

Git & Github



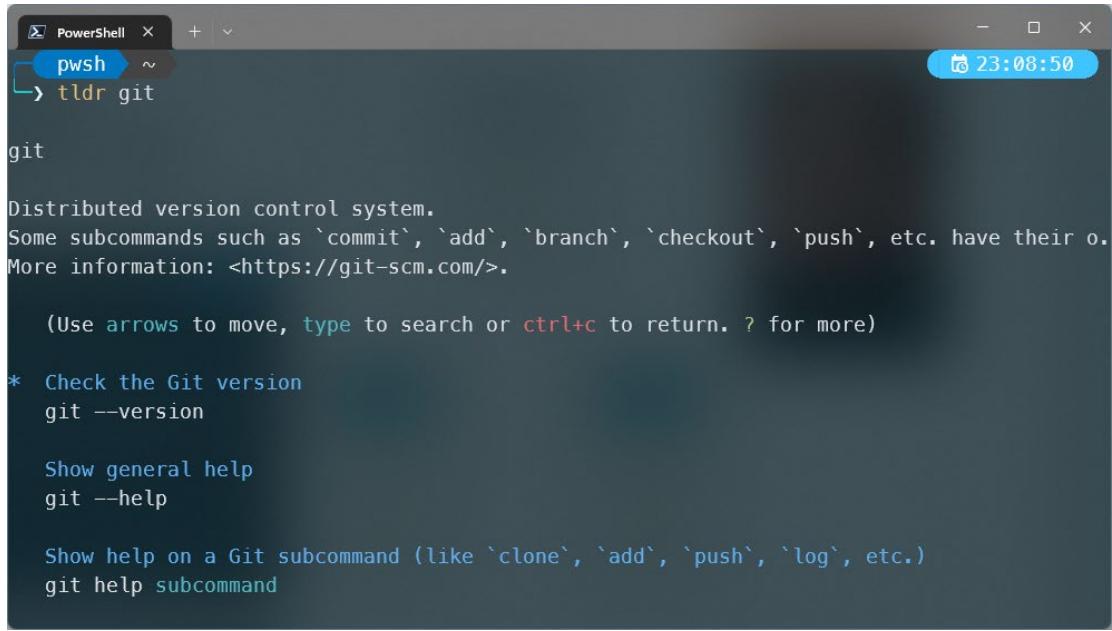
slido



什麼是  git?

什麼是 git ?

- 分散式版本控管系統
- 沒得用了只好自己寫一個?
- 為什麼要用 Git ?



```
pwsh ~
tldr git

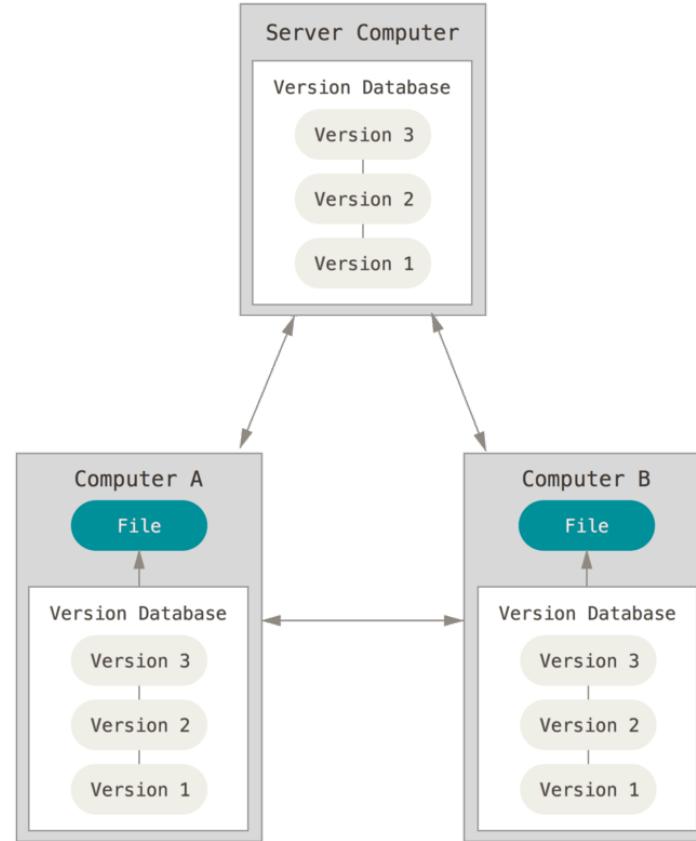
git
Distributed version control system.
Some subcommands such as `commit`, `add`, `branch`, `checkout`, `push`, etc. have their o.
More information: <https://git-scm.com/>.

(Use arrows to move, type to search or ctrl+c to return. ? for more)

* Check the Git version
git --version

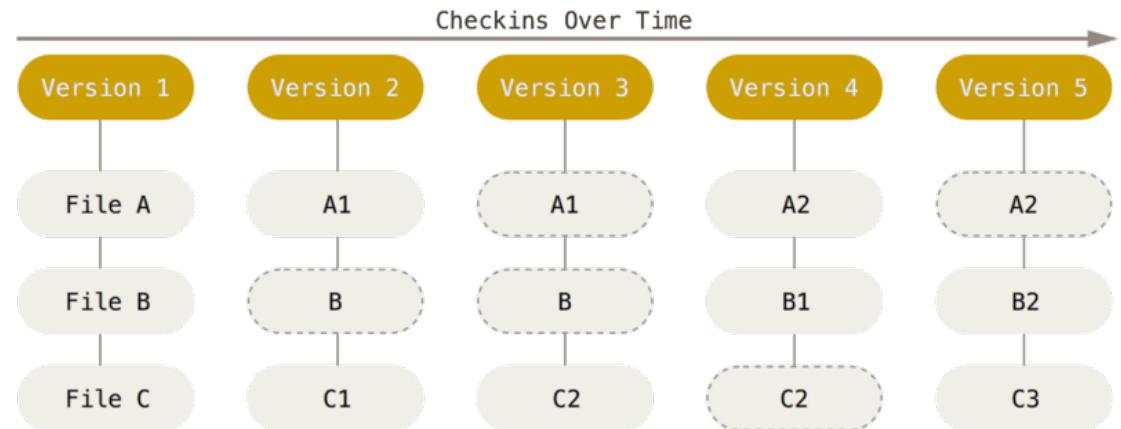
Show general help
git --help

Show help on a Git subcommand (like `clone`, `add`, `push`, `log`, etc.)
git help subcommand
```



什麼是 git ?

