Computer Programming II Lab

張麒竑

Feb 17, 2022

助教們

- 張麒竑
- 簡傑
- 江宗樺
- 許凱竣
- 吳武峰

Outline

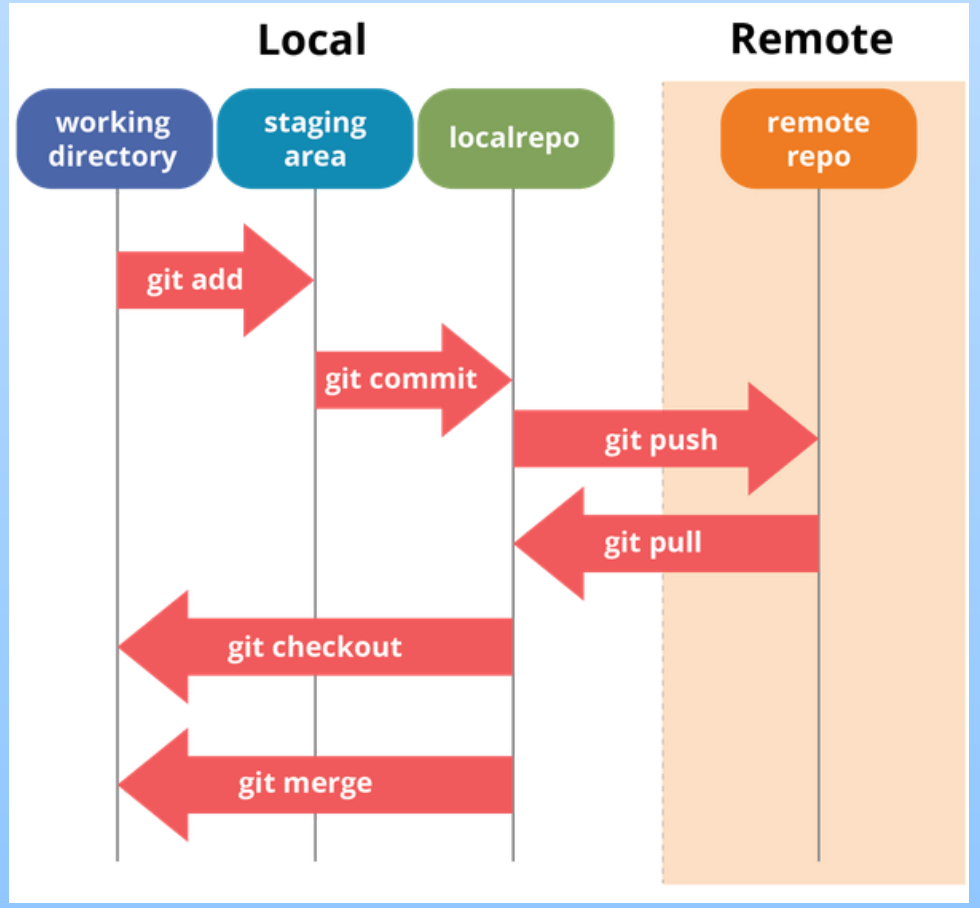
- 版本控制系統 - Git
  - Git
  - GitHub
  - 基本操作
  - 參考資料
- LeetCode
  - 辦個帳號來玩玩看吧
- 歡樂的解題時間(?)
  - LeetCode 1. Two Sum
  - OJ

版本控制系統 - Git

Git

- Git 可以把檔案的狀態作為更新歷史記錄保存起來。因此可以把編輯過的檔案復原到以前的狀態，也可以顯示編輯過內容的差異。
- 當有人想將編輯過的舊檔案上傳到伺服器、覆蓋其他人的最新檔案時，系統會發出警告，因此可以避免在無意中覆蓋他人的編輯內容。

