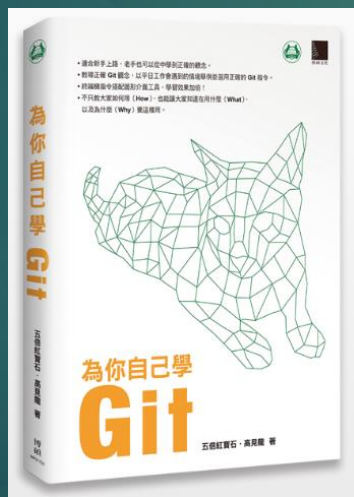


Git/GitHub 學習參考資料



參考資料：為你自己學 Git (網路版)
<https://gitbook.tw/>

陽明生資所 陳卓逸老師實驗室
專任研究助理 陳躍中 製作
2018.07.12

前言

2

- ▶ 此簡報注重於如何在 Windows 作業系統使用 Git, 並將檔案上傳至 GitHub , 以及從 GitHub 下載檔案到個人電腦。

Git & GitHub

3

- ▶ Git 為一款版本控制軟體。而 GitHub 是一個 Git 伺服器，此網站上的應用程式可以讓大家透過 web 的操作介面，來操作一些原本需要較複雜 Git 指令才能做到的事情。

安裝 Git 在 Windows 作業系統

4

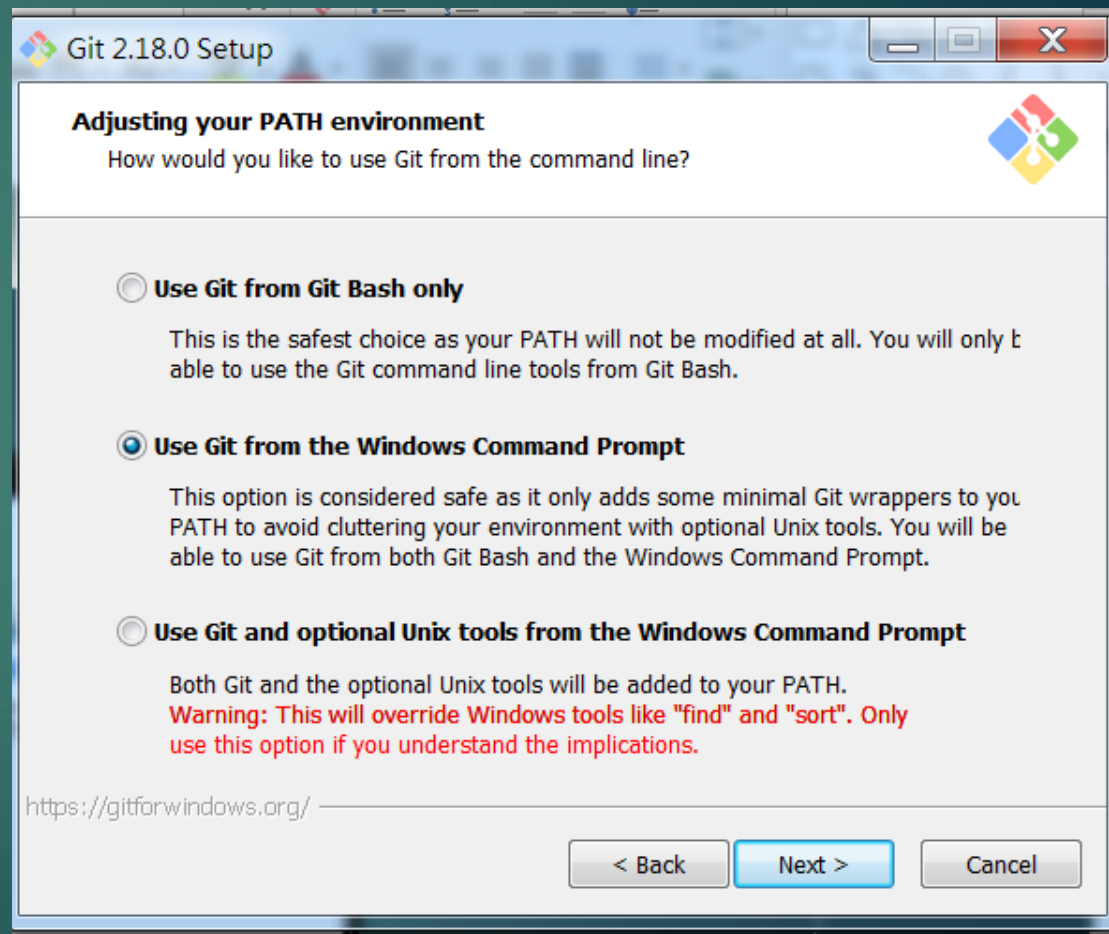
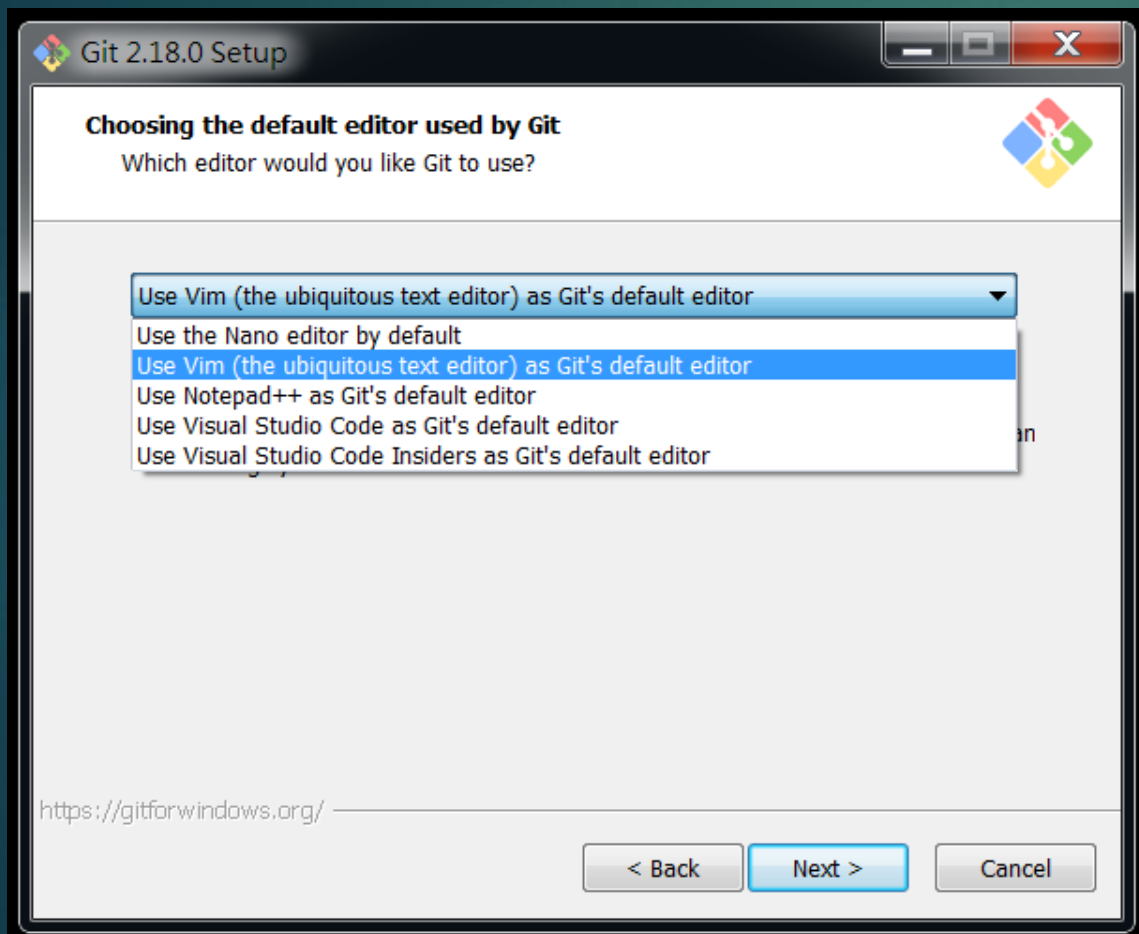
- ▶ 至官方網站下載合適的版本，網址：<https://git-scm.com/download/win>
- ▶ 我在安裝位置選擇 D 槽，優點是可以不用佔 C 槽 (作業系統) 的硬碟空間，缺點是切換資料夾比直接從 C 槽麻煩，以上見仁見智。

