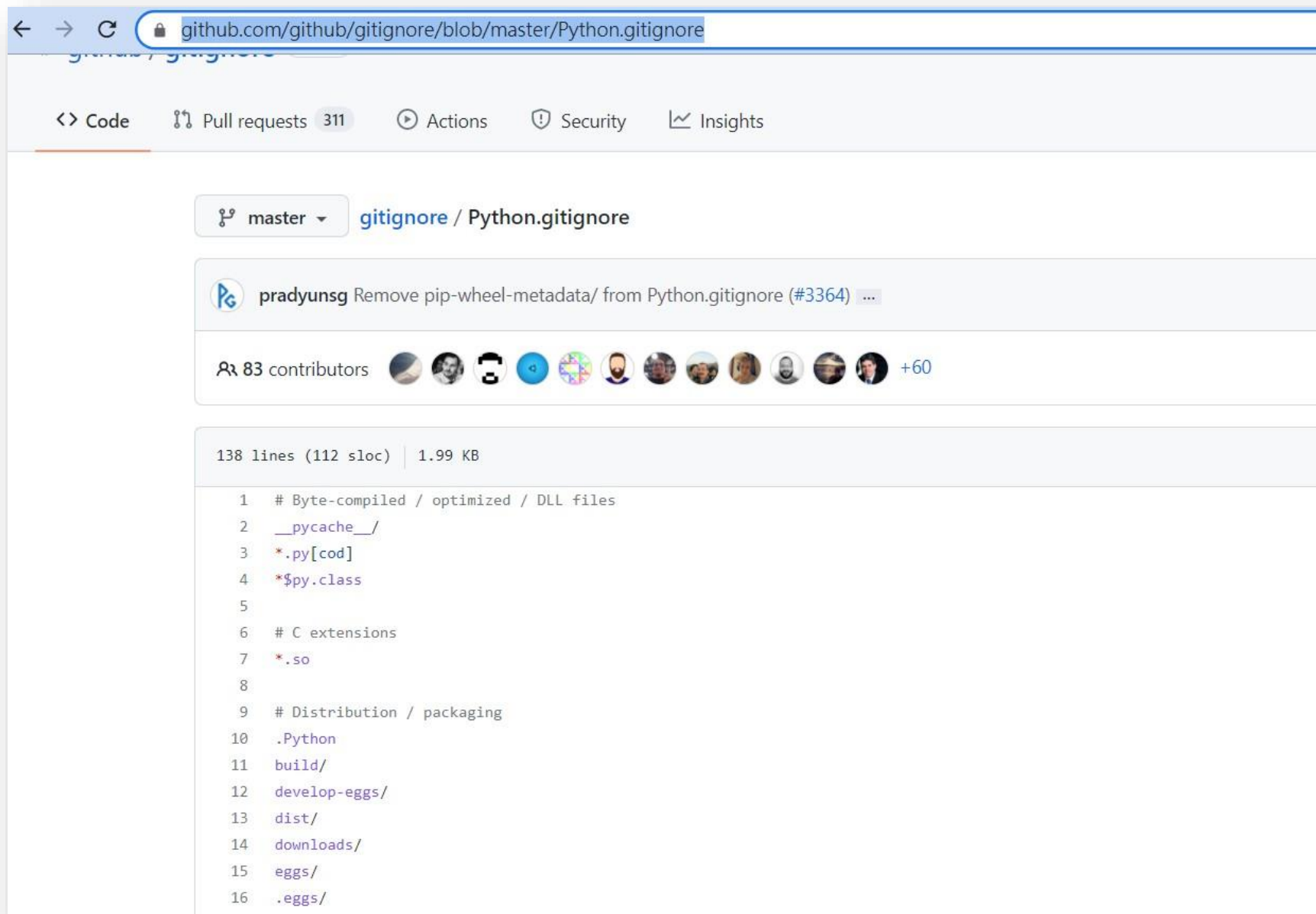
Untracked files:
(use "git add <file>..." to include in what will be committed)
.gitignore

PS D:\GoogleDrive\教學文件\Git\chapter5\sourcecode\merge> git rm -f test2.png
rm 'test2.png'
PS D:\GoogleDrive\教學文件\Git\chapter5\sourcecode\merge> git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
(use "git add <file>..." to include in what will be committed)
.gitignore