Git


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout
git merge

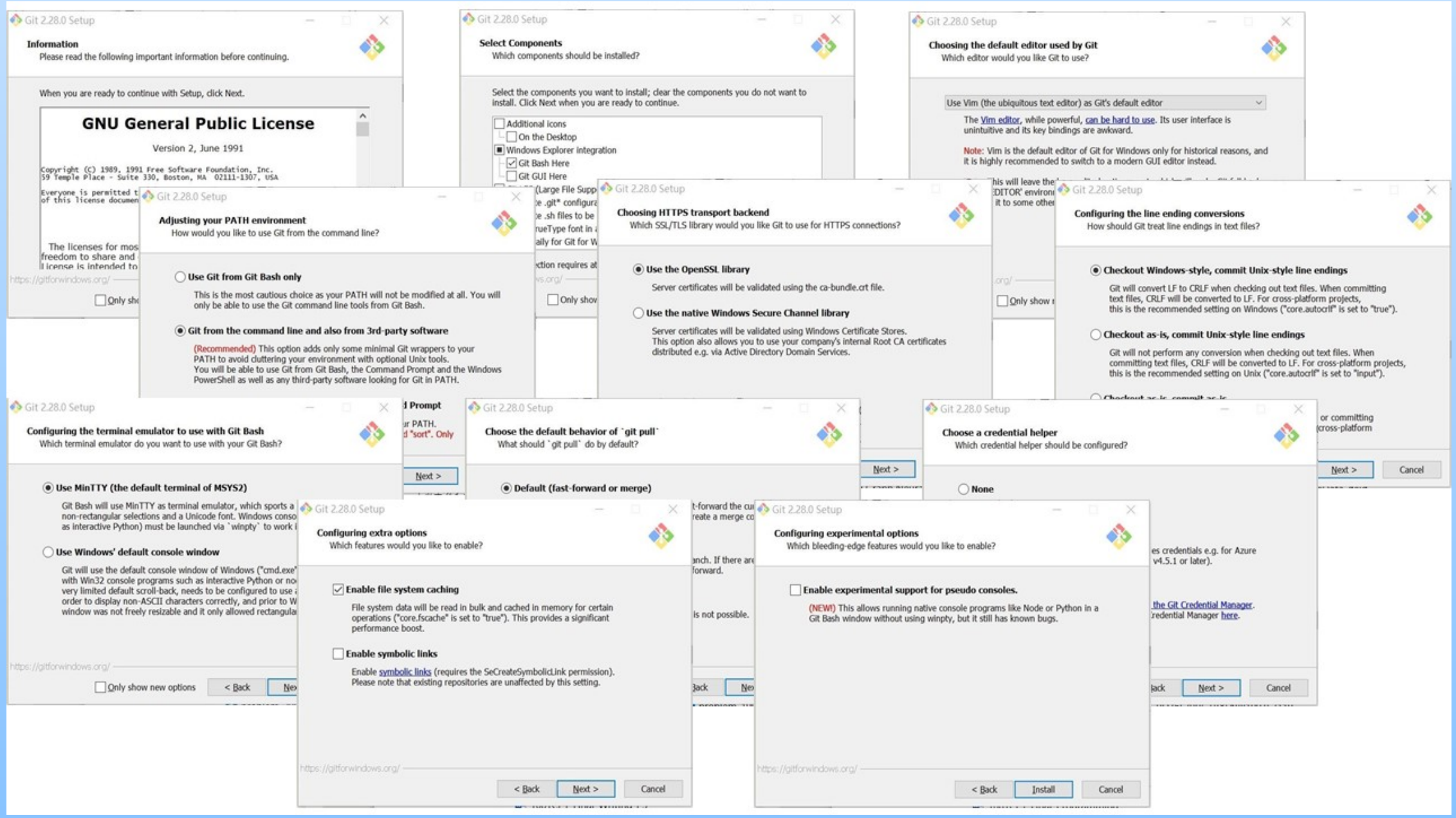
安裝 Git

- Windows
  https://git-scm.com/downloads
- MacOS

```
brew install git
```

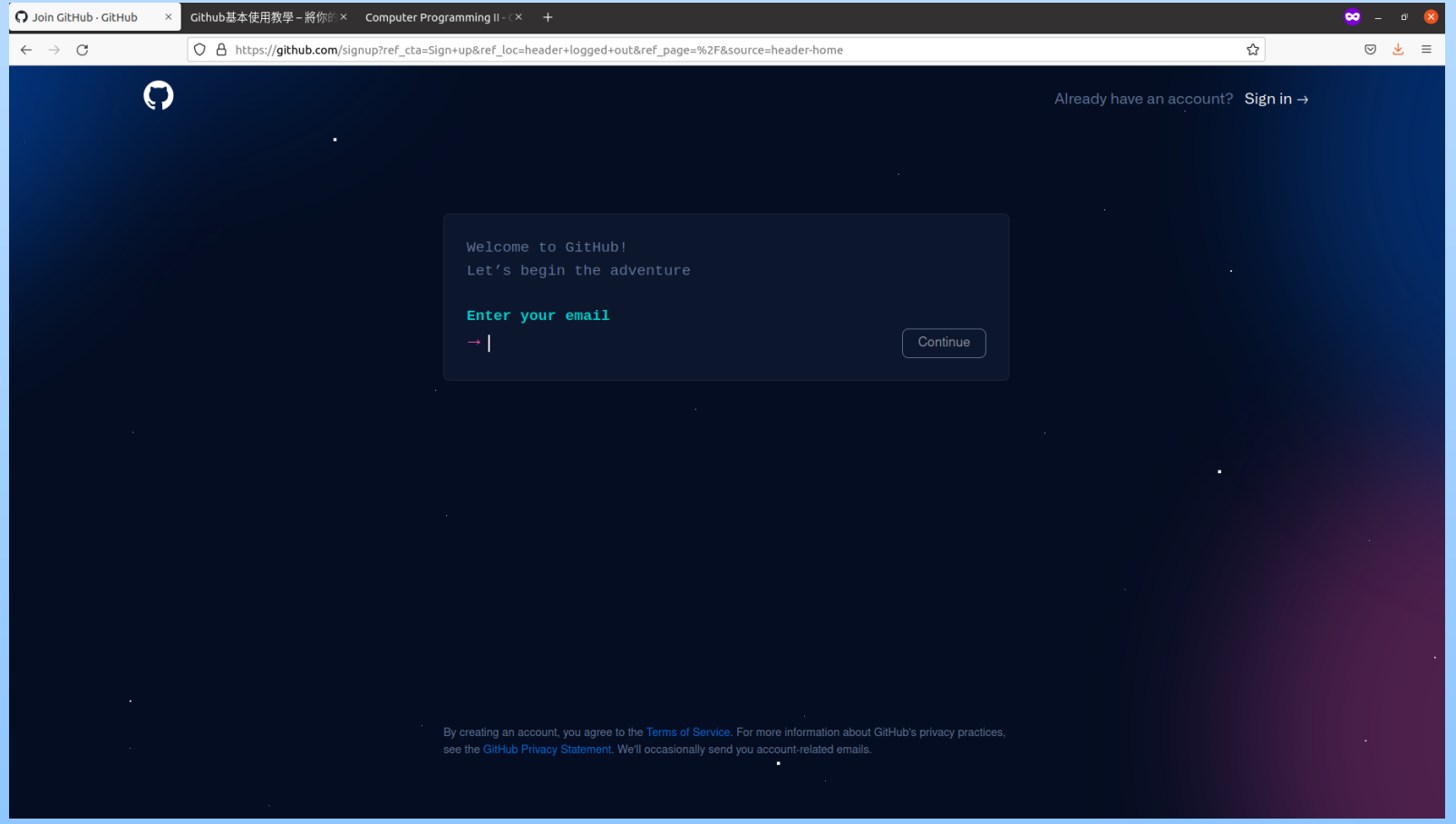
- Linux (Ex. Ubuntu)

```
apt-get install git
```