補充：特別注意在操作 Git 時，
指令要在正確的目錄下才能正常運作。



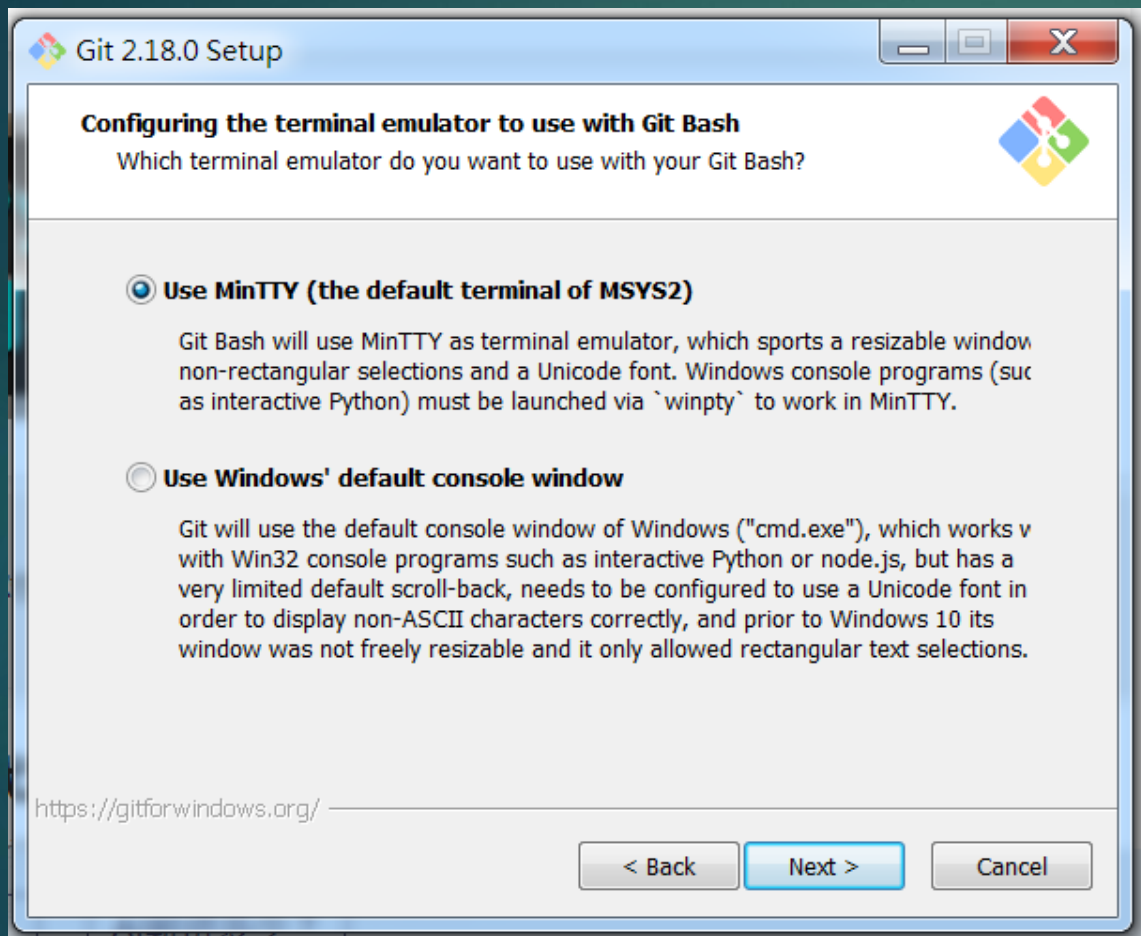
左(選擇文字編輯器，這邊推薦選 Vim)
右(選擇 Use Git from the Windows Command Prompt)

5



選擇 Use MinTTY (the default terminal of MSYS2)

6

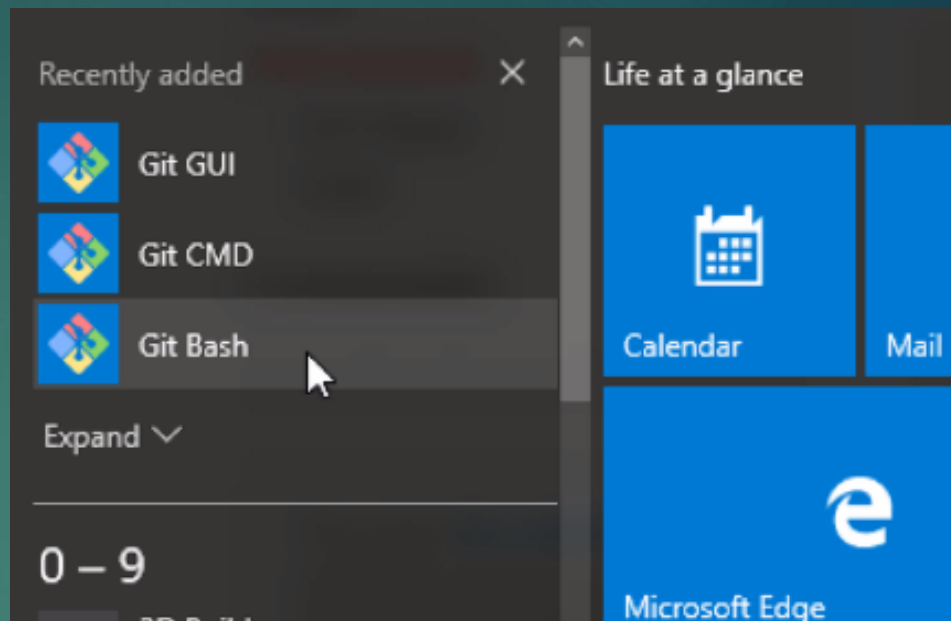
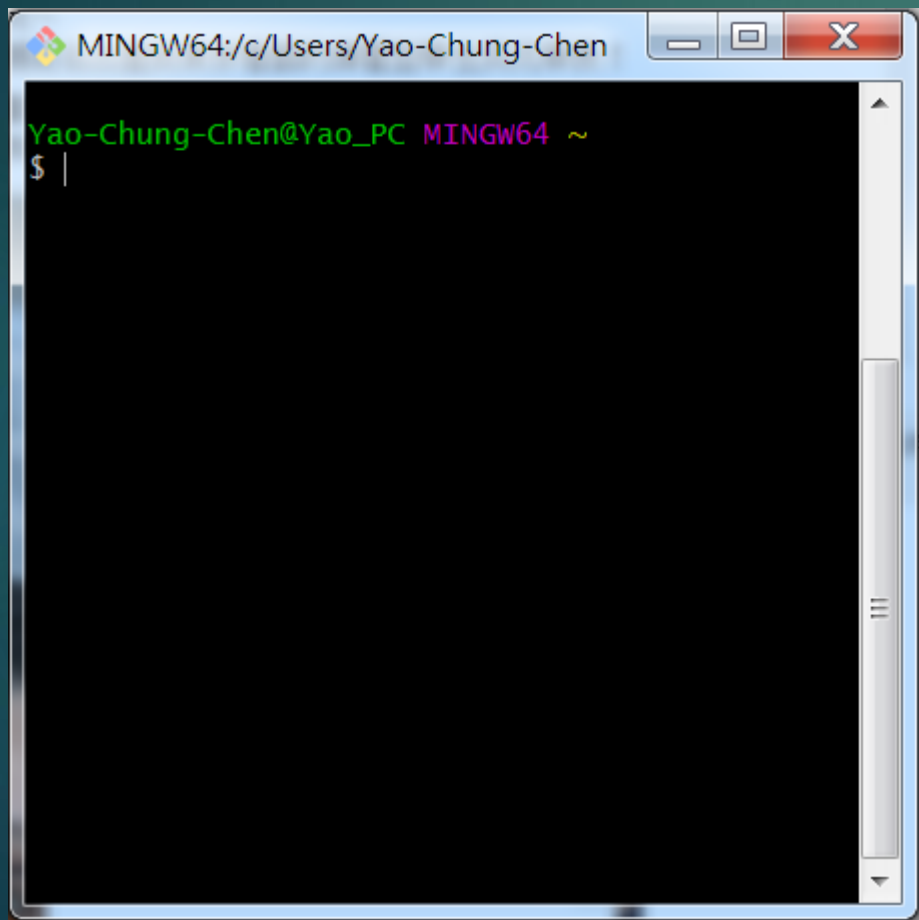


開啟 Git Bash



7

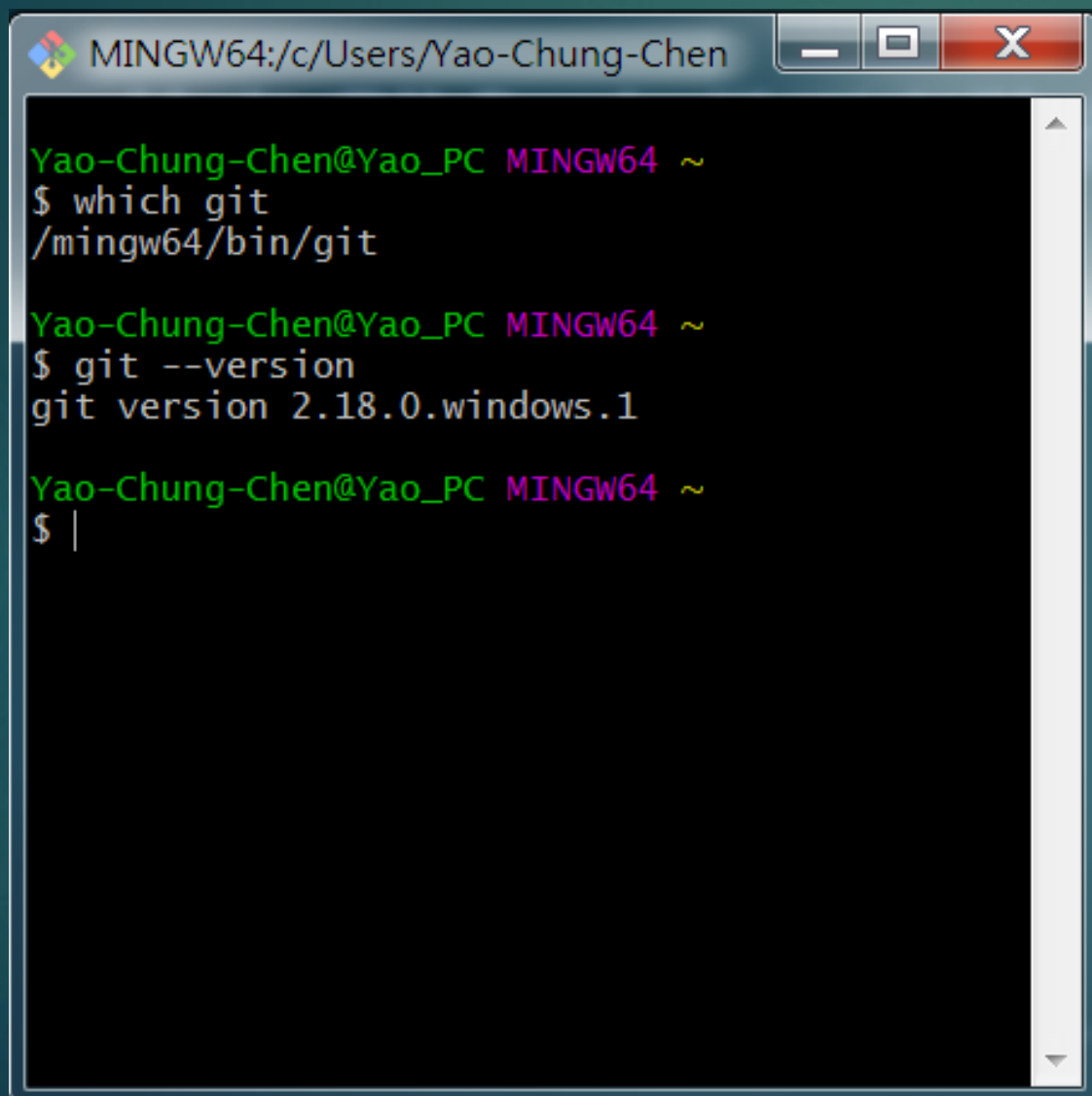
- ▶ 安裝好之後，請至程式集找 Git Bash



←開啟 Git Bash 後，會看到像是左邊黑色視窗，此作業視窗模擬 Linux 系統的軟體 (Bash)