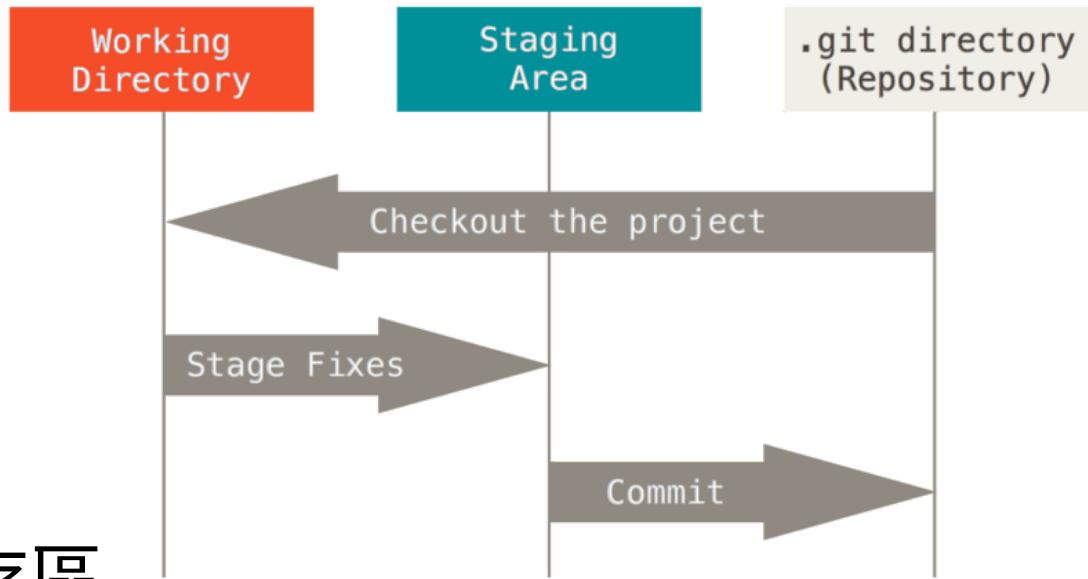
- 快照(snapshot)的紀錄方式
- 只會紀錄有變更的項目



git 工作流程

- 已提交 Committed
- 已預存 Staged
- 已修改 Modified

1. 在工作目錄修改(modify)檔案
2. 將檔案的快照新增(add)到預存區
3. 提交(commit)在預存區的快照到 Git 目錄



從入門到放棄— git 的使用方法

學會看文件

git help commit

git-commit(1) Manual Page

NAME

git-commit - Record changes to the repository

SYNOPSIS

```
git commit [-a | --interactive | --patch] [-s] [-v] [-u<mode>] [--amend]
           [--dry-run] [(-c | -C | --squash) <commit> | --fixup [(amend | reword):]<commit>]
           [-F <file> | -m <msg>] [--reset-author] [--allow-empty]
           [--allow-empty-message] [--no-verify] [-e] [--author=<author>]
           [--date=<date>] [--cleanup=<mode>] [--[no-]status]
           [-i | -o] [--pathspec-from-file=<file> [--pathspec-file-nul]]
           [(--trailer <token>[(=|:)<value>])...]
           [-S<keyid>]
           [-] [<pathspec>...]
```

DESCRIPTION

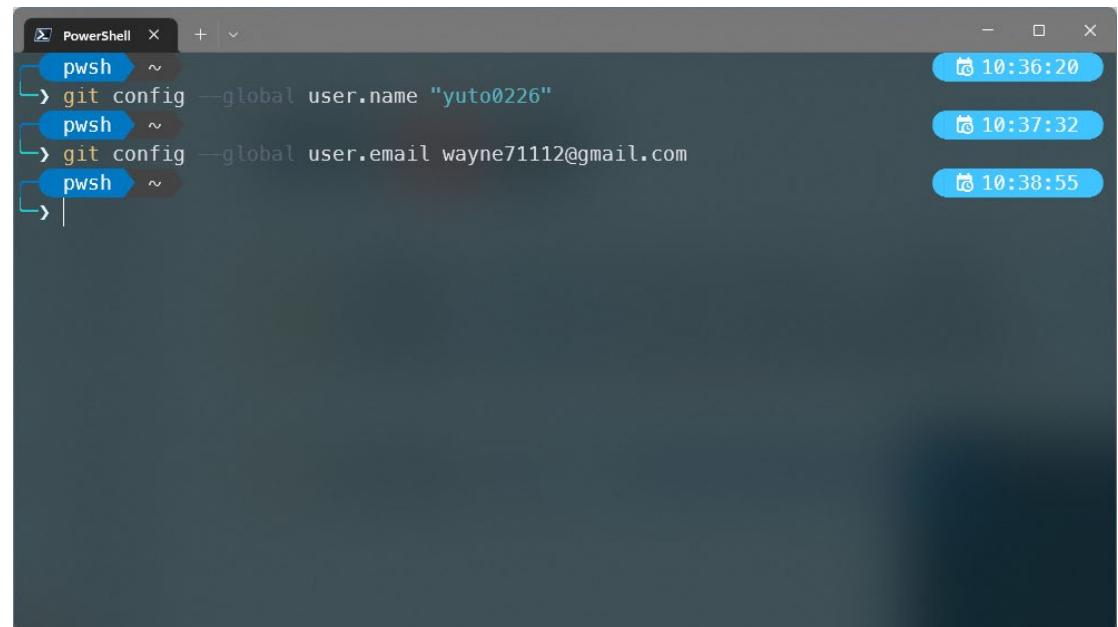
Create a new commit containing the current contents of the index and the given log message describing the changes. The new commit is a direct child of HEAD, usually the tip of the current branch, and the branch is updated to point to it (unless no branch is associated with the working tree, in which case HEAD is "detached" as described in [git-checkout\(1\)](#)).

設定 git

git config --global user.name “名字” # 設定名字

git config --global user.email 電子郵件 # 設定電子郵件

git config --list # 列出所有設定



A screenshot of a PowerShell window titled "PowerShell". The window shows three command-line entries:

- "pwsh ~"
- "git config --global user.name "yuto0226"
- "pwsh ~"
- "git config --global user.email wayne71112@gmail.com"
- "pwsh ~"

Each command is timestamped on the right side of the window:

- git config --global user.name "yuto0226" at 10:36:20
- git config --global user.email wayne71112@gmail.com at 10:37:32
- pwsh ~ at 10:38:55

建立一個 Repository

```
mkdir my_repo
```

建立一個資料夾

```
git init .\my_repo\
```

初始化倉庫(Repository)

```
cd .\my_repo\
```

切換目錄

The screenshot shows a PowerShell window titled 'PowerShell' running on Windows. The session starts with the command `pwsh ~\Desktop`. The user then runs `mkdir my_repo`, which creates a new directory named 'my_repo' at the specified path. A file listing command is run to show the contents of the directory:

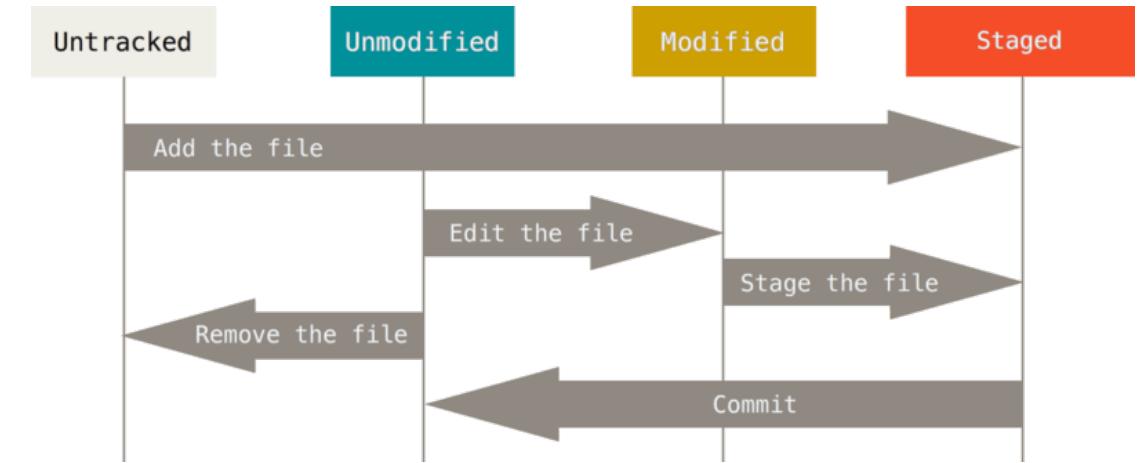
Mode	LastWriteTime	Length	Name
d---	2023/1/8 上午 10:30		my_repo