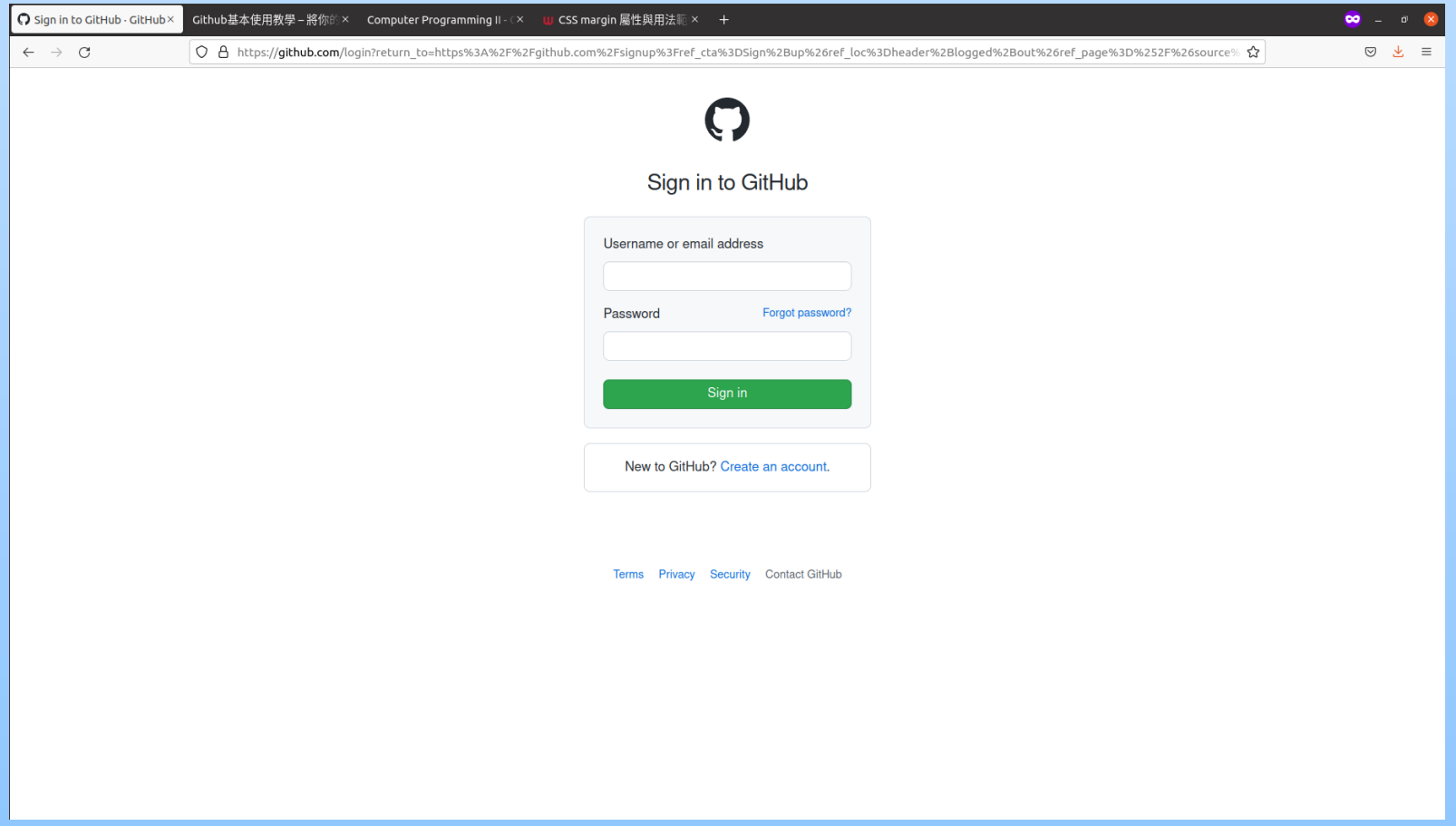

Git 2.28.0 Setup
Information
Please read the following important information before continuing.
When you are ready to continue with Setup, click Next.
GNU General Public License
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Additional icons
On the Desktop
Windows Explorer integration
Git Bash Here
Git GUI Here
Choosing the default editor used by Git
Which editor would you like Git to use?
Use Vim (the ubiquitous text editor) as Git's default editor
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.
Note: Vim is the default editor of Git for Windows only for historical reasons, and it is highly recommended to switch to a modern GUI editor instead.
Adjusting your PATH environment
How would you like to use Git from the command line?
Use Git from Git Bash only
This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
Git from the command line and also from 3rd-party software
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools.
You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.
Choosing HTTPS transport backend
Which SSL/TLS library would you like Git to use for HTTPS connections?
Use the OpenSSL library
Server certificates will be validated using the ca-bundle.crt file.
Use the native Windows Secure Channel library
Server certificates will be validated using Windows Certificate Stores. This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.
Configuring the line ending conversions
How should Git treat line endings in text files?
Checkout Windows-style, commit Unix-style line endings
Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
Checkout as-is, commit Unix-style line endings
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").
Configuring the terminal emulator to use with Git Bash
Which terminal emulator do you want to use with your Git Bash?
Use MinTTY (the default terminal of MSYS2)
Use Windows' default console window
Choose the default behavior of `git pull`
What should `git pull` do by default?
Default (fast-forward or merge)
Choose a credential helper
Which credential helper should be configured?
None
Configuring extra options
Which features would you like to enable?
Enable file system caching
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.
Enable symbolic links
Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.
Configuring experimental options
Which bleeding-edge features would you like to enable?
Enable experimental support for pseudo consoles.
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.
https://gitforwindows.org/
Only show new options
< Back
Next >
Cancel
Install

在 GitHub 註冊帳號

https://github.com/



登入你的 GitHub 帳號



設定你的 Local 端配置

```
git config --global user.name <your_name>
git config --global user.email <your_email>
```

```
(^o^) [cch] [~/Documents/1102cp2_lab/Lab_20220217] $ git config --global user.name "Chi-Hung Chang"
(^o^) [cch] [~/Documents/1102cp2_lab/Lab_20220217] $ git config --global user.email "chihung861224@gmail.com"
(^o^) [cch] [~/Documents/1102cp2_lab/Lab_20220217] $ git config --list
user.name=Chi-Hung Chang
user.email=chihung861224@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=git@github.com:chang861224/1102cp2_lab.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
(^o^) [cch] [~/Documents/1102cp2_lab/Lab_20220217] $
```

設定 SSH-KEY

```
ssh-keygen -t rsa -C <your_email>
```

```
(^o^) [cch] [~/Documents/1102cp2] $ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Chi-Hung Chang/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Chi-Hung Chang/.ssh/id_rsa
Your public key has been saved in /c/Users/Chi-Hung Chang/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:5cqcb9PnoVsED9KJZfxe/cwgcaytI6F6N1HPgXqi28k Chi-Hung Chang@cch20-pc
The key's randomart image is:
```