檢查 Git Bash 是否安裝成功

8



```
MINGW64:/c/Users/Yao-Chung-Chen

Yao-Chung-Chen@Yao_PC MINGW64 ~
$ which git
/mingw64/bin/git

Yao-Chung-Chen@Yao_PC MINGW64 ~
$ git --version
git version 2.18.0.windows.1

Yao-Chung-Chen@Yao_PC MINGW64 ~
$ |
```

輸入 `which git` 驗證是否有安裝成功

輸入 `git --version` 檢查版本 (版本可能會有所不同)

若看到如左圖的類似訊息，就代表安裝成功拉！

圖形化介面工具(GUI, Graphic User Interface) 的輔助

- ▶ 有些使用者對終端機(或是命令提示字元)的操作不習慣，因此在懂得 Git 的系統運作及原理後，可以選擇是否使用 GUI 來操作。
- ▶ 這本書有推薦兩款 Git 的 GUI: (1) GitHub Desktop (2) SourceTree

設定 Git 使用者設定

10

- ▶ 打開 Git Bash 輸入使用者名稱和 Email：

並輸入 `$ git config --global user.name "你的使用者名稱"`

`$ git config --global user.email "你的電子信箱"`

- ▶ 檢視設定狀況：`$ git config --list`
可能會出現很多設定的結果，
重點是要看 `user.name` 和 `user.email`

A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/Yao-Chung-Chen'. The prompt is 'Yao-Chung-Chen@Yao_PC MINGW64 ~'. The command '\$ git config --list' has been executed, and the output is displayed. The output includes various Git settings, with 'user.name=ferygood' and 'user.email=yaochung41@gmail.com' highlighted by red arrows from the text on the left. The terminal window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Yao-Chung-Chen@Yao_PC MINGW64 ~  
$ git config --list  
core.symlinks=false  
core.autocrlf=true  
core.fscache=true  
color.diff=auto  
color.status=auto  
color.branch=auto  
color.interactive=true  
help.format=html  
rebase.autosquash=true  
http.sslcainfo=D:/Git/mingw64/ssl/certs/ca-bundle.crt  
http.sslbackend=openssl  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
credential.helper=manager  
user.name=ferygood  
user.email=yaochung41@gmail.com  
Yao-Chung-Chen@Yao_PC MINGW64 ~
```

新增、初始 Repository

11

1. 使用 `mkdir` 指令建立了 `git-practice` 目錄。
2. 使用 `cd` 指令切換到剛剛建立的 `git-practice` 目錄。
3. 使用 `git init` 指令初始化這個目錄，主要目的是要讓 Git 開始對這個目錄進行版本控制。

```
MINGW64:/d/My_Git/git-practice
Yao-Chung-Chen@Yao_PC MINGW64 ~
$ cd d:

Yao-Chung-Chen@Yao_PC MINGW64 /d
$ cd My_Git/

Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git
$ mkdir git-practice

Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git
$ cd git-practice

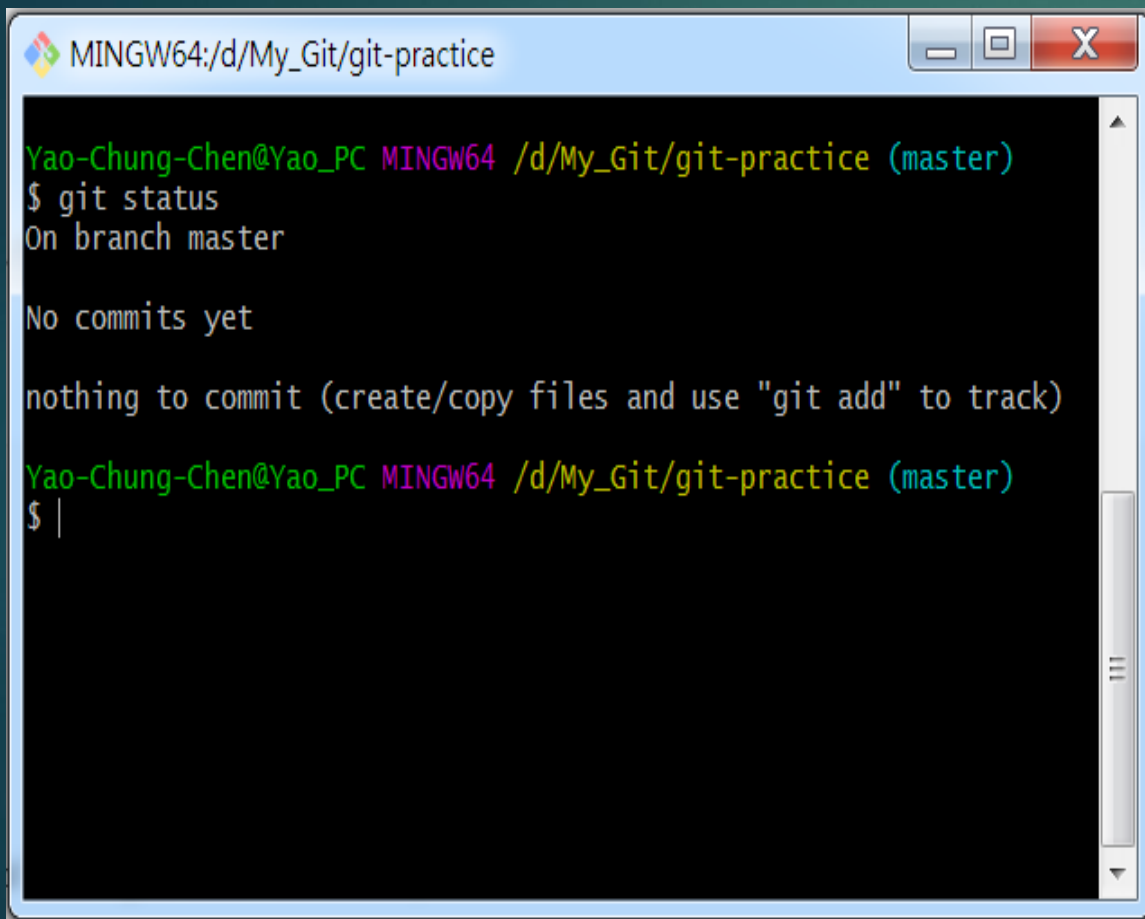
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice
$ git init
Initialized empty Git repository in D:/My_Git/git-practice/.git/

Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$
```

► 其中，第 3 步這個指令會在這個目錄裡建立了一個 `.git` 目錄，整個 Git 的精華就都是在這個目錄裡了。如果各位有興趣可以先看一下這個目錄裡面的內容，不過現在並打算介紹裡面的細節，請各位先體會一下使用 Git 的手感，待後面的章節會再慢慢介紹這個目錄裡到底在賣什麼藥。

檢視目前狀態

12