nothing added to commit but untracked files present (use "git add" to track)
PS D:\GoogleDrive\教學文件\Git\chapter5\sourcecode\merge> git add .
PS D:\GoogleDrive\教學文件\Git\chapter5\sourcecode\merge> git commit -m "add gitignore"
[master 3f0e8b8] add gitignore
1 file changed, 2 insertions(+)
create mode 100644 .gitignore
```

■ github搜尋gitignore

A screenshot of a web browser showing the GitHub repository page for the file 'Python.gitignore' in the 'gitignore' repository. The browser's address bar shows the URL 'github.com/github/gitignore/blob/master/Python.gitignore'. The page has a navigation bar with links for 'Code', 'Pull requests' (311), 'Actions', 'Security', and 'Insights'. Below the navigation bar, there is a section for the file 'Python.gitignore' with a dropdown menu set to 'master'. A commit by 'pradyunsg' is highlighted, with the message 'Remove pip-wheel-metadata/ from Python.gitignore (#3364)'. Below the commit, there are 83 contributors, with 10 avatars shown and '+60' indicating more. The file statistics show '138 lines (112 sloc)' and '1.99 KB'. The file content is displayed as a code block with line numbers 1 through 16.

github.com/github/gitignore/blob/master/Python.gitignore

<> Code Pull requests 311 Actions Security Insights

master gitignore / Python.gitignore

pradyunsg Remove pip-wheel-metadata/ from Python.gitignore (#3364) ...

83 contributors +60

138 lines (112 sloc) 1.99 KB

```
1 # Byte-compiled / optimized / DLL files
2 __pycache__/
3 *.py[cod]
4 *$py.class
5
6 # C extensions
7 *.so
8
9 # Distribution / packaging
10 .Python
11 build/
12 develop-eggs/
13 dist/
14 downloads/
15 eggs/
16 .eggs/
```




■ 使用python-gitignore

```
> __pycache__
> .git
◆ .gitignore u
≡ dev1.txt
🔗 guess_game.py
≡ test1.txt
≡ test2.txt
≡ test3.txt

1 # Byte-compiled / optimized / DLL files
2 __pycache__ /
3 *.py[co]
4 *$py.class
5
6 # C extensions
7 *.so
8
9 # Distribution / packaging
10 .Python
11 build/
12 develop-eggs/
13 dist/
14 downloads/
15 eggs/
16 .eggs/
17 lib/
18 lib64/
19 parts/
20 sdist/
21 var/
```

將被忽略



- 如果使用 `git commit --amend`
 - 推送時需要使用 `git push [origin master] -f`
 - `-f(force)` 強制更新遠端

```
! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/17app001/git-demo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
Writing objects: 100% (5/5), 4.06 KiB | 2.03 MiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/17app001/git-demo.git
+ 908efbd...38c2af6 master -> master (forced update)

C:\Users\Jerry\OneDrive\桌面\Django\git\demo>
```

A decorative graphic on the left side of the slide, featuring a blue square, a red square, and a yellow square, with a black crosshair-like structure overlaid on them.

git clone

複製專案方法



■ 直接從github檔案下載

- clone沒有權限設定，只要專案是公開的都可以進行複製，**唯獨寫(push)**需要帳號跟密碼及.git資訊

The screenshot shows the GitHub interface for the repository 'codewithjerry / git-test1'. The repository is public and has 1 branch and 0 tags. The 'Code' button is highlighted with a red box, and a yellow callout box with the text '直接下載' (Direct Download) points to it. The repository contains three files: test1.txt, test2.txt, and test3.txt. The 'Clone' dropdown menu is open, showing options for cloning via HTTPS, SSH, or GitHub CLI, and a link to the repository URL. Other options include 'Open with GitHub Desktop' and 'Download ZIP'.

codewithjerry / git-test1 Public

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

codewithjerry modify test1

File	Commit
test1.txt	modify test1
test2.txt	add test2.txt
test3.txt	add test3.txt

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

Clone

HTTPS SSH GitHub CLI New

https://github.com/codewithjerry/git-test1

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

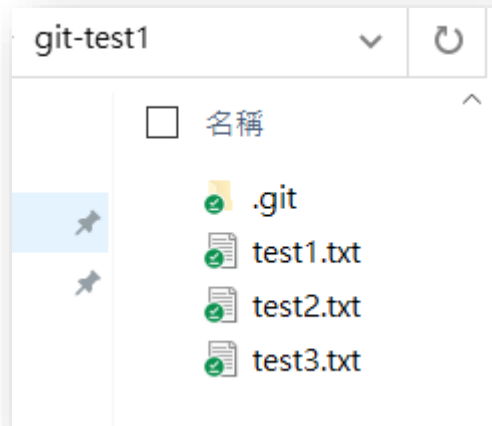
© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



■ 使用git clone指令

- <https://github.com/你的帳號名稱/XXXXXX>
- 會在該路徑內新增專案名稱目錄

需要帳密