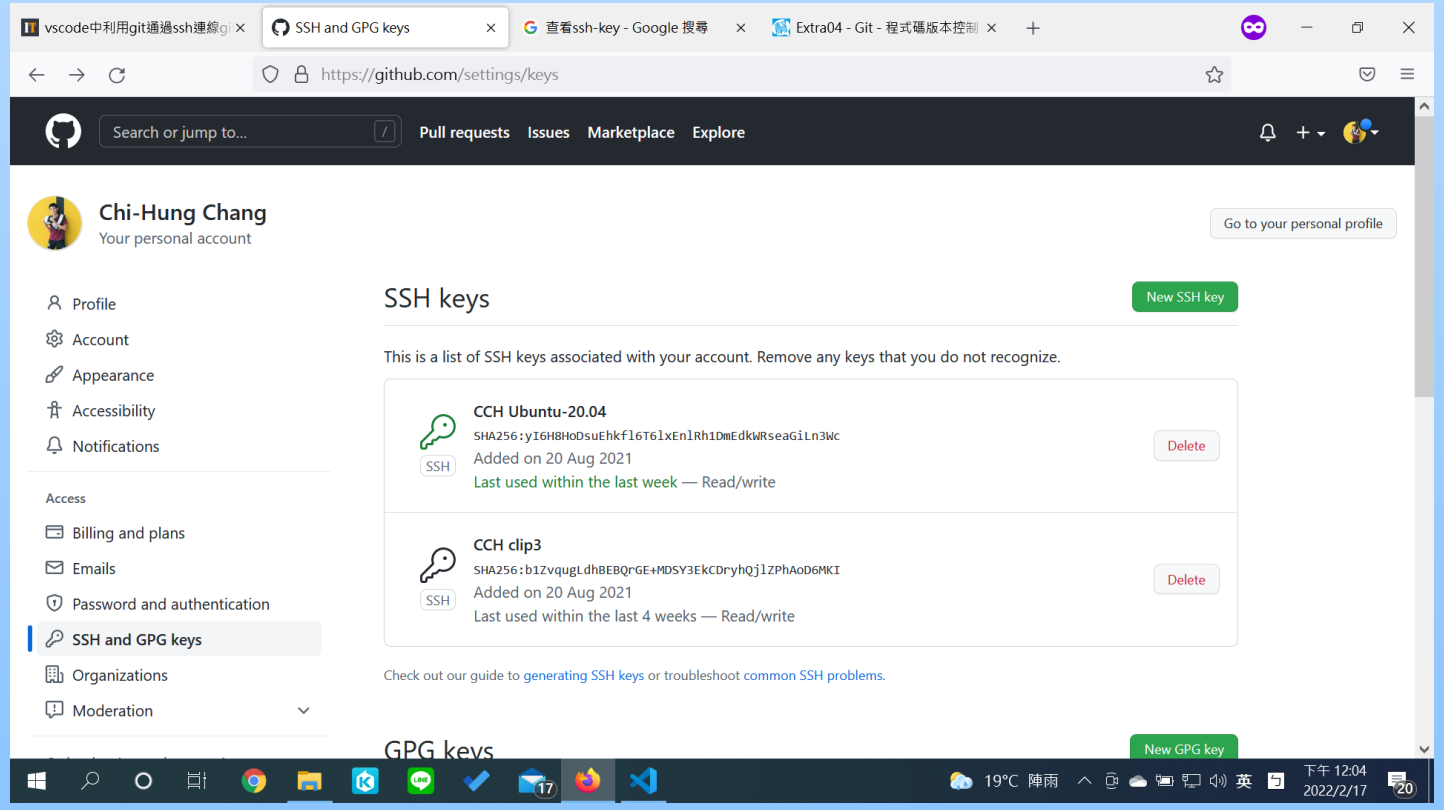
設定 SSH-KEY

在家目錄底下就會出現一個 `.ssh` 資料夾，裡面有兩個檔案，其中 `id_rsa.pub` 存的是 SSH 金鑰

```
(^o^) [cch] [~/Documents/1102cp2] $ cd ~/.ssh
(^o^) [cch] [~/.ssh] $ ls -l
-rw-r--r--  1 cch cch  2610 Feb 17 12:44 id_rsa
-rw-r--r--  1 cch cch   577 Feb 17 12:44 id_rsa.pub
(^o^) [cch] [~/.ssh] $ cat id_rsa.pub
ssh-rsa ..................................... Chi-Hung Chang @cch20-pc
(^o^) [cch] [~/.ssh] $
```