```
MINGW64:/d/My_Git/git-practice

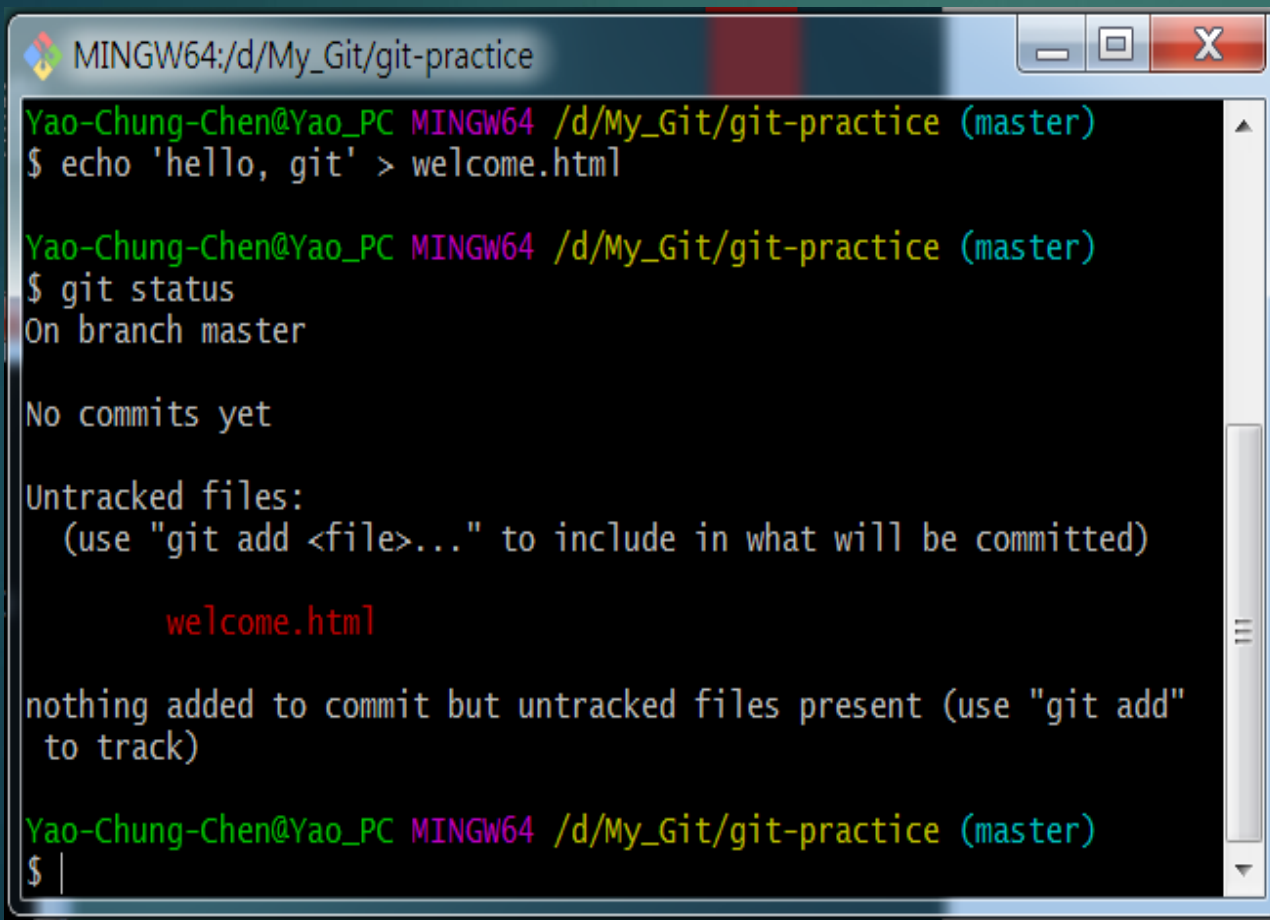
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ |
```

\$ git status (用來看狀態)

A screenshot of a Windows terminal window titled 'MINGW64:/d/My_Git/git-practice'. The window shows a series of commands and their outputs. The user is on the 'master' branch. They first run 'echo 'hello, git' > welcome.html'. Then they run 'git status', which shows that 'welcome.html' is an untracked file. The terminal text is as follows:

```
MINGW64:/d/My_Git/git-practice
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ echo 'hello, git' > welcome.html
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ git status
On branch master

No commits yet

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

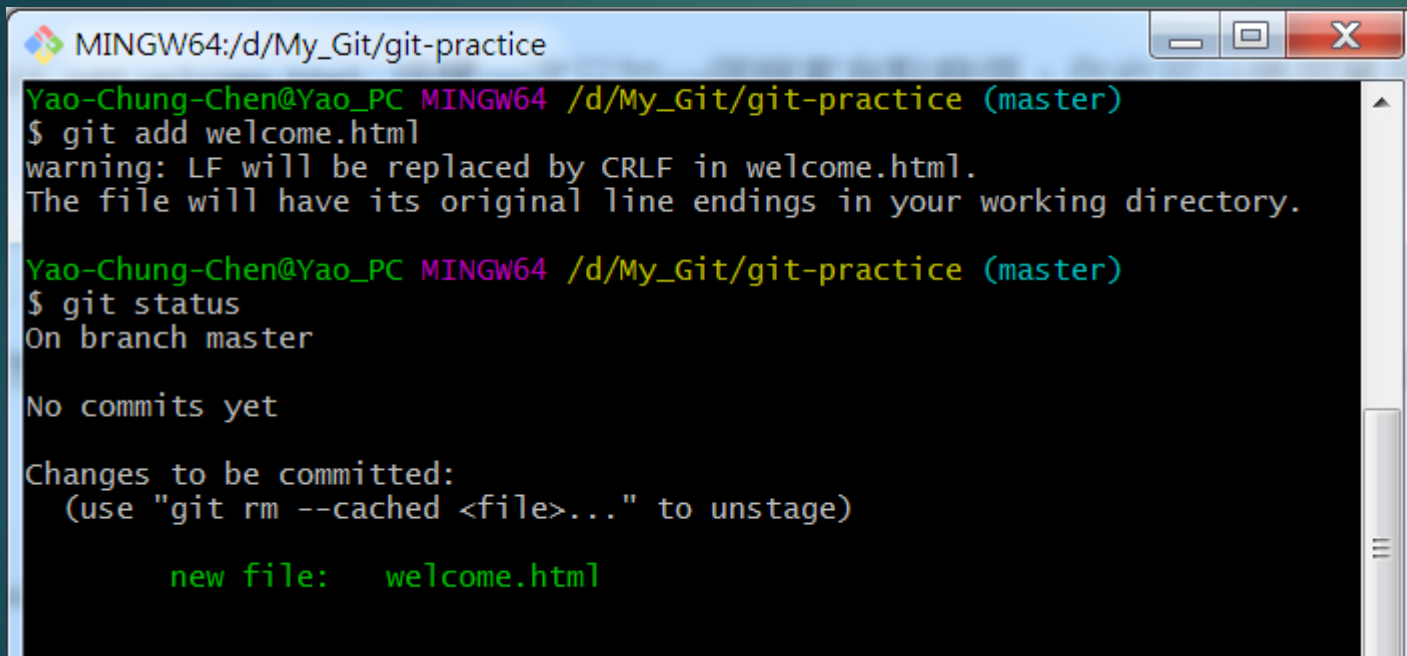
        welcome.html

nothing added to commit but untracked files present (use "git add"
to track)
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ |
```

- ▶ `$ echo 'hello, git' > welcome.html`
- ▶ `$ git status`
welcome.html 為 untracked file
- ▶ 接下來要把檔案交給 Git，讓 Git 追蹤它

把檔案交給 Git (儲存至暫存區)

14

A screenshot of a Windows command prompt window titled "MINGW64:/d/My_Git/git-practice". The prompt shows the user "Yao-Chung-Chen@Yao_PC" in the "MINGW64" environment, currently on the "master" branch. The user enters the command "\$ git add welcome.html". The output shows a warning: "warning: LF will be replaced by CRLF in welcome.html. The file will have its original line endings in your working directory." The user then enters "\$ git status". The output shows "On branch master", "No commits yet", and "Changes to be committed: (use \"git rm --cached <file>...\" to unstage)". Under "Changes to be committed:", it lists "new file: welcome.html".

```
MINGW64:/d/My_Git/git-practice
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ git add welcome.html
warning: LF will be replaced by CRLF in welcome.html.
The file will have its original line endings in your working directory.

Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   welcome.html
```

- ▶ \$ git add 檔案名稱
如 \$ git add welcome.html
- ▶ 可以發現已經從 untracked file 變成 new file 了。
表示這個檔案已經被儲存到鎮存區了 Staging Area/Index
- ▶ \$ git add *.html
- ▶ \$ git add --all

將暫存區內的檔案提交道倉庫存檔

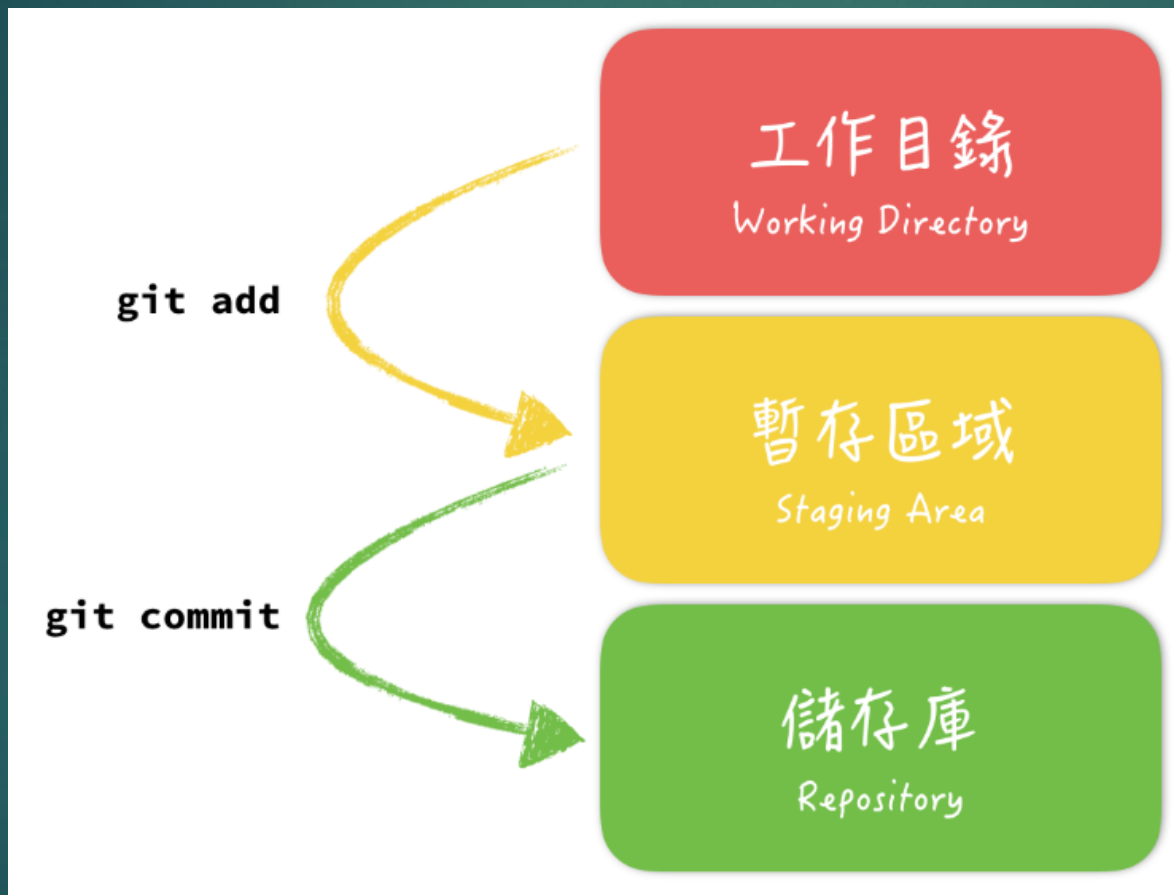
15