Next, the user runs `git init .\my_repo\`, which initializes an empty Git repository in the 'my_repo' directory. The output of this command is shown in the terminal. Finally, the user runs `cd .\my_repo\` to change into the repository directory, and then `pwsh ~\Desktop\my_repo` to open a new PowerShell session directly within the repository. The timestamp in the top right corner of the window indicates the sequence of commands was performed between 10:30:24 and 10:33:40.

First commit

`git add <檔案名稱>`

`git commit`



Add new file

```
echo 'hello world' > ttussc.md      # 創建檔案  
cat .\ttussc.md
```

TASK

建立一個 README.md
裡面寫上你的自我介紹



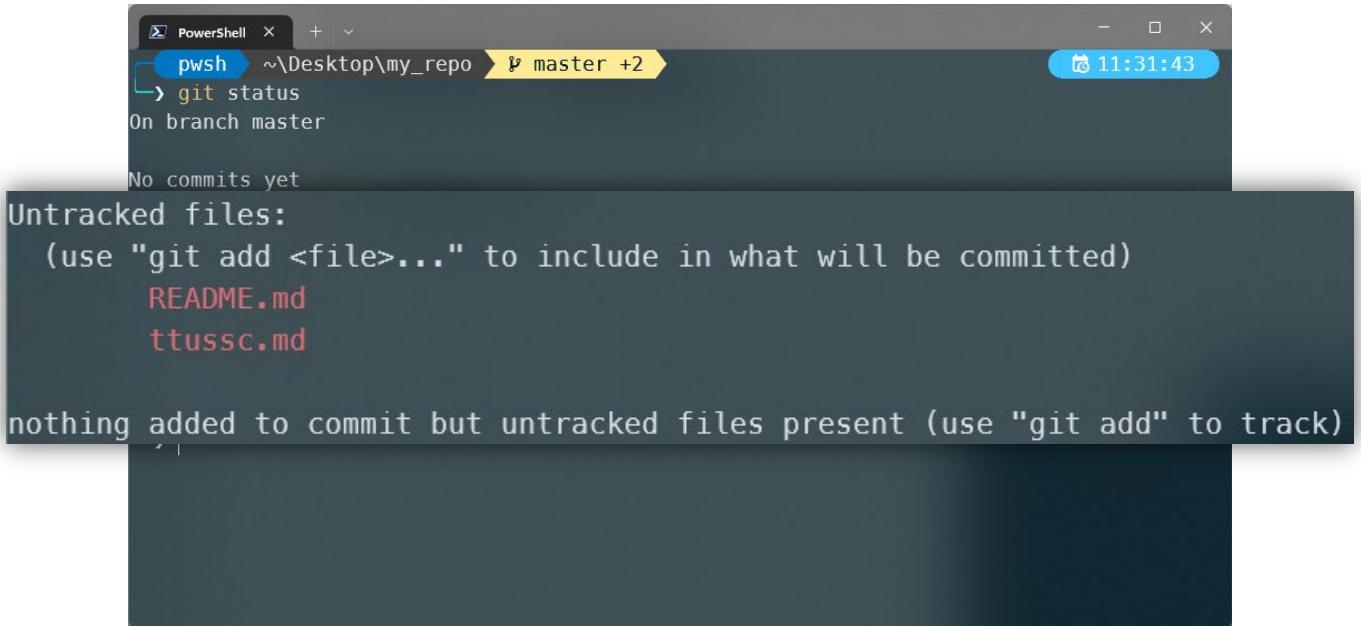
A screenshot of a PowerShell window titled "PowerShell". The session shows the following commands and their outputs:

- pwsh ~\Desktop\my_repo ➔ master
- echo 'hello world' > ttussc.md
- pwsh ~\Desktop\my_repo ➔ master +1
- cat .\ttussc.md
- hello world
- pwsh ~\Desktop\my_repo ➔ master +1
- vim README.md
- pwsh ~\Desktop\my_repo ➔ master +2
- cat .\README.md
- # 羅崧璋
- 2004/02/26
- I1B 31
- pwsh ~\Desktop\my_repo ➔ master +2

The terminal has a dark theme and shows timestamps on the right side of each command line.

Check status

git status # 顯示檔案的狀態



```
PowerShell ~\Desktop\my_repo master +2
git status
On branch master

No commits yet

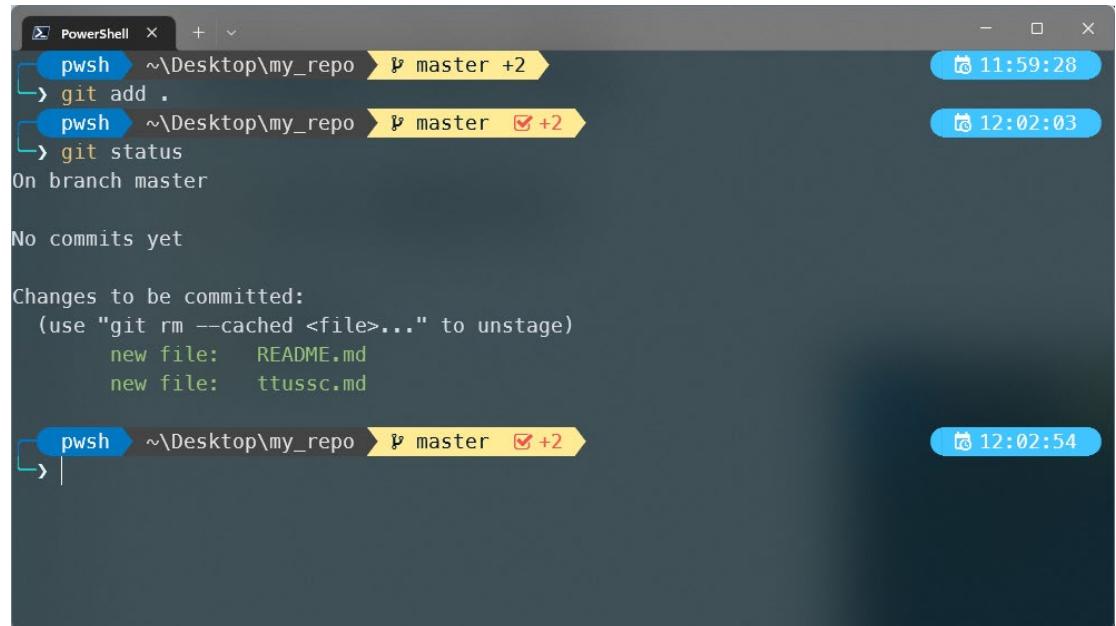
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md
    ttussc.md

nothing added to commit but untracked files present (use "git add" to track)
```

Stage files

git add <檔案名稱>
git add ./ git add *

把檔案新增到預存區(Index)
add 所有檔案



The screenshot shows a PowerShell window with the following session:

```
pwsh ~\Desktop\my_repo p master +2
> git add .
pwsh ~\Desktop\my_repo p master ✘+2
> git status
On branch master

No commits yet

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

pwsh ~\Desktop\my_repo p master ✘+2
>
```

The terminal shows the user navigating to their repository, adding files, and then checking the status. The status output indicates two new files have been added. The terminal has a dark theme and shows timestamps at the top right: 11:59:28, 12:02:03, and 12:02:54.