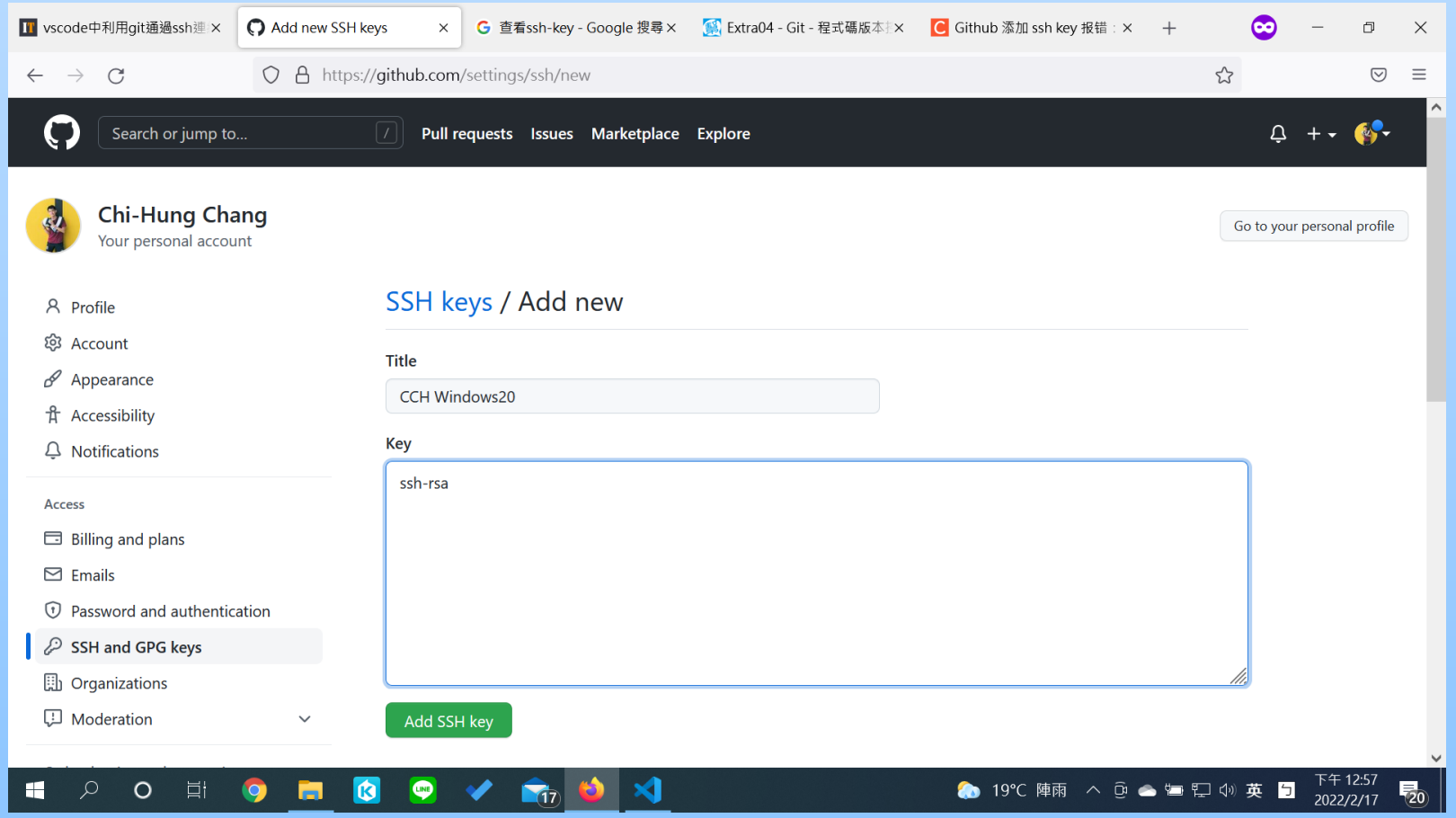
設定 SSH-KEY

- 在你的 GitHub 帳號點選 `setting` （在右上角個人帳號點下去的倒數第二個）
- 點選左側的 `SSH and GPG Keys`



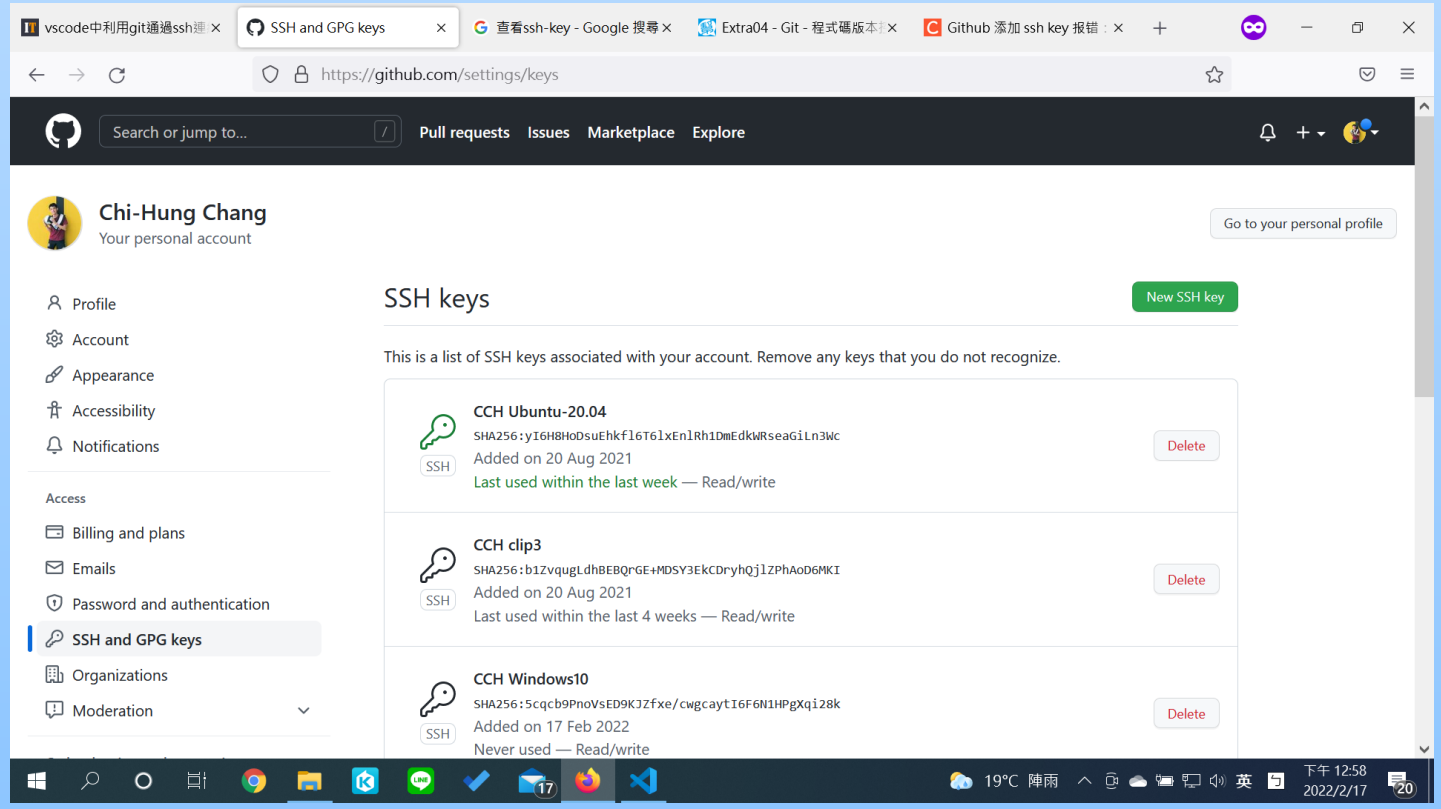
設定 SSH-KEY

點選右上角綠色的 `New SSH key`，並將剛剛在 `id_rsa.pub` 內的 SSH 金鑰複製進去



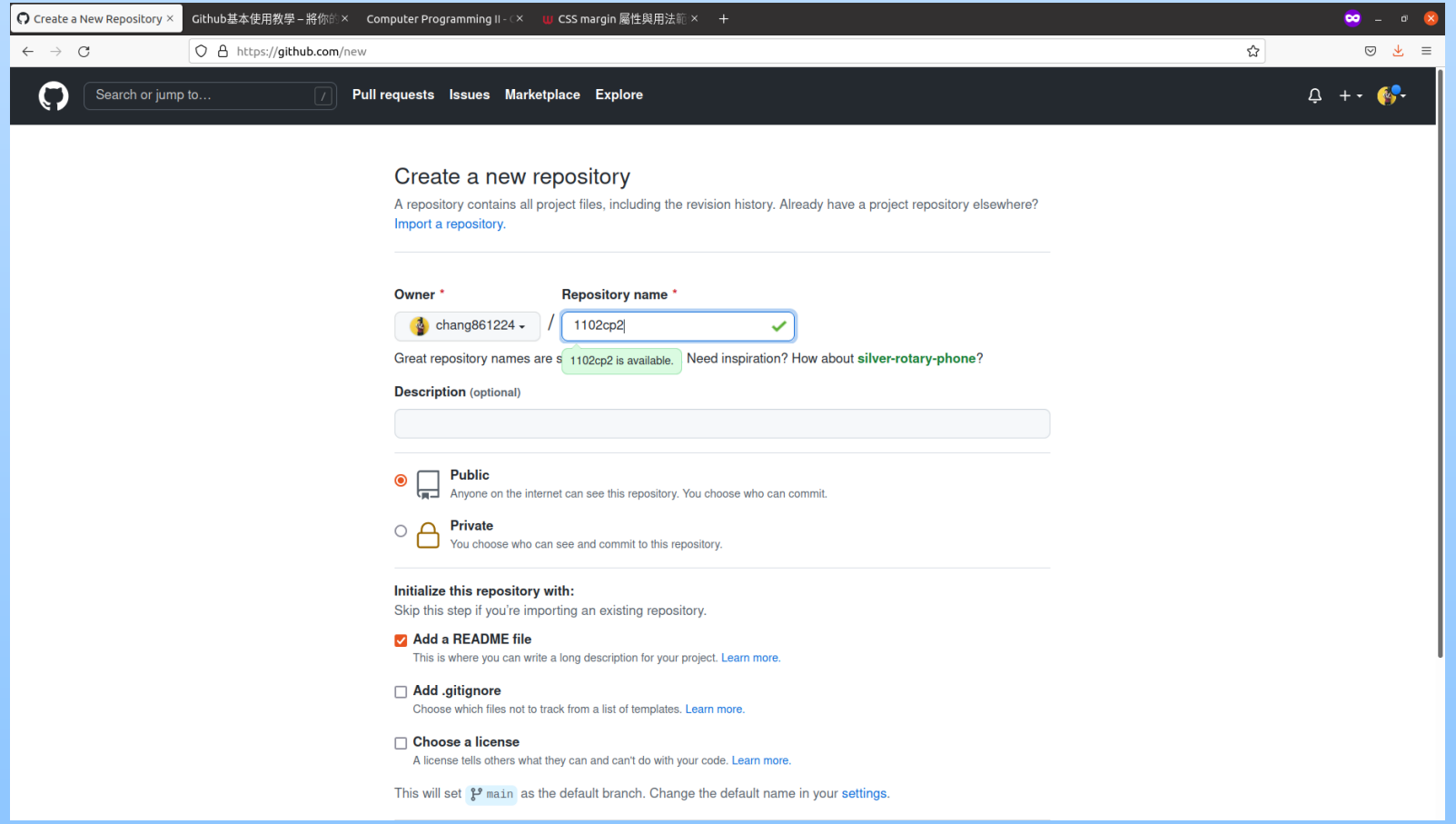
設定 SSH-KEY

完成後按 `Add SSH key` ，就會回到 SSH keys的頁面，你剛剛新增的 SSH key 也會出現（如果一切正常的話啦）



基本操作

- 在自己的 GitHub 上面創建一個資料夾



基本操作

- 將 GitHub 上面的資料夾下載到自己的電腦 Local 端

```
git clone <repository_url>
```

```
(^o^) [cch] [~/Documents] $ git clone git@github.com:chang861224/1102cp2.git
Cloning into '1102cp2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
(^o^) [cch] [~/Documents] $ ls -l
drwxrwxr-x  3 cch cch  4096 Feb 16 21:45 1102cp2
drwxrwxr-x  5 cch cch  4096 Feb 16 16:27 1102cp2_lab
(^o^) [cch] [~/Documents] $
```

  - 如果是用 git bash 進行操作的話，要選 `HTTP` 的 URL (https://....)
  - 如果是用 terminal 進行操作的話，要選 `SSH` 的 URL (git@github.com:.....)

基本操作

- 進入資料夾，查看資料夾狀態

```
cd <repository_name>
git status
```