```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ git commit -m 'add a new file: welcome.html'
[master (root-commit) 28169df] add a new file: welcome.html
1 file changed, 1 insertion(+)
create mode 100644 welcome.html
```

- ▶ `$ git commit -m '內容'`
- ▶ Commit 完成才算是完成儲存的任務
- ▶ 左圖 'add a new file: welcome.html' 為記錄這次儲存的内容跟之前的比較做了哪些變更。
- ▶ `$ git commit >` 進入 vim

\$git add & \$git commit 的關係

16



檢視記錄 \$ git log

17

```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ git log
commit 28169dff1a6742138c3a201cbcbaeaf29373f851 (HEAD -> master)
Author: ferygood <yaochung41@gmail.com>
Date:   Tue Jul 10 23:21:15 2018 +0800

    add a new file: welcome.html
```

- ▶ SHA-1 (Secure Hash Algorithms 1) 40 位數的 十六進位 Pseudo random

```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git/git-practice (master)
$ git log --oneline --graph
* 28169df (HEAD -> master) add a new file: welcome.html
```

HEAD 是什麼？

- ▶ HEAD 是一個指標，指向某一個分支，通常你可以把 HEAD 當作「目前所在的分支」看待。在 .git 目錄裡有一個檔名為 HEAD 的檔案，就是紀錄 HEAD 的內容。
- ▶ 不過 HEAD 也不一定總是指向某個分支，當 HEAD 沒有指向某個分支的時候便會造成“detached HEAD”的狀態。

```
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (master)
$ cat .git/HEAD
ref: refs/heads/master

Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (master)
$ cat .git/refs/heads/master
38b78c3820df269f53f0192d9e956f8caacfa8a4
```

```
$ cat .git/HEAD
```

從檔案看起來，HEAD 目前正指向 master 分支。

```
$ cat .git/refs/heads/master
```

看內容可以發現所謂 Master 分支為 40 字元檔案。

使用分支

- ▶ 何時使用分支？新增功能、修正 Bug、想實驗看看某些新的做法.....等
- ▶ 使用分支：\$ git branch

```
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/  
My_Git/Rosalind_Bioinformatics_Stronghold (master)  
  
$ git branch  
* master
```

有星號* 代表現在正在這個分支上。

- ▶ 新增分支：\$ git branch cat (新增一個叫做 cat 的分支)
分支改名稱：\$ git branch -m cat tiger (把 cat 改成 tiger)

```
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/  
My_Git/Rosalind_Bioinformatics_Stronghold (master)  
  
$ git branch cat  
  
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/  
My_Git/Rosalind_Bioinformatics_Stronghold (master)  
  
$ git branch  
cat  
* master
```

刪除、切換分支

20

```
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (master)

$ git branch
cat
dog
* master

Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (master)

$ git branch -d dog
Deleted branch dog (was 38b78c3).

Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (master)

$ git branch
cat
* master
```

```
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (master)

$ git checkout cat
Switched to branch 'cat'

Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (cat)

$ git branch
* cat
master
```

合併分支

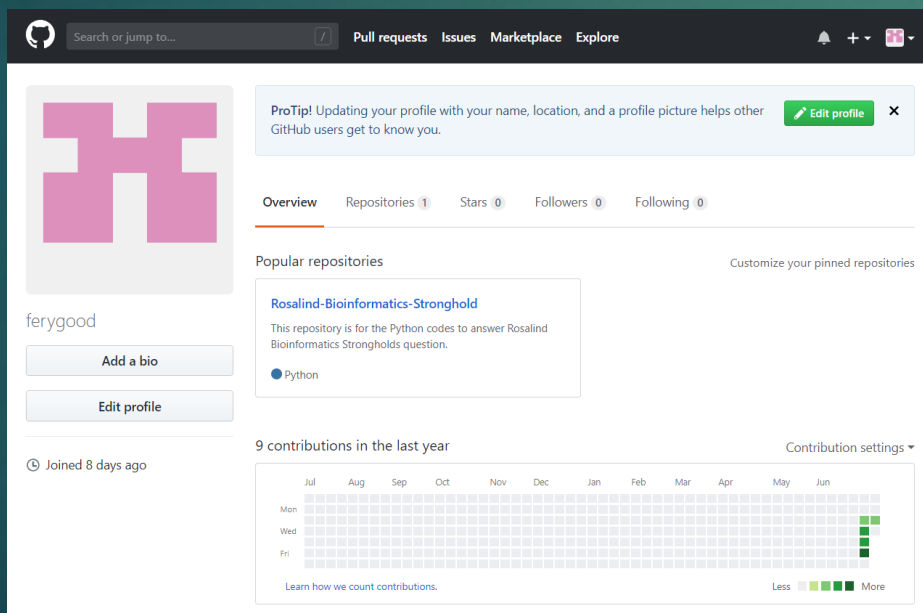
- ▶ 假設現在 master 分支，想要把 cat 分支併入：`$ git merge cat`
- ▶ 合併分支是指向某個 commit 的指標，合併分支其實也就是合併 commit
- ▶ `$ git reflog` (可以看之前做過什麼 commit/ 分支的改動)

```
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Stronghold (master)
$ git reflog
38b78c3 (HEAD -> master, origin/master, cat) HEAD@{0}: checkout: moving from cat to master
38b78c3 (HEAD -> master, origin/master, cat) HEAD@{1}: checkout: moving from master to cat
```

遠端共同協作 – 使用 GitHub

23

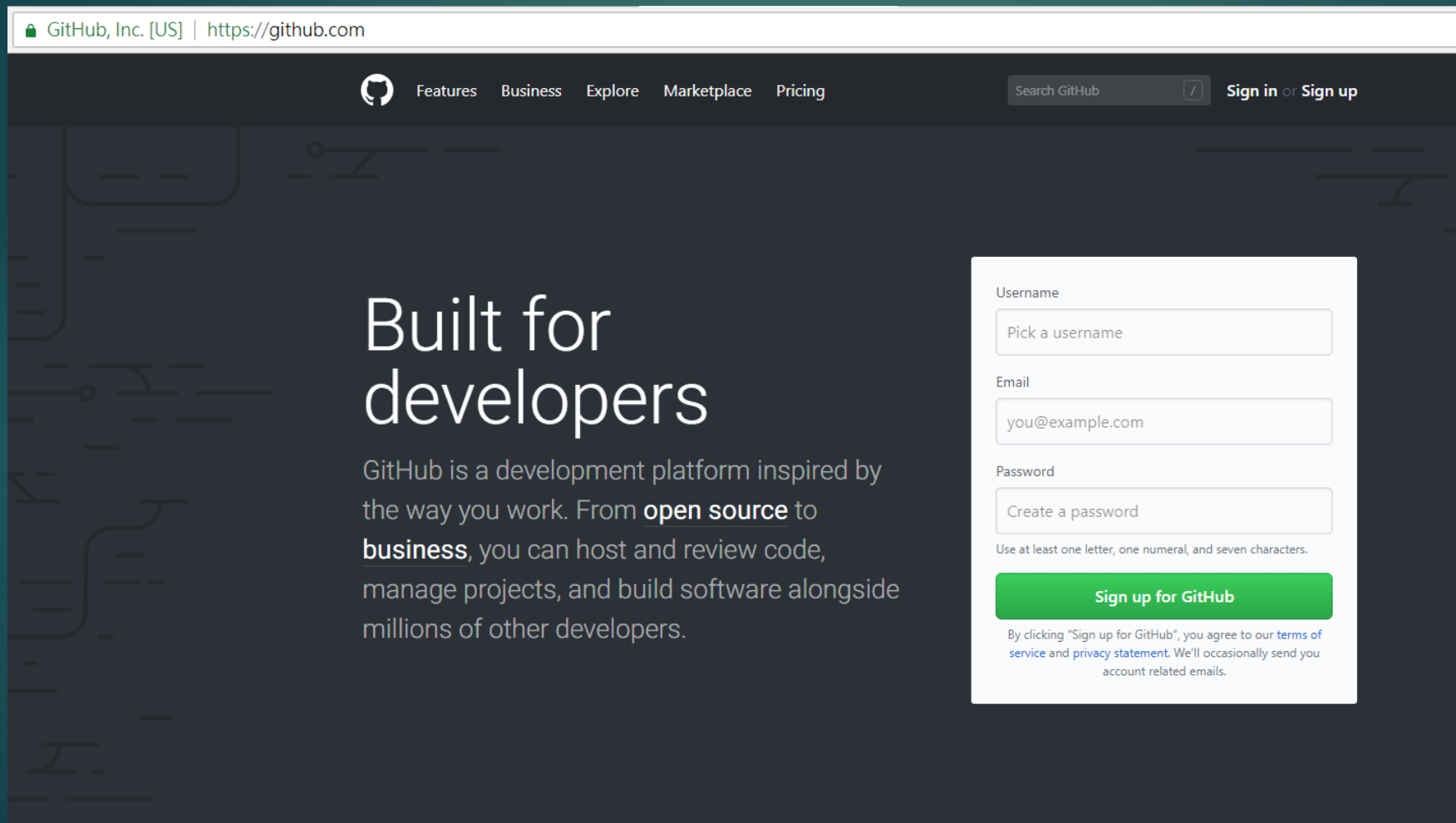
- ▶ GitHub 為一個目前全球最大的商業 Git Server，在這裡可以跟其他開發者交流，貢獻幫忙其他人的專案，其他人也可以回饋到你的專案，建立良性循環。也是開發者最好的履歷，曾經做過哪些專案、做過哪些貢獻、寫過哪些 code 等等。
- ▶ Open Source 免費。
- ▶ 學生優惠方案 <https://education.github.com/pack>



申請 GitHub 帳號

24

► <https://github.com>

A screenshot of the GitHub website's sign-up page. The browser's address bar shows 'GitHub, Inc. [US] | https://github.com'. The page has a dark header with the GitHub logo, navigation links (Features, Business, Explore, Marketplace, Pricing), a search bar, and 'Sign in or Sign up' links. The main content area has a dark background with a light-colored sign-up form on the right. The form includes fields for Username, Email, and Password, each with a placeholder text. Below the password field is a note about password requirements. A green 'Sign up for GitHub' button is at the bottom of the form. To the left of the form, the text 'Built for developers' is prominently displayed, followed by a paragraph describing GitHub as a development platform for open source and business.

GitHub, Inc. [US] | <https://github.com>

Features Business Explore Marketplace Pricing

Search GitHub

Sign in or Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

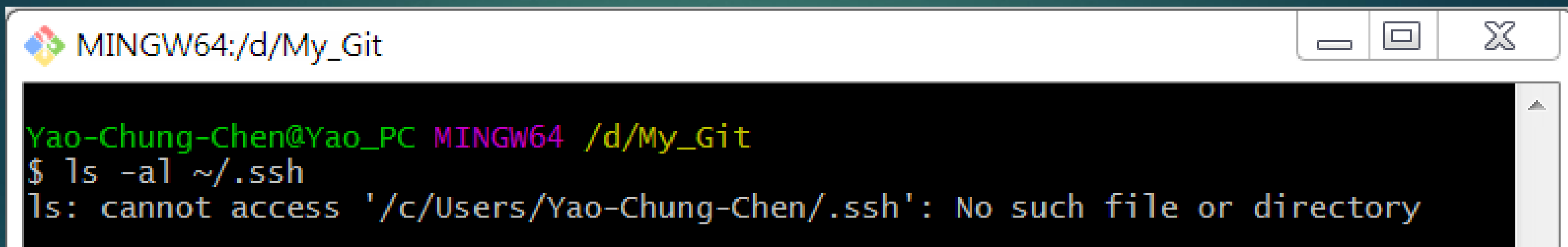
Use at least one letter, one numeral, and seven characters.

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

檢查是否有 SSH key

25



A screenshot of a Windows command prompt window titled "MINGW64:/d/My_Git". The window shows the following text:

```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git
$ ls -al ~/.ssh
ls: cannot access '/c/Users/Yao-Chung-Chen/.ssh': No such file or directory
```

▶ 參考資料

<http://trunk-studio.com/blog/ssh-for-windows/>

製作 SSH key

26

```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git
1 $ ssh-keygen -t rsa -b 4096 -C "GitHub yaochung41@gmail.com"
Generating public/private rsa key pair.
2 Enter file in which to save the key (/c/Users/Yao-Chung-Chen/.ssh/id_rsa):
Created directory '/c/Users/Yao-Chung-Chen/.ssh'.
3 Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Yao-Chung-Chen/.ssh/id_rsa.
Your public key has been saved in /c/Users/Yao-Chung-Chen/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:GIFFDXZs7kNiMKgr/Xkaz/6n4oPPYovYwZV97+QGwZ0 GitHub yaochung41@gmail.com
The key's randomart image is:
+----[RSA 4096]-----+
|  . +*+. |
|  . +. 0+ |
|  . 0.+ . . |
|  . 000= E |
| 0 0.0+S. |
|0... .+. |
| . 0..0 00 |
| 0 0*=+ +0 |
| . 0.00B+0+0 |
+-----[SHA256]-----+
```

- ▶ 1. 在 Git Bash 輸入此行程式碼，記得最後是要寫上自己的 Email
- ▶ 2. 這邊按 Enter 即可
- ▶ 3. 設定你自己想要的 SSH key 密碼(passphrase)；下一行為再次輸入 passphrase 做確認
- ▶ 4. 有跑出下方矩形圖案就代表完成這個設定

確認 SSH-agent 啟動/ 將SSH-key 加入到 GitHub

27

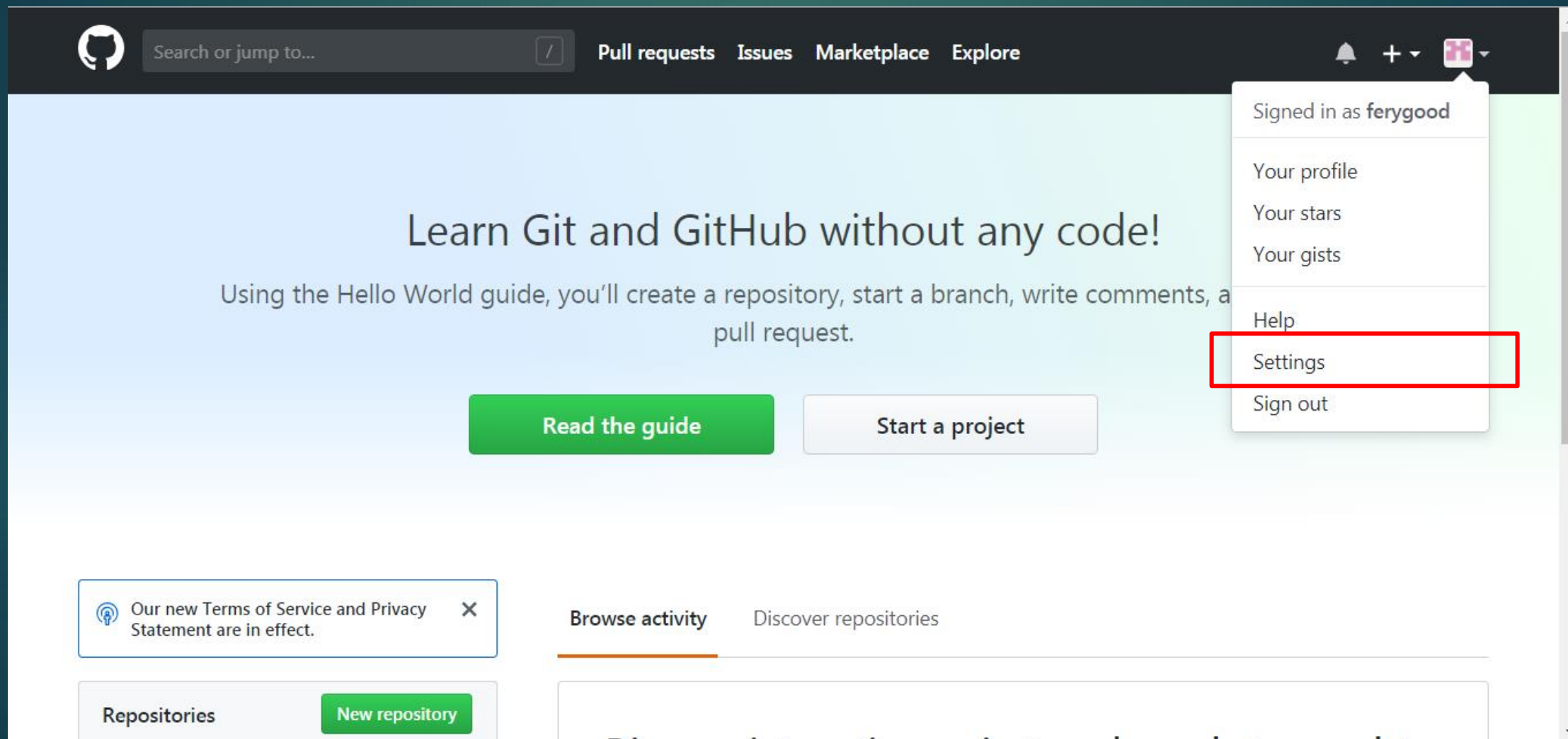
```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git
$ eval "$(ssh-agent -s)"
Agent pid 5008
```