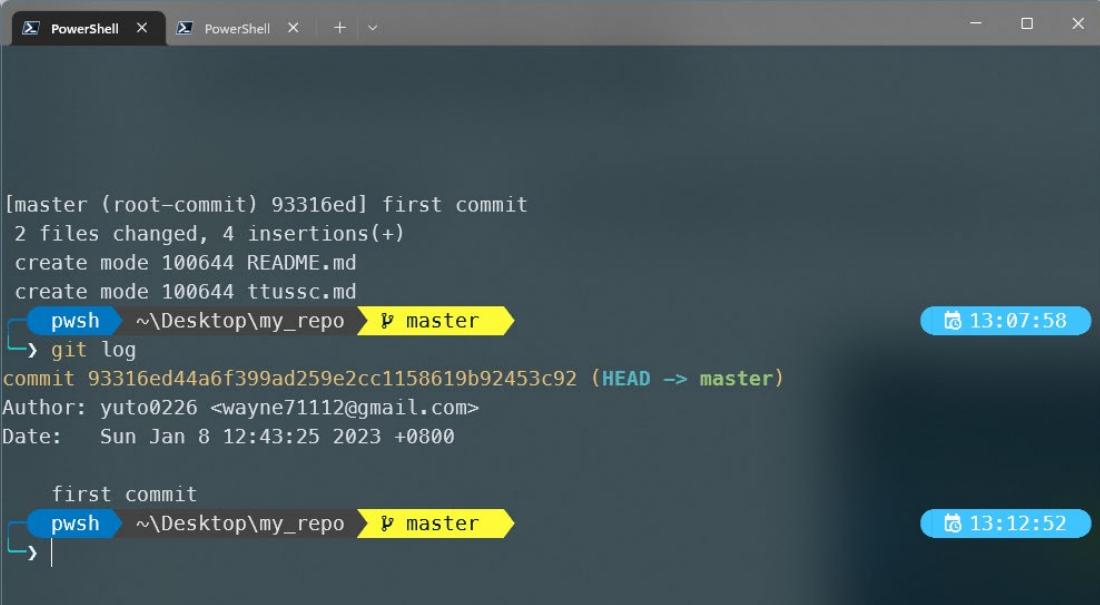
Commit files

`git commit`

提交檔案至本地倉庫(Local Repository)

`git log`

檢視提交的歷史記錄



```
[master (root-commit) 93316ed] first commit
 2 files changed, 4 insertions(+)
  create mode 100644 README.md
  create mode 100644 ttussc.md
pwsh ~\Desktop\my_repo p master
git log
commit 93316ed44a6f399ad259e2cc1158619b92453c92 (HEAD -> master)
Author: yuto0226 <wayne71112@gmail.com>
Date:   Sun Jan 8 12:43:25 2023 +0800

  first commit
pwsh ~\Desktop\my_repo p master
|
```

花俏的 git log

git log

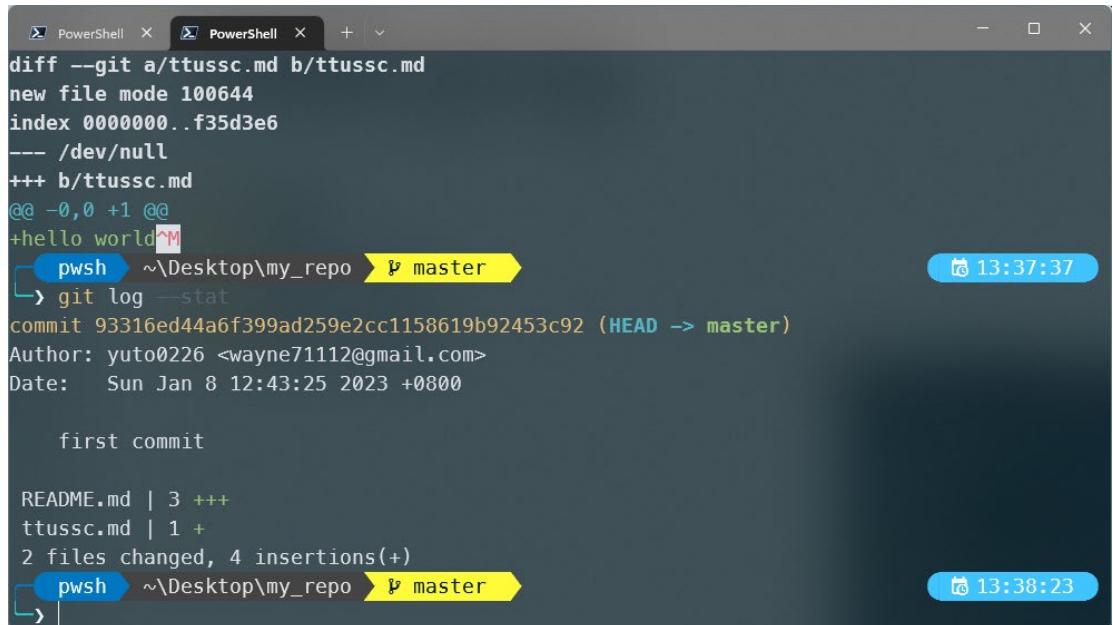
查看提交歷史

git log -p -2

-p 列出變更/-2 最新兩個提交

git log --stat

--stat 統計



```
diff --git a/ttussc.md b/ttussc.md
new file mode 100644
index 000000..f35d3e6
--- /dev/null
+++ b/ttussc.md
@@ -0,0 +1 @@
+hello world

pwsh ~\Desktop\my_repo p master
git log --stat
commit 93316ed44a6f399ad259e2cc1158619b92453c92 (HEAD -> master)
Author: yuto0226 <wayne71112@gmail.com>
Date: Sun Jan 8 12:43:25 2023 +0800

first commit

README.md | 3 +++
ttussc.md | 1 +
2 files changed, 4 insertions(+)

pwsh ~\Desktop\my_repo p master
|
```

花俏的 git log

git log --pretty=format:"%h %s" --graph

```
--- /dev/null
+++ b/ttussc.md
@@ -0,0 +1 @@
+hello world
pwsh ~\Desktop\my_repo p master
git log --stat
commit 93316ed44a6f399ad259e2cc1158619b92453c92 (HEAD -> master)
Author: yuto0226 <wayne71112@gmail.com>
Date:   Sun Jan 8 12:43:25 2023 +0800

first commit

README.md | 3 +++
ttussc.md | 1 +
2 files changed, 4 insertions(+)
pwsh ~\Desktop\my_repo p master
git log --pretty=format:"%h %s" --graph
* 93316ed first commit
pwsh ~\Desktop\my_repo p master
|
```

TASK

- 把 ttussc.md 的內容後面加上 I love you
- commit 到 Local Repository

A screenshot of a PowerShell terminal window titled "PowerShell". The terminal shows the following command history:

```
pwsh ~\Desktop\my_repo > p master
> vim ttussc.md
pwsh ~\Desktop\my_repo > p master ~1
> git add .\ttussc.md
pwsh ~\Desktop\my_repo > p master ✘~1
> git commit -m "add I love you to ttussc.md"
[master ec756e5] add I love you to ttussc.md
1 file changed, 1 insertion(+)
pwsh ~\Desktop\my_repo > p master
>
```

The terminal interface includes a dark background with light-colored text. The status bar at the bottom right shows the current time.