```
(^o^) [cch] [~/Documents] $ cd 1102cp2
(^o^) [cch] [~/Documents/1102cp2] $ vim test.c
(^o^) [cch] [~/Documents/1102cp2] $ git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
(^o^) [cch] [~/Documents/1102cp2] $
```

基本操作

- 新增修改的檔案

```
git add <filenames>
```

```
(^o^) [cch] [~/Documents/1102cp2] $ git add test.c
(^o^) [cch] [~/Documents/1102cp2] $ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   test.c

(^o^) [cch] [~/Documents/1102cp2] $
```

基本操作

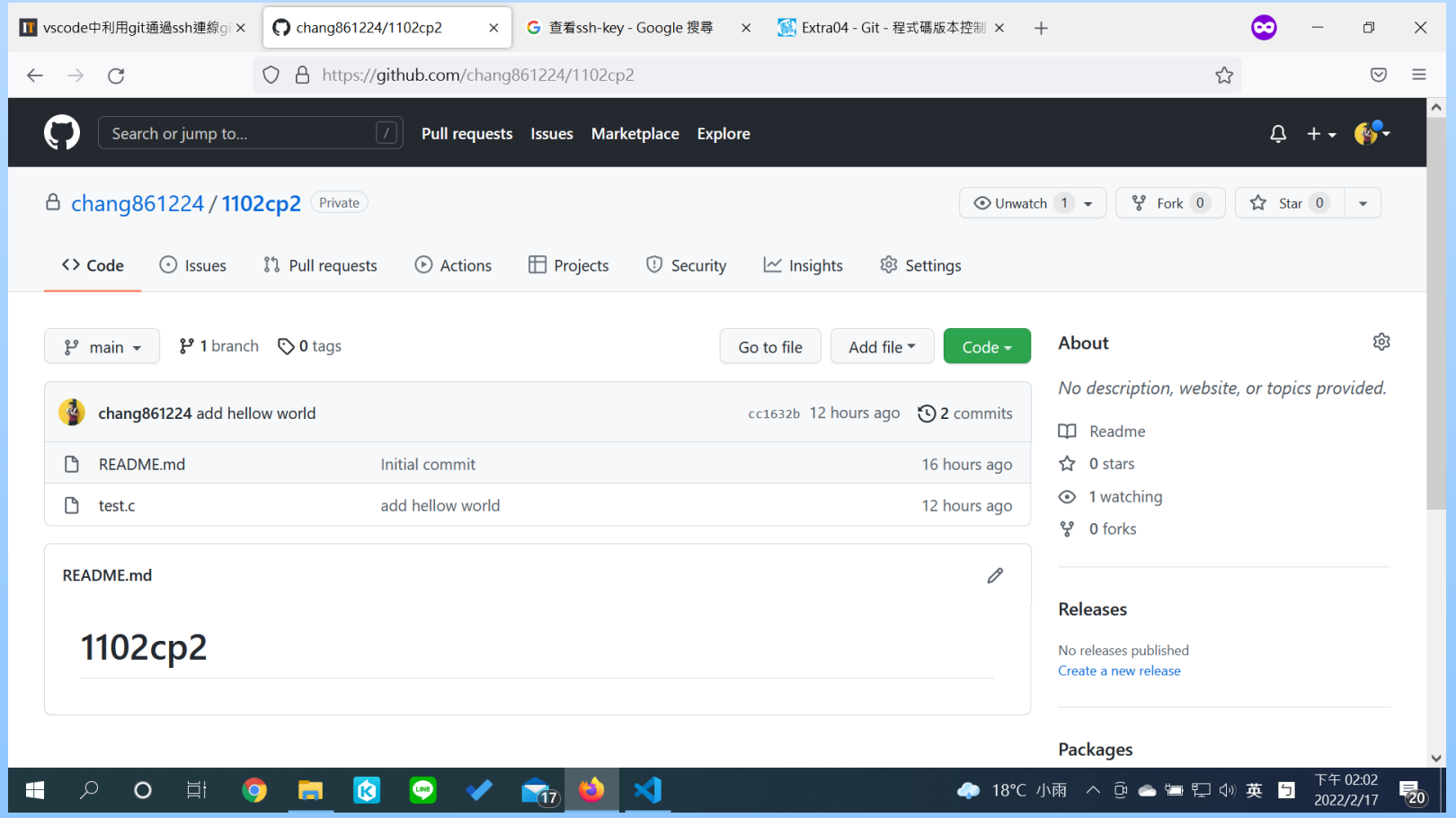
- 新增 commit，然後上傳 GitHub

```
git commit -m <your_commits>
git push origin main
```

```
(^o^) [cch] [~/Documents/1102cp2] $ git commit -m "add hellow world"
[main cc1632b] add hellow world
 1 file changed, 6 insertions(+)
 create mode 100644 test.c
(^o^) [cch] [~/Documents/1102cp2] $ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 355 bytes | 355.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:chang861224/1102cp2.git
   ac0ce95..cc1632b  main -> main
```