```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git
$ clip < ~/.ssh/id_rsa.pub
```

- ▶ `$ eval "$(ssh-agent -s)"`
確認 ssh-agent 啟動
- ▶ `$ clip < ~/.ssh/id_rsa.pub`
將 SSH key 複製到剪貼簿

登入 GitHub，至 Settings 設定你的 SSH key

28



The screenshot shows the GitHub homepage. At the top, there's a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the right, there's a notification bell, a plus sign, and a user profile icon. The user profile icon is open, showing a dropdown menu with the following options: "Signed in as ferygood", "Your profile", "Your stars", "Your gists", "Help", "Settings" (highlighted with a red rectangle), and "Sign out".

Learn Git and GitHub without any code!

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

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

Our new Terms of Service and Privacy Statement are in effect.

[Browse activity](#) [Discover repositories](#)

Repositories [New repository](#)

選擇左方欄位 SSH and GPG keys

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Public profile

Name

Public email

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

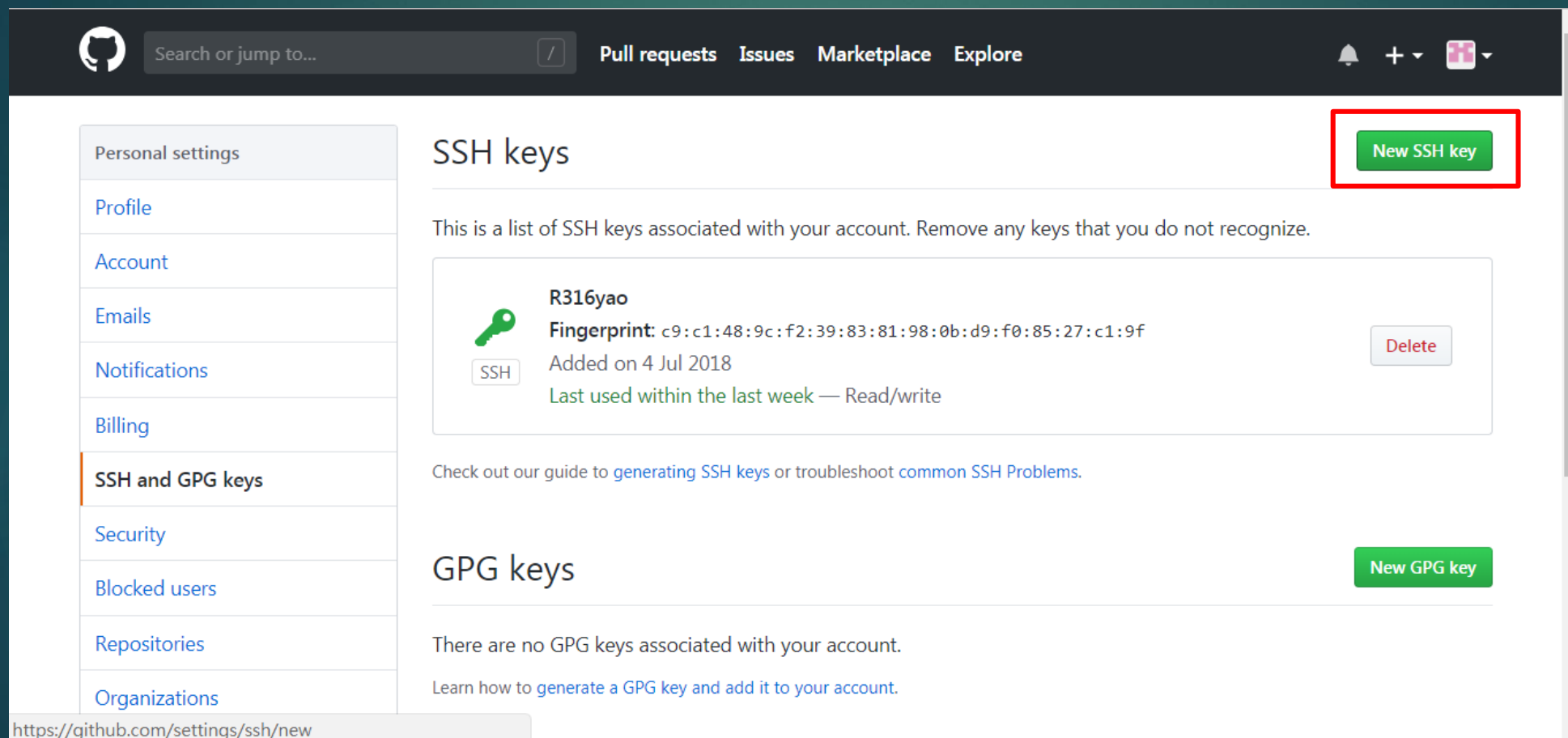
Profile picture

Upload new picture

<https://github.com/settings/keys>

點擊右上方 New SSH key

30



The screenshot shows the GitHub 'SSH keys' settings page. On the left is a sidebar with navigation links: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys (highlighted), Security, Blocked users, Repositories, and Organizations. The main content area is titled 'SSH keys' and features a green 'New SSH key' button in the top right corner, which is enclosed in a red rectangular box. Below the title, a message states: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' A single SSH key is listed with the title 'R316yao', a fingerprint 'c9:c1:48:9c:f2:39:83:81:98:0b:d9:f0:85:27:c1:9f', and the text 'Added on 4 Jul 2018' and 'Last used within the last week — Read/write'. A 'Delete' button is located to the right of the key details. Below the key list, there is a link to 'generating SSH keys' and a link to 'common SSH Problems'. At the bottom of the main content area, there is a section for 'GPG keys' with a green 'New GPG key' button. A message states: 'There are no GPG keys associated with your account.' and a link to 'generate a GPG key and add it to your account.' The browser address bar at the bottom shows the URL 'https://github.com/settings/ssh/new'.

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users


Repositories

Organizations

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 **R316yao**
Fingerprint: c9:c1:48:9c:f2:39:83:81:98:0b:d9:f0:85:27:c1:9f
Added on 4 Jul 2018
Last used within the last week — Read/write

SSH

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

<https://github.com/settings/ssh/new>

新增 SSH keys 到你的 GitHub 帳號

31

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

SSH keys / Add new

Title

Home_yao

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCTMu4wPFjqvROO5MxZtyIUjcsBahckmgZWPgsriz2nttRd6dDG+6kbGt
dU06838zO05oEGJw99WFX1AysZv+IS6nBliz9431ylBsVaDZSwPs4JoYUUqOO5Z6t6kzFgdAvLtPgGysoaqOa2HbKN
+0mixTNGxHs4jnM3kbWhFD8Re1K/TUpZfzxGbEL9tIU5ua6MIDZePbh/odzAXL46cV/EINGTyhzMwy191UdB3AlpyR
3zm9mX+1cpid/YUVmYwD2l+F/q3uD0CmZyWta/oeFyPbx8OFnnEq6sTxdP+EscacI+oya0v9fuPx0NqEWN7U5flqP
Rwu7h9V7XWnyayu2GCSuMs5T6MZAe0yC2IVXXqRTniMPB1Ofy2aRba3MptqP4lWdfRzx+N1+JPV59Z+d8CBAEfu
mTRks7ZMw2yF5J5RJZebQ64HLXYTrqKJcfpOBfDHensHLH6nWfkiAZNJFdyDxce7B+TXV1tkWOusjxOic+cPwmS1q
7oKTq91zxih8YdGa7vN8aKwMIUkMdoAulk2lkaTeAdlKoW/X1Y3O84fL8dftZ3yRMcioBpybtjdE/TOPjThvaaqbybSX
deaZTQhtSZlpT6U3juwa/fQt+KKjz0/QsuRoWa6MWYd0/RCmlijYyqqmiBJkfUTb7ZDnZGBXJJKNS3fnSlhoLrTQ==
GitHub yaochung41@gmail.com
```

Add SSH key

- ▶ Title: 命名你的 SSH key (可取任意喜愛的名稱)
- ▶ Key: 記得我們在投影片 #27 有複製 key 到剪貼簿嗎？在這裡直接在 key 的位置點擊滑鼠右鍵複製貼上
- ▶ 完成 Title 和 Key 的設定之後，就可以點擊下方的 Add SSH key

測試 SSH key 並連上 GitHub

```
Yao-Chung-Chen@Yao_PC MINGW64 /d/My_Git
1 $ ssh -T git@github.com
The authenticity of host 'github.com (192.30.253.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
2 Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.113' (RSA) to the list of known hosts.
3 Enter passphrase for key '/c/Users/Yao-Chung-Chen/.ssh/id_rsa':
Hi ferygood! You've successfully authenticated, but GitHub does not provide shell access.
```


- 1.輸入此行進行 GitHub 連結
- 2.輸入 yes (確定要連結)
- 3.輸入你當初設定的 SSH key passphrase

連結成功的 SSH keys 圖案會變成綠色的，反之為黑色。

SSH keys

New SSH key


This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



SSH

R316yao
Fingerprint: c9:c1:48:9c:f2:39:83:81:98:0b:d9:f0:85:27:c1:9f
Added on 4 Jul 2018
Last used within the last week — Read/write

Delete



SSH

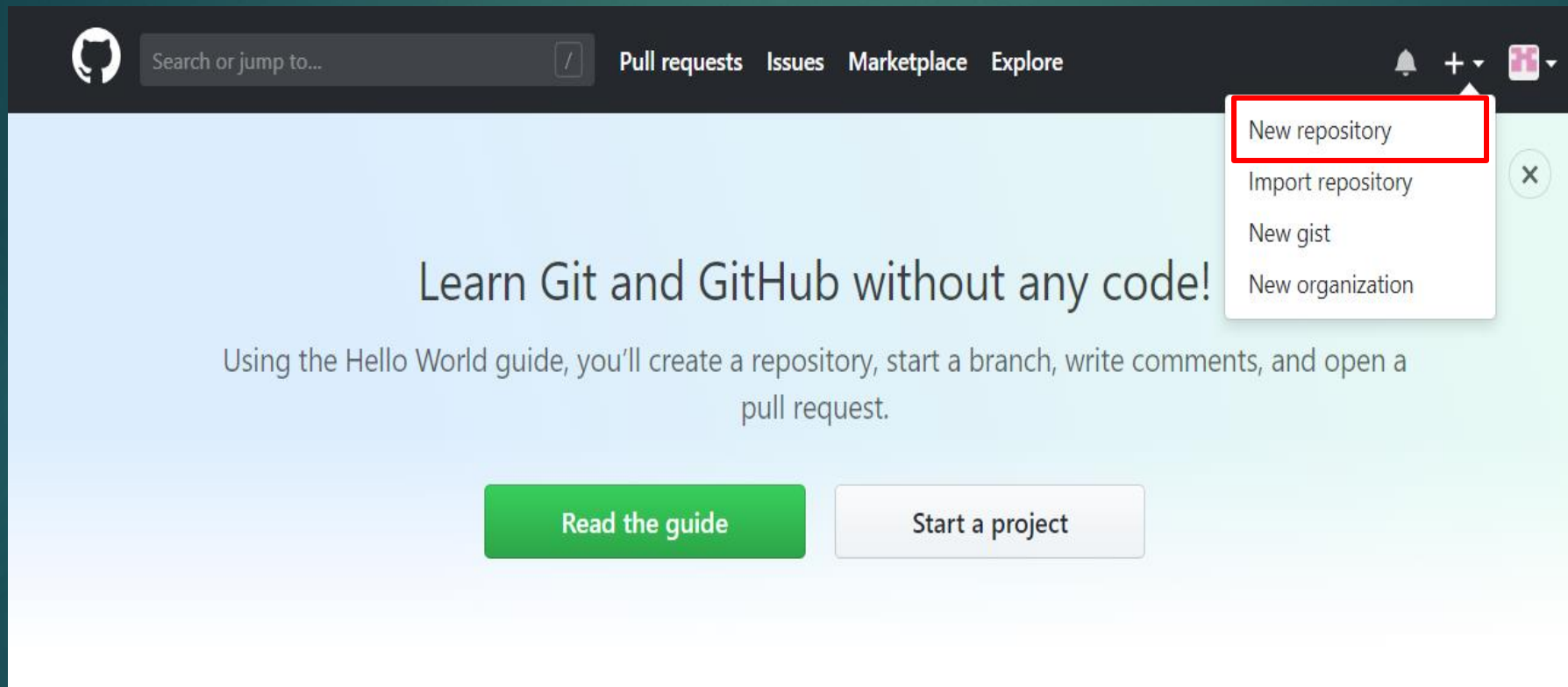
Home_yao
Fingerprint: e2:21:bf:6f:99:79:da:c4:49:c3:af:ee:f7:34:e2:3c
Added on 11 Jul 2018
Last used within the last week — Read/write