```
commit 24ad0acc90686f058cf91ad3f8e40d4913485f6 (HEAD -> master, origin/master,
origin/HEAD)
Author: codewithjerry <codewithjerry1@gmail.com>
Date:   Fri Sep 17 13:28:16 2021 +0800

    modify test1

commit db216df3bd15aaf90aa2fb8c39dd645e07f1bc1e
Author: Jerry Chen <codewithjerry1@gmail.com>
Date:   Fri Sep 17 09:17:01 2021 +0800

    add test2.txt

commit fe07ba776a4690aceec031a3ec09afeec0849575
Author: Jerry Chen <codewithjerry1@gmail.com>
Date:   Fri Sep 17 09:17:28 2021 +0800

    add test3.txt

commit f0a4194001e5b9b937d82d4a44090ea4ff5c1684
Author: Jerry Chen <codewithjerry1@gmail.com>
Date:   Fri Sep 17 09:15:41 2021 +0800

    1st commit
```



MINGW64:/c/Users/Jerry/OneDrive/桌面/django

```
Jerry@DESKTOP-E7K1RS1 MINGW64 ~/OneDrive/桌面/django
$ git clone https://github.com/17app001/git-demo
Cloning into 'git-demo'...
remote: Enumerating objects: 73, done.
remote: Counting objects: 100% (73/73), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 73 (delta 13), reused 66 (delta 11), pack-reused 0
Receiving objects: 100% (73/73), 6.75 KiB | 531.00 KiB/s, done.
Resolving deltas: 100% (13/13), done.
```

```
Jerry@DESKTOP-E7K1RS1 MINGW64 ~/OneDrive/桌面/django
$
```

完整檔案結構(包含.git)

django > git-demo				
搜尋 git-demo				
名稱	修改日期	類型	大小	
.git	2021/12/8 下午 02:43	檔案資料夾		
1.py	2021/12/8 下午 02:43	Python 來源檔案	1 KB	
1.txt	2021/12/8 下午 02:43	文字文件	1 KB	
2.txt	2021/12/8 下午 02:43	文字文件	1 KB	
3.txt	2021/12/8 下午 02:43	文字文件	1 KB	
4.txt	2021/12/8 下午 02:43	文字文件	1 KB	
5.txt	2021/12/8 下午 02:43	文字文件	1 KB	



遠端更新檔案

使用github更新檔案



- <https://github.com/17app001/git-demo>

The screenshot shows the GitHub repository page for `17app001 / git-demo`. The repository name is highlighted with a yellow box and labeled "倉庫名稱" (Repository Name). The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The "Settings" link is highlighted with a red box. The branch selection area shows "master" as the current branch, with "1 branch" and "0 tags" indicated. A yellow box labeled "分支" (Branch) points to this area. The "Code" button is highlighted with a red box and labeled "下載" (Download). The commit history table is highlighted with a red box and contains the following data:

File	Commit Message	Time Ago
1.py	Update 1.py	8 minutes ago
1.txt	Revert "update all"	2 hours ago
2.txt	Revert "update all"	2 hours ago
3.txt	Revert "update all"	2 hours ago
4.txt	update 4.txt	42 minutes ago
5.txt	add 5.txt	38 minutes ago

At the bottom, there is a light blue banner with the text "Help people interested in this repository understand your project by adding a README." and a green button labeled "Add a README".



■ 可線上編輯跟commit

17app001 / git-demo Public

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master git-demo / 1.py / <> Jump to Go to file ...

17app001 Update 1.py Latest commit 51d744f 12 minutes ago History

1 contributor

2 lines (2 sloc) 40 Bytes

```
1 print('Hello World!123')
2 print('done!')
```

修改複製
刪除..

Raw Blame View file Edit file Delete file

© 2021 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About



■ 編輯後commit

git-demo / 1.py in master Cancel changes

<> Edit file Preview changes Spaces 2 No wrap