基本操作

接下來再去你的 GitHub 的這個資料夾，就可以看到你上傳的檔案



基本操作

- 查看版本歷史紀錄

```
git log
```

```
(^o^) [cch] [~/Documents/1102cp2] $ git log
commit cc1632b294d9460864c459c393796012bb1a6bc1 (HEAD -> main, origin/main, origin/HEAD)
Author: Chi-Hung Chang <chihung861224@gmail.com>
Date:   Thu Feb 17 01:53:40 2022 +0800

    add hellow world

commit ac0ce95c6c8e70bf0790a013ca1b7e55fe6f9091
Author: Chi-Hung Chang <chihung861224@gmail.com>
Date:   Wed Feb 16 21:35:17 2022 +0800

    Initial commit
(^o^) [cch] [~/Documents/1102cp2] $
```

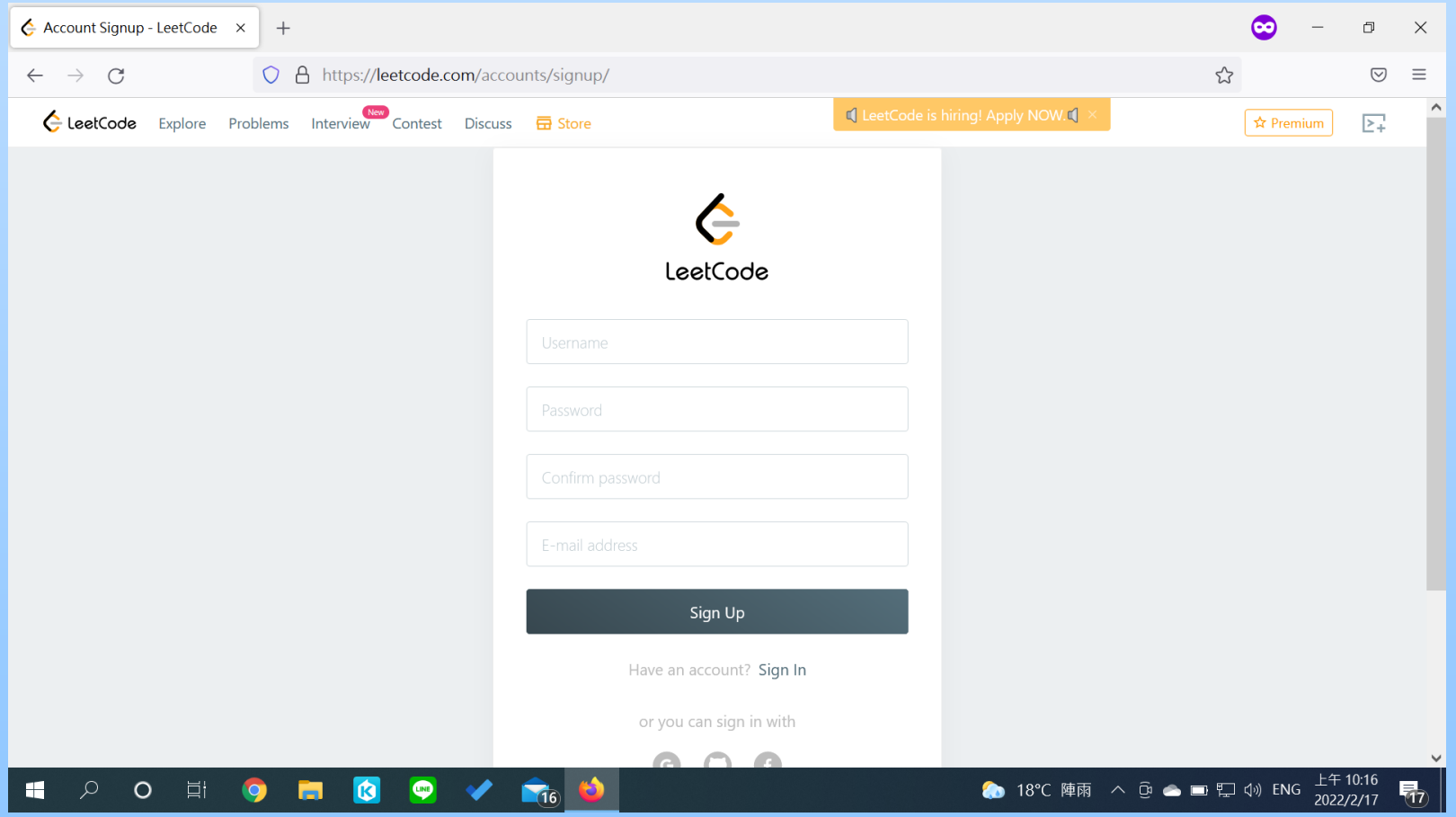
參考資料

- 1.6 開始 - 初次設定 Git
- 2.3 Git 基礎 - 檢視提交的歷史記錄
- 連猴子都能懂的 Git 入門指南
- Git 與 GitHub 版本控制基本指令與操作入門教學
- 30 天精通 Git 版本控管
- vscode 中利用 git 通過 ssh 連線 github 的方法
- Git Tutorial