Delete

Push 上傳到 GitHub

34

- ▶ 在 GitHub 上開新專案：要把檔案上傳到 GitHub 需要先在上面開一個新的專案，如圖：



新增 Repository

35

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: ferygood / Repository name: Rosalind Bioinformatics Armory ✓

Great repository names: Your new repository will be created as Rosalind-Bioinformatics-Armory-system.

Description (optional): This repository is for the codes to answer Rosalind Bioinformatics Army question.

☒ Public: Anyone can see this repository. You choose who can commit.

☐ Private: You choose who can see and commit to this repository.

☐ Initialize this repository with a README: This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None ⓘ

Create repository

- ▶ Repository name > 任意名稱
- ▶ Description > 紀錄這個 repository 內容的描述
- ▶ Public > Open source (Free)

上傳檔案 / Push 步驟

36

ferygood / Rosalind-Bioinformatics-Armory

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

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

Set up in Desktop or HTTPS SSH git@github.com:ferygood/Rosalind-Bioinformatics-Armory.git

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

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

```
echo "# Rosalind-Bioinformatics-Armory" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:ferygood/Rosalind-Bioinformatics-Armory.git
git push -u origin master
```

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

```
git remote add origin git@github.com:ferygood/Rosalind-Bioinformatics-Armory.git
git push -u origin master
```

...or import code from another repository

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

Import code

- 1) 先點進你想要建立連結的資料夾中，於空白處點擊滑鼠右鍵開啟 **Git Bash**
- 2) 在 **Git Bash** 內依序輸入紅色框中的程式碼。
特別注意 `$ git remote` 那一行
每個人會依據其創建資料夾名稱不同而不一樣。
- 3) 輸入到最後一行 `$git push` 時
會要求你輸入 `ssh key` 的密碼，
輸入完密碼鍵入 **Enter** 按鍵，
及完成將 `README.md` 檔案上傳的動作。

檢視是否有上傳 (Push) 成功

37

Rosalind_Bioinformatics_Armory

檔案 常用 共用 檢視

← → ↕ ↑ > 本機 > DATA (D:) > R316yao > My_Git > Rosalind_Bioinformatics_Armory

名稱	修改日期	類型
.git	2018/7/11 下午 0...	檔案資料夾
001 INI Introduction to the Bioinformatics Armory	2018/7/11 下午 0...	檔案資料夾
002 DBPR Introduction to Protein Databases	2018/7/11 下午 0...	檔案資料夾
003 GBK GeneBank Introduction	2018/7/11 下午 0...	檔案資料夾
004 FRMT Data Formats	2018/7/11 下午 0...	檔案資料夾
005 MEME New Motif Discovery	2018/7/11 下午 0...	檔案資料夾
006 NEED Pairwise Global Alignment	2018/7/11 下午 0...	檔案資料夾
007 TFSQ FASTQ format introduction	2018/7/11 下午 0...	檔案資料夾
008 PTRR Protein Translation	2018/7/11 下午 0...	檔案資料夾
009 PHRE Read Quality Distribution	2018/7/11 下午 0...	檔案資料夾
010 FILT Read Filtration by Quality	2018/7/11 下午 0...	檔案資料夾
011 RVCO Complementing a Strand of DNA	2018/7/11 下午 0...	檔案資料夾
012 SUBO Suboptimal Local Alignment	2018/7/11 下午 0...	檔案資料夾
013 BPHR Base Quality Distribution	2018/7/11 下午 0...	檔案資料夾
014 CLUS Global Multiple Alignment	2018/7/11 下午 0...	檔案資料夾
015 ORFR Finding Genes with ORFs	2018/7/11 下午 0...	檔案資料夾
016 BFIL Base Filtration by Quality	2018/7/11 下午 0...	檔案資料夾
README.md	2018/7/11 下午 0...	MD 檔案

ferygood / Rosalind-Bioinformatics-Armory

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

This repository is for the codes to answer Rosalind Bioinformatics Army question. Edit

Add topics

19 commits 1 branch 0 releases 1 contributor

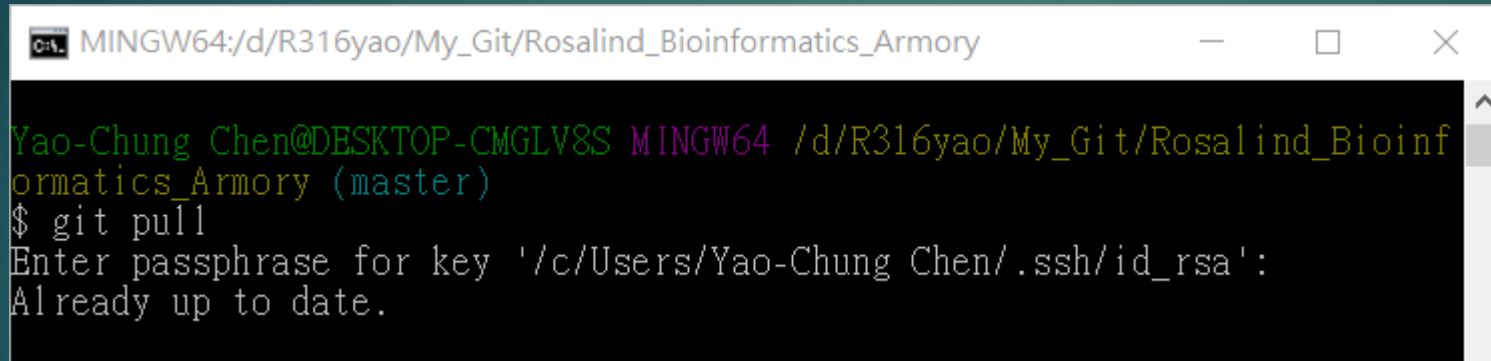
Branch: master New pull request Create new file Upload files Find file Clone or download

ferygood Merge branch 'master' of github.com:ferygood/Rosalind-Bioinformatics-Armory Latest commit bf0663c 29 minutes ago

001 INI Introduction to the Bioinformatics Armory	vim	44 minutes ago
002 DBPR Introduction to Protein Databases	add a print python file in each repository to test	30 minutes ago
003 GBK GeneBank Introduction	add a print python file in each repository to test	30 minutes ago
004 FRMT Data Formats	add a print python file in each repository to test	30 minutes ago
005 MEME New Motif Discovery	add a print python file in each repository to test	30 minutes ago
006 NEED Pairwise Global Alignment	add a print python file in each repository to test	30 minutes ago
007 TFSQ FASTQ format introduction	add a print python file in each repository to test	30 minutes ago
008 PTRR Protein Translation	add a print python file in each repository to test	30 minutes ago
009 PHRE Read Quality Distribution	add a print python file in each repository to test	30 minutes ago
010 FILT Read Filtration by Quality	add a print python file in each repository to test	30 minutes ago
011 RVCO Complementing a Strand of DNA	add a print python file in each repository to test	30 minutes ago
012 SUBO Suboptimal Local Alignment	add a print python file in each repository to test	30 minutes ago
013 BPHR Base Quality Distribution	add a print python file in each repository to test	30 minutes ago
014 CLUS Global Multiple Alignment	add a print python file in each repository to test	30 minutes ago
015 ORFR Finding Genes with ORFs	add a print python file in each repository to test	30 minutes ago
016 BFIL Base Filtration by Quality	add a print python file in each repository to test	30 minutes ago
README.md	add README.md file	38 minutes ago

將檔案從 GitHub 下載 (pull) 至本機資料夾

- ▶ 指令為 `$git pull`
輸入完指令會要求輸入你的 SSH key passphrase。

A screenshot of a Windows terminal window titled 'MINGW64:/d/R316yao/My_Git/Rosalind_Bioinformatics_Armory'. The terminal shows the user 'Yao-Chung Chen@DESKTOP-CMGLV8S' in the 'MINGW64' environment at the directory '/d/R316yao/My_Git/Rosalind_Bioinformatics_Armory (master)'. The user enters the command '\$ git pull'. The terminal then prompts 'Enter passphrase for key '/c/Users/Yao-Chung Chen/.ssh/id_rsa':'. The user's input is not visible, but the terminal responds with 'Already up to date.'.

```
C:\> MINGW64:/d/R316yao/My_Git/Rosalind_Bioinformatics_Armory
Yao-Chung Chen@DESKTOP-CMGLV8S MINGW64 /d/R316yao/My_Git/Rosalind_Bioinformatics_Armory (master)
$ git pull
Enter passphrase for key '/c/Users/Yao-Chung Chen/.ssh/id_rsa':
Already up to date.
```