不想被放在 git 裡面上傳的檔案

- 程式編譯的執行檔
- 各式各樣的暫存檔…
- .gitignore

不想被放在 git 裡面上傳的檔案

- screct.yml # 忽略 screct.yml 檔案
- *.exe # 忽略所有 exe 檔案

TASK

- 新增hello.c並編譯(程式碼在hackmd)
- commit 上去並且不能夠有執行檔(用.gitignore)

The screenshot shows a PowerShell window with the following command history:

```
pwsh ~\Desktop\my_repo > master
> vim hello.c
pwsh ~\Desktop\my_repo > master +1
> vim .gitignore
pwsh ~\Desktop\my_repo > master +2
> git add .
pwsh ~\Desktop\my_repo > master ✘ +2
> git commit -m "add hello.c and .gitignore"
[master 299ea0f] add hello.c and .gitignore
2 files changed, 7 insertions(+)
create mode 100644 .gitignore
create mode 100644 hello.c
pwsh ~\Desktop\my_repo > master
>
```

The terminal uses color-coded syntax highlighting for commands and file paths. The commit message is shown in green, and the commit output is in white. The current time is displayed in blue bubbles on the right side of the terminal.



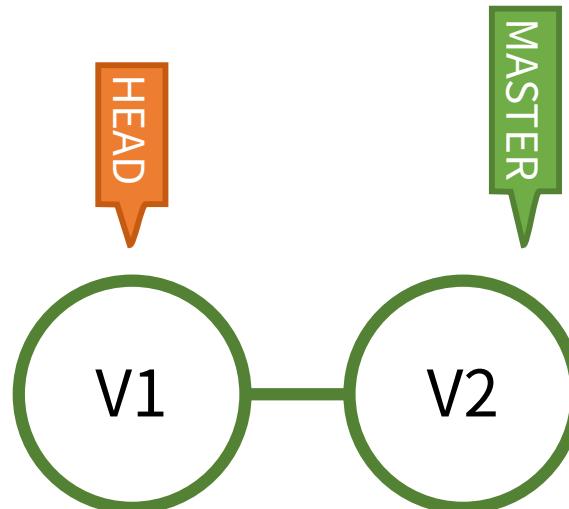
git 的分支概念

❖ git 的分支概念



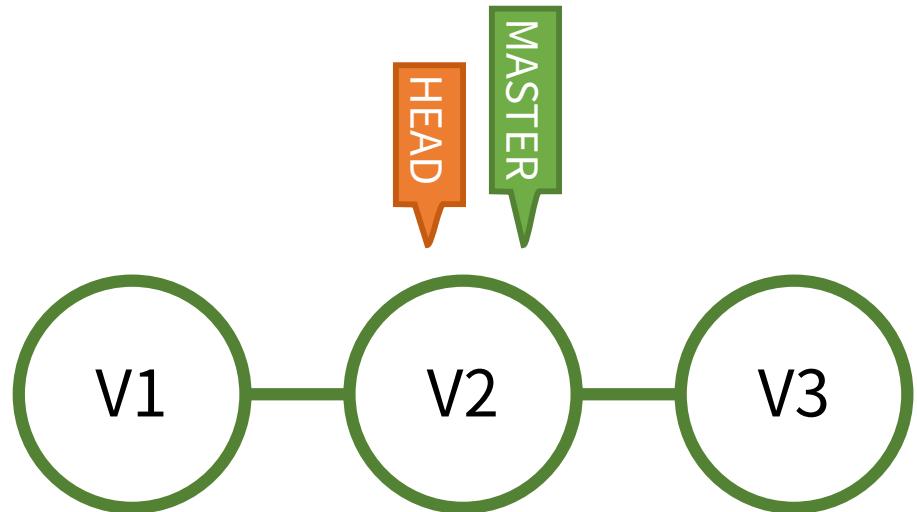
```
git init  
git add  
git commit
```

❖ git 的分支概念



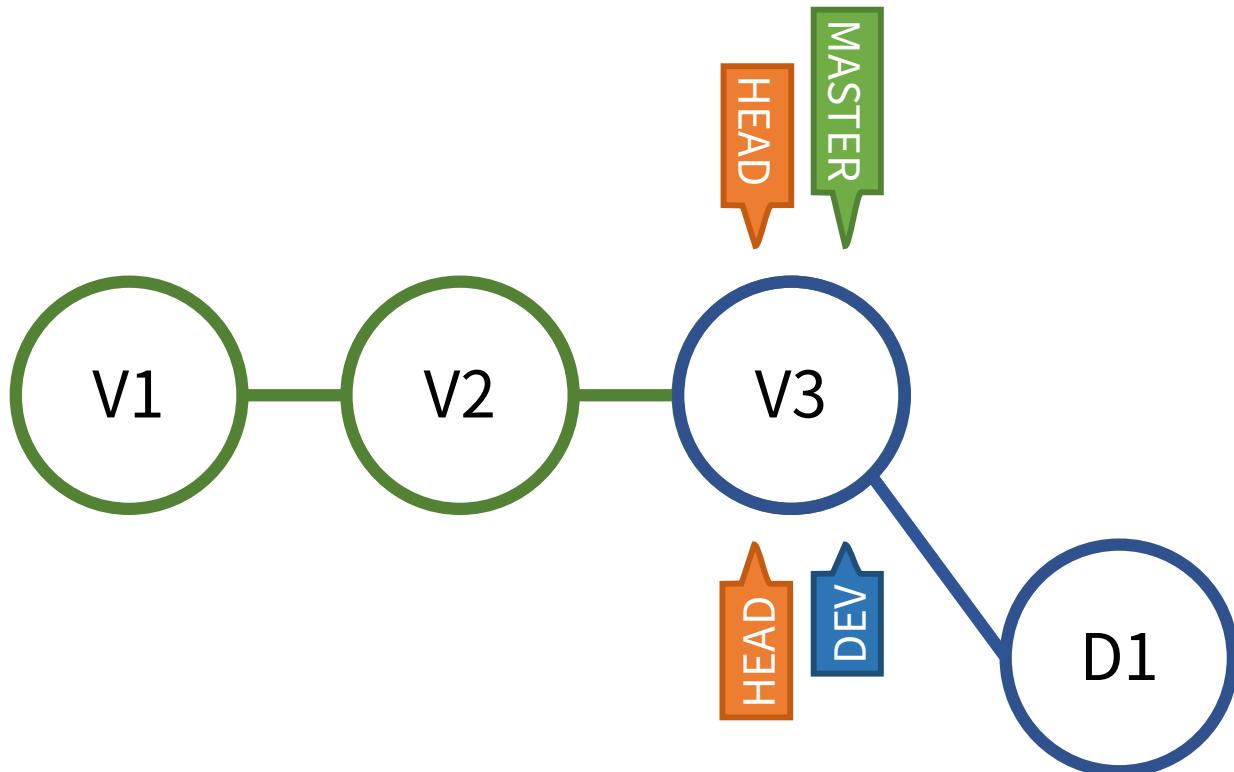
```
git add  
git commit  
git remote add  
git push
```