```
1 print('Hello World123')
2 print('done!')
3
```

Commit changes

Create 1.py

Add an optional extended description...

☒ Commit directly to the master branch.

☐ Create a new branch for this commit and start a pull request. [Learn](#)

Commit changes Cancel

等同遠端多一個commit

A decorative graphic on the left side of the slide, featuring overlapping blue, red, and yellow squares with a black crosshair.

git pull

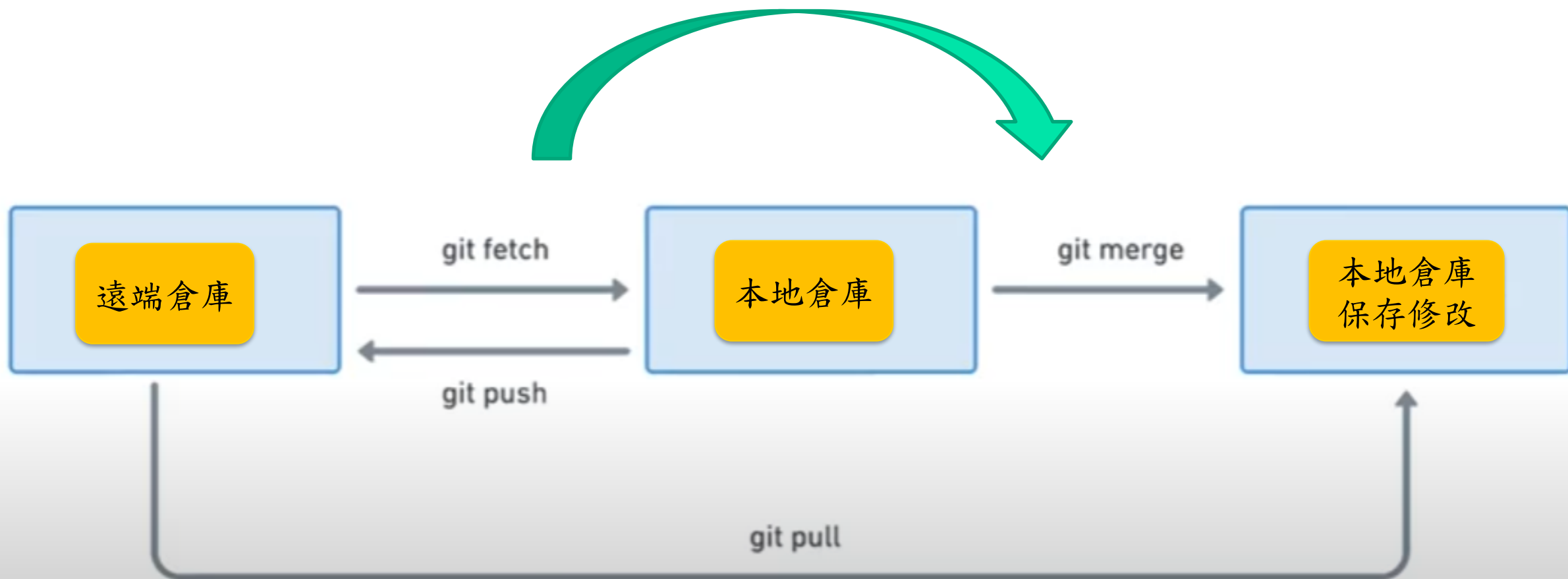
遠端拉取回本地端
(本地端跟遠端進行同步)



■ 為何需要pull?

- 遠端/其他使用者 有更新資料需要先進行拉取，否則無法push
- 一般pull會進行同步跟自動合併(如有衝突需要先處理)

■ Git pull





- 使用git pull
 - 遠端倉庫如新增/修改檔案
 - 本地端使用push會失敗
 - 需要先pull(同步)



```
PS C:\Users\Jerry\Google 雲端硬碟\教學文件\Git\git-demo> git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 646 bytes | 43.00 KiB/s, done.
From https://github.com/codewithjerry/git-demo
   8bd6c1f..98c8145  master    -> origin/master
Updating 8bd6c1f..98c8145
Fast-forward
 6.txt | 1 +
1 file changed, 1 insertion(+)
create mode 100644 6.txt
```



Git Remote的一些操作



- `git clone <repository URL>` clone a remote repository to local disk
- `git push origin <branch_name>` push local branch `<branch_name>` to remote branch dev, if `dev` branch does not exist remotely, it would be created. (`--delete` to delete a remote branch)
- `git branch -a` list all branches both local and remote
- `git fetch` download objects and refs from remote repository
- `git pull` - Fetch from remote and integrate with local branch
- `git push origin <tag_name>` push local tag to github
- `git push origin --delete <tag_name>` to delete github tags



- git remote -v
 - 檢視目前所有遠端url

```
Jerry@DESKTOP-E7K1RS1 MINGW64 /d/GoogleDrive/教學文件/Git/chapter4/
ster)
$ git remote
origin

Jerry@DESKTOP-E7K1RS1 MINGW64 /d/GoogleDrive/教學文件/Git/chapter4/
ster)
$ git remote -v
origin https://github.com/codewithjerry/git-test1.git (fetch)
origin https://github.com/codewithjerry/git-test1.git (push)
```

- git push → 本地推送到遠端
- git pull → 遠端更新到本地(with merge)



- `git remote remove origin`
 - 移除遠端origin
- `git remote add origin https://xx.xxxxxx/xxx.git`
 - 新增倉庫名稱
- 推送到其他分支
 - `git push origin dev`
- 遠端分支的刪除
 - `git push --delete origin dev`



- `git push [origin master] -f`
 - 強制更新遠端(不理衝突，譬如使用 `- --amend` 後)