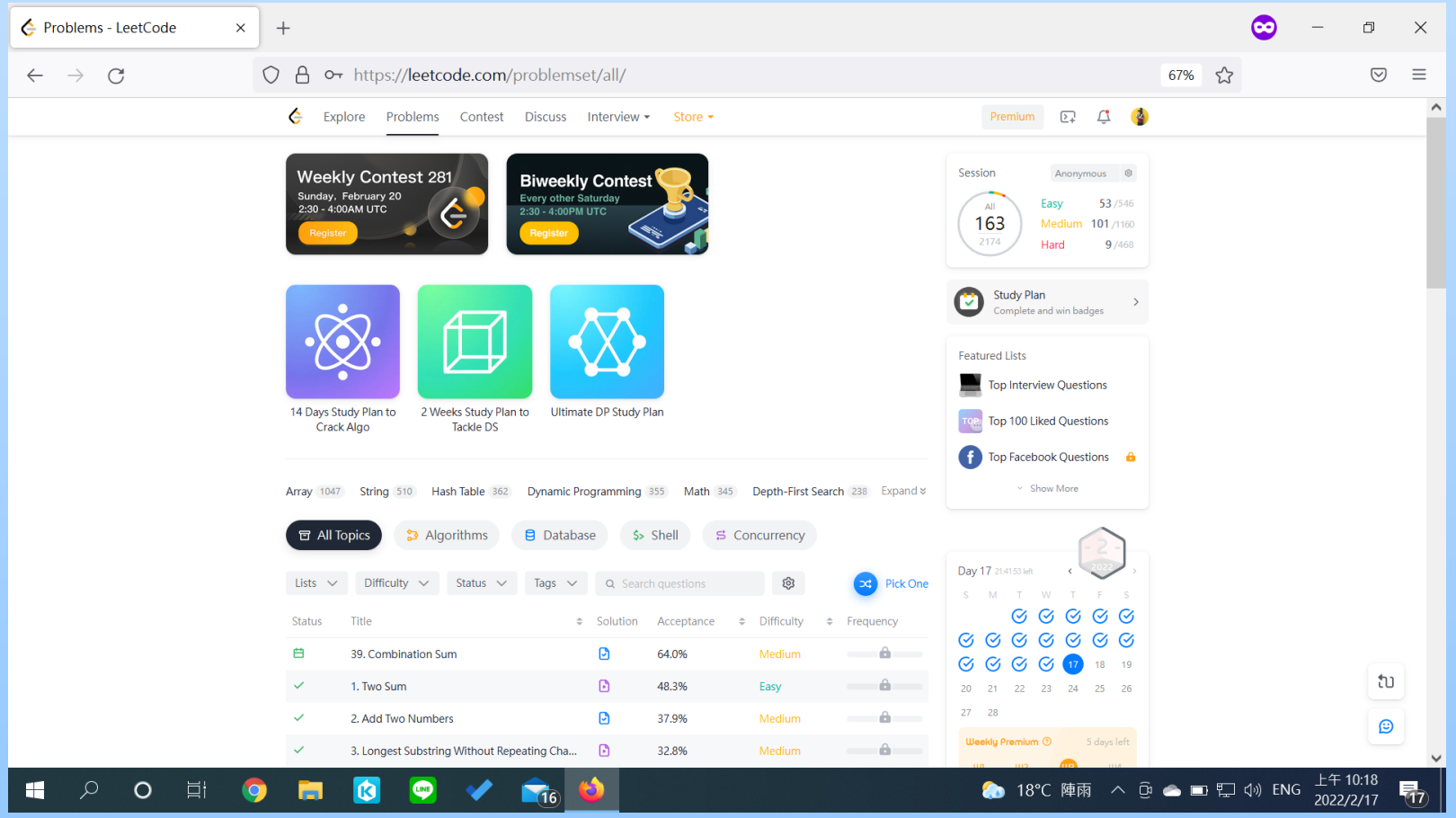
LeetCode

https://leetcode.com/

申請一個帳號


Account Signup - LeetCode
https://leetcode.com/accounts/signup/
LeetCode
Explore
Problems
Interview
Contest
Discuss
Store
LeetCode is hiring! Apply NOW
Premium
LeetCode
Username
Password
Confirm password
E-mail address
Sign Up
Have an account? Sign In
or you can sign in with
18°C 陣雨
ENG
上午 10:16
2022/2/17

題目列表之很多的題目



有興趣，閒閒沒事做的時候就可以隨便找幾題來寫～（追隨老師的腳步XD）

或者可以每天寫一題系統隨機跳出來的隨機任務～

歡樂的解題時間

- LeetCode
- Online Judge

LeetCode 1. Two Sum

給你一個數字的陣列，以及一個 target neumber，你的目標是要在這個數字陣列中找出兩個元素（不能重複取），其相加結果會等於 target number，並回傳這兩個元素的 index。

- 範例輸入：`nums = [2,7,11,15], target = 9`
- 範例輸出：`[0,1]`

題目限制：

- `1 <= nums,length <= 10^4`
- `-10^9 <= nums[i] <= 10^9`
- `-10^9 <= target <= 10^9`
- 只會有唯一一組解！

特別注意：題目是要你寫一個 solution 的 function，不是要你寫出整組程式！

LeetCode 1. Two Sum

解題思路

1. Brute force：兩層 for 迴圈，遇到相加等於 target number 時就回傳
    - 時間複雜度：$O(n^2)$
    - 空間複雜度：$O(1)$
2. hash table：先計算每個數字各出現幾次，再看哪兩個數字和等於 target number
    - 時間複雜度：$O(n)$
    - 空間複雜度：$O(n)$

Online Judge 越大我越愛

有 `n` 個數字存放於陣列 `a` 內，現在想要找兩個數字，`i` 跟 `j`（`i < j`），使得 `a[i] - a[j]` 越大越好。

- 輸入說明
  - 第一行有個數字 `n`，代表現在有 `n` 個數字
  - 接下來一行 `a[1] , a[2] .... , a[n]`
- 輸出說明
  - `a[i] - a[j]` 的最大值

Online Judge 越大我越愛

- 範例輸入

```
5
5 4 3 2 1
```

- 範例輸出

```
4
```

題目限制：

- `2 <= n <= 100000`
- `1 <= a[i], a[j] <= 100000`
- `i < j`

Any Question?