❖ git 的分支概念



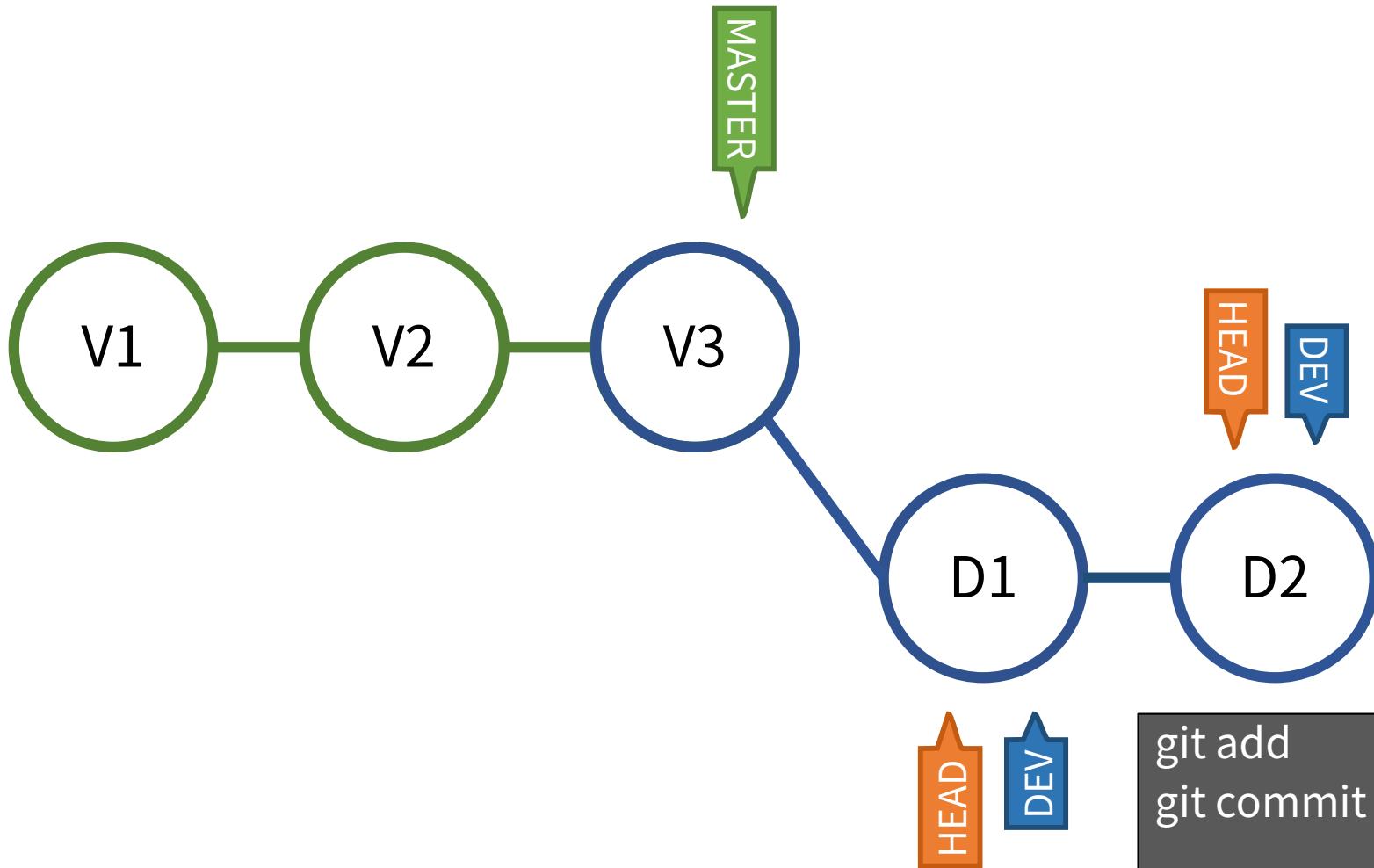
git add
git commit
git push

❖ git 的分支概念



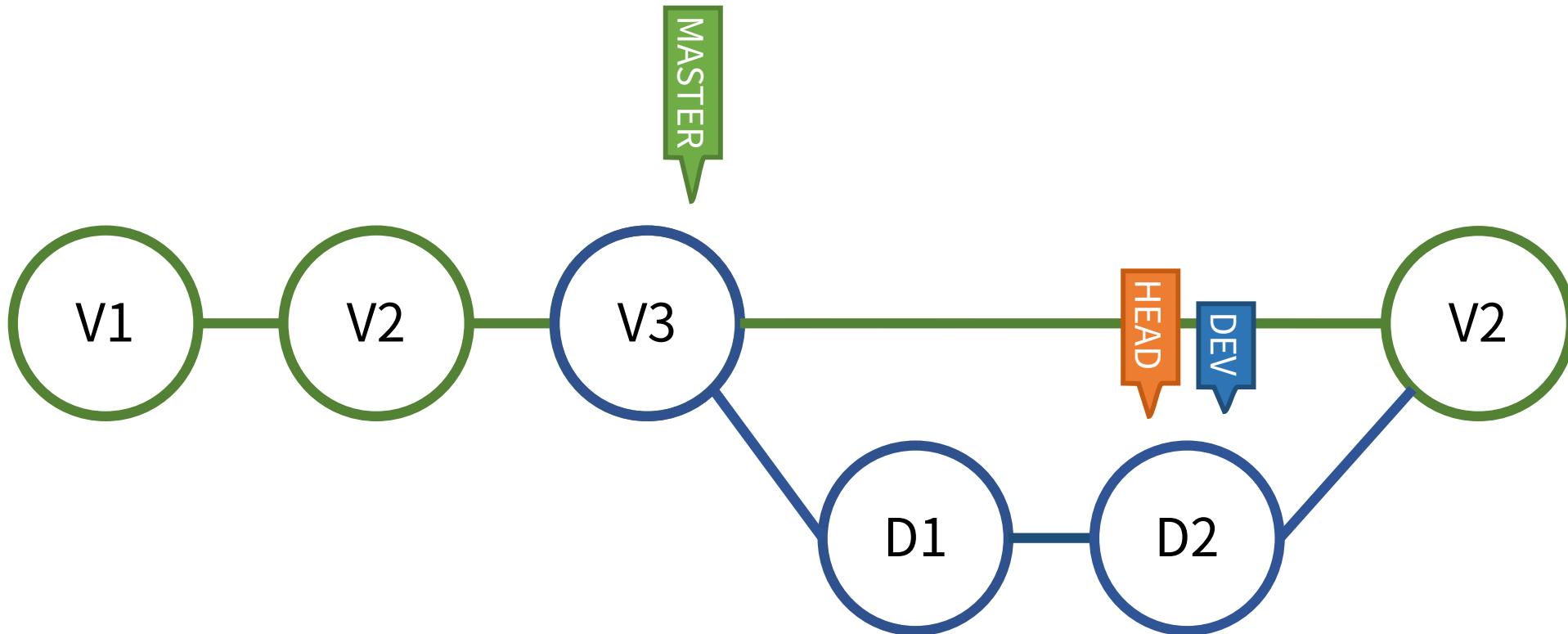
```
git branch dev  
git checkout dev  
git add  
git commit
```

❖ git 的分支概念



```
git add  
git commit
```

❖ git 的分支概念



git merge
git push

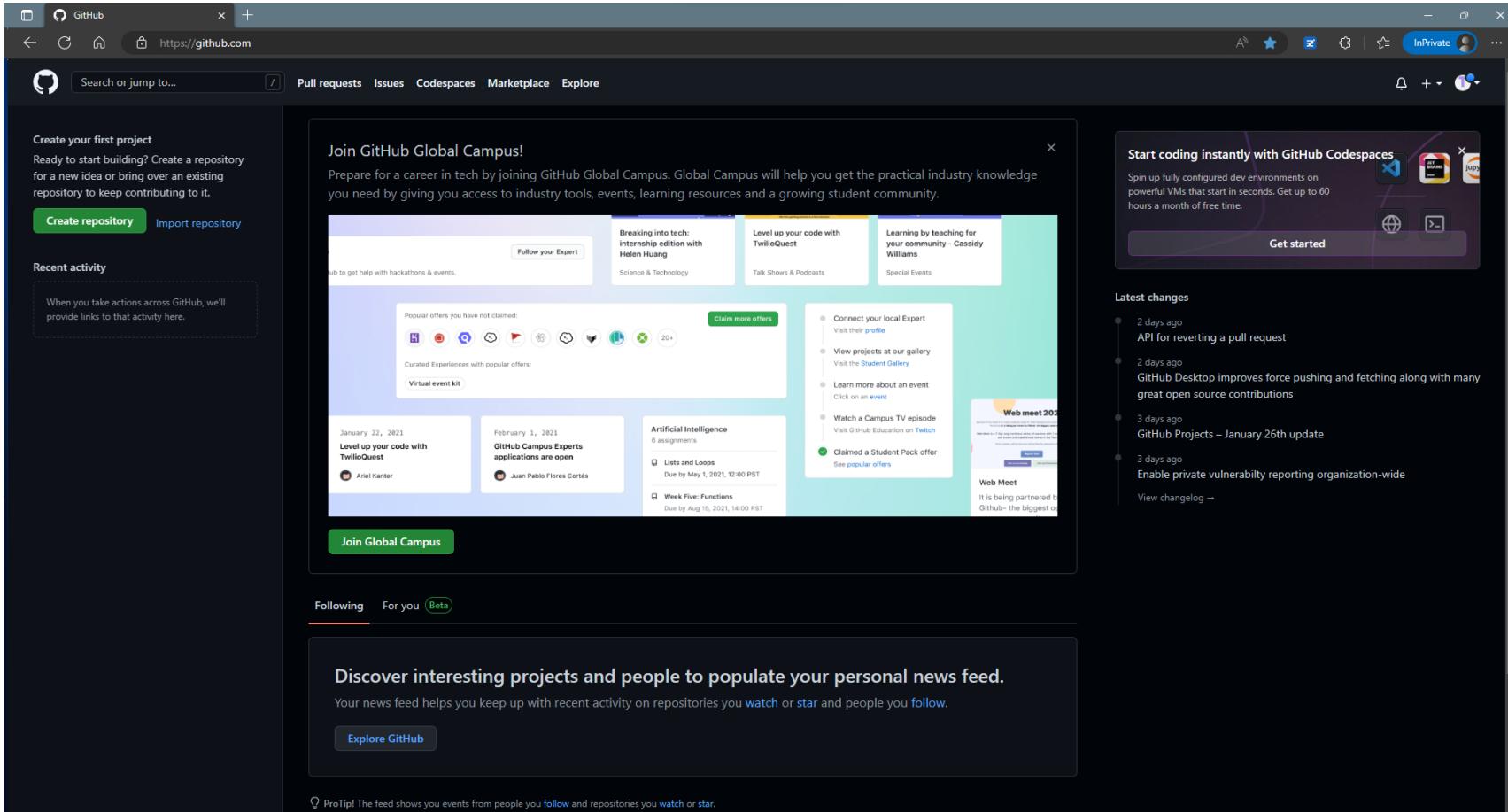


多人分工？雲端硬碟？

什麼是 GitHub

- GitHub是一個線上軟體原始碼代管服務平台，使用 Git 作為版本控制軟體 — Wikipedia
- 講白了就是版本控管的雲端硬碟

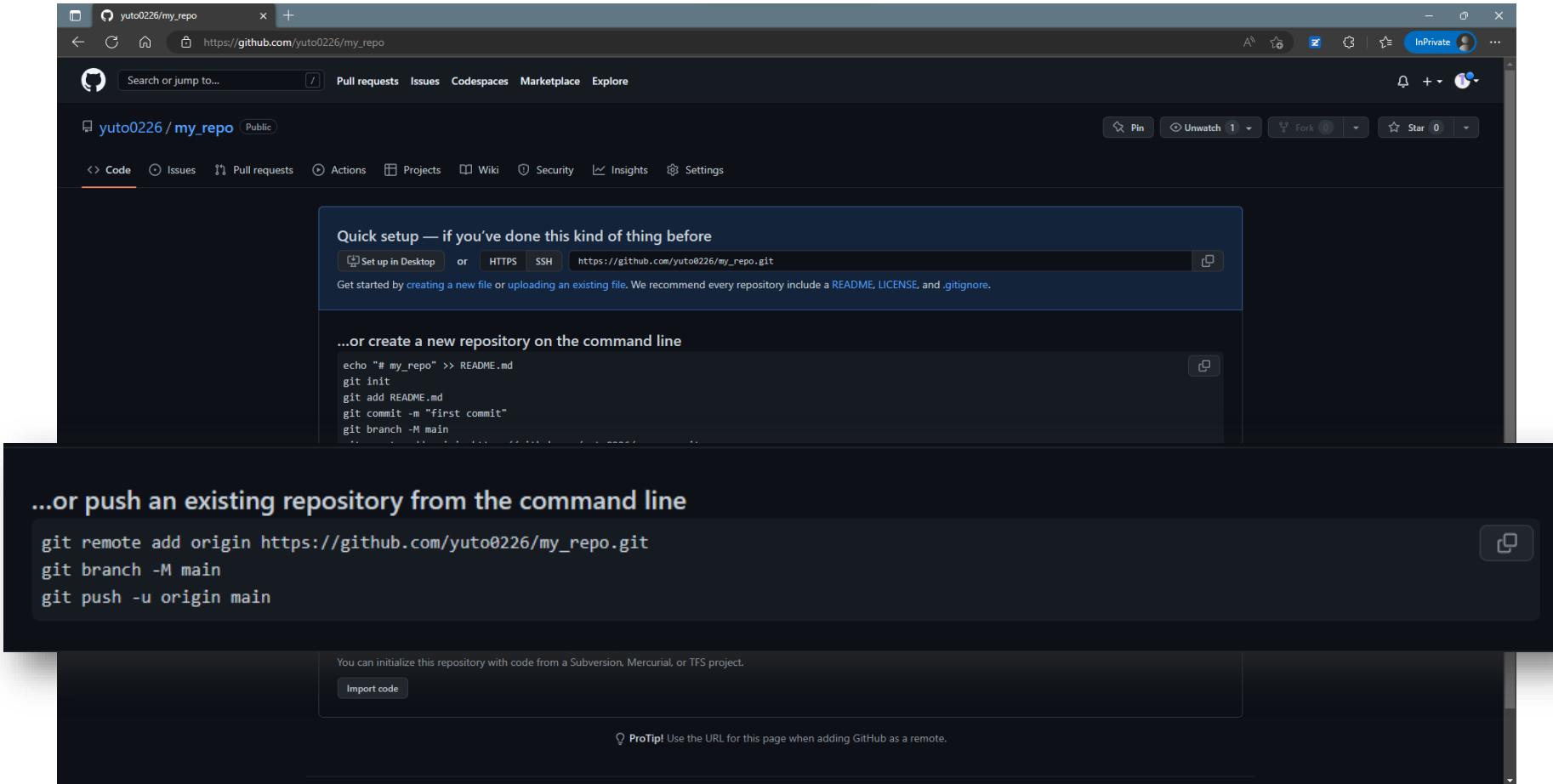
申辦 GitHub 帳號



上傳 Local Repository

- 在 **GitHub** 裡面創建 Respository
- 新增一個 Remote 連到 **GitHub** 上面的 Repository
- 把 Local Repository **push** 上 **Github**

上傳 Local Repository



上傳 Local Repository

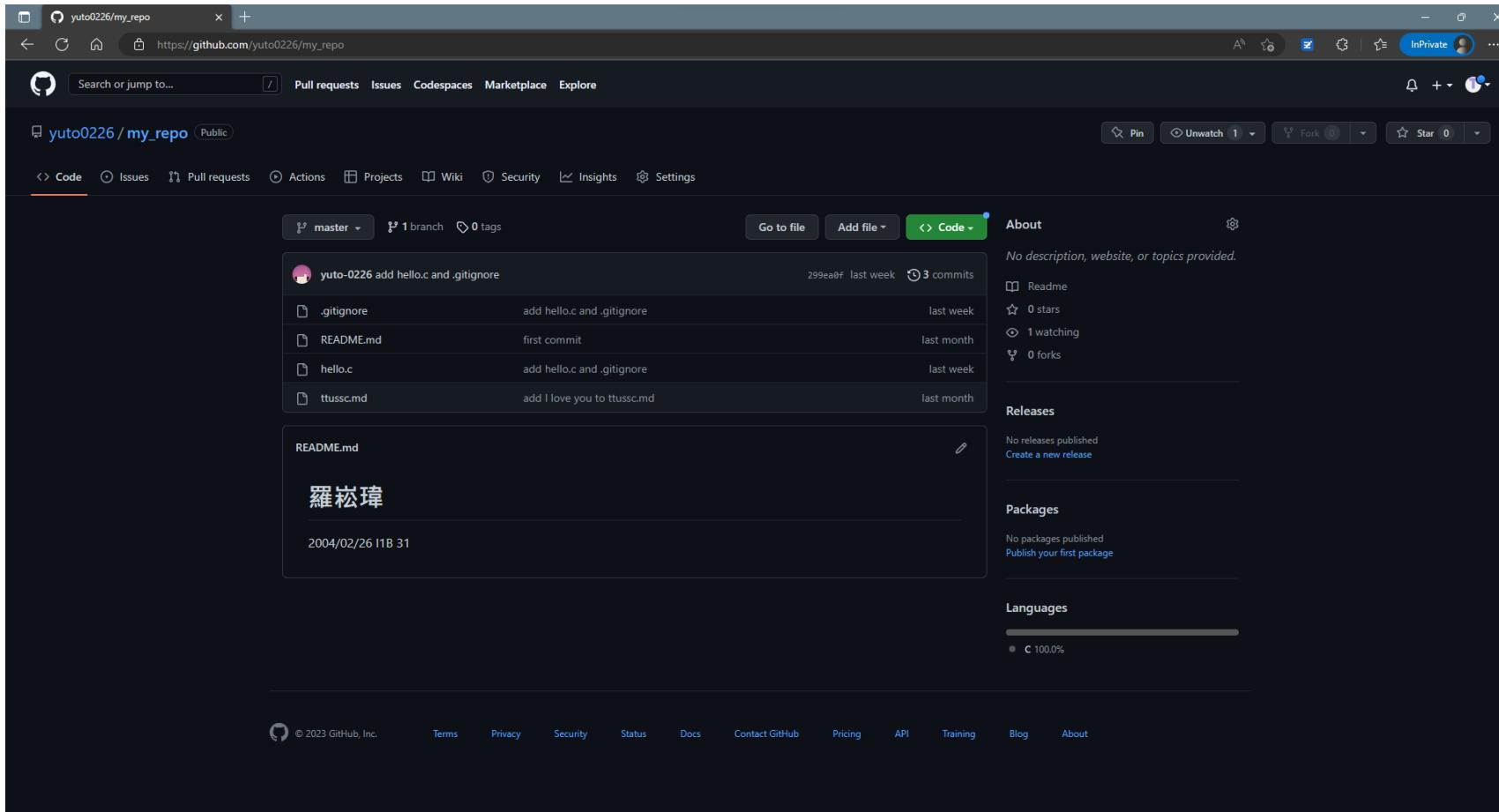
```
git remote add <遠端名稱> <url> # 新增遠端資料庫  
git push <遠端名稱> <分支名稱> # 更新遠端資料庫
```

The screenshot shows a PowerShell window with the following command history and output:

```
PowerShell ~\Desktop\my_repo p master  
git remote add origin https://github.com/yuto0226/my_repo.git  
git push origin master  
info: please complete authentication in your browser...  
Enumerating objects: 11, done.  
Counting objects: 100% (11/11), done.  
Delta compression using up to 20 threads  
Compressing objects: 100% (7/7), done.  
Writing objects: 100% (11/11), 985 bytes | 985.00 KiB/s, done.  
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/yuto0226/my_repo.git  
 * [new branch] master -> master  
psh ~\Desktop\my_repo p master  
|
```

The terminal shows the command `git remote add origin https://github.com/yuto0226/my_repo.git` being run, followed by `git push origin master`. It prompts for browser authentication. The progress of the push operation is shown, including object enumeration, counting, compressing, writing, and total transfer details. Finally, it indicates that a new branch `master` was pushed to the remote repository.

上傳 Local Repository



TASK

- 新增count.c (hackmd直接複製貼上即可)
- 新增一個commit
- push 上 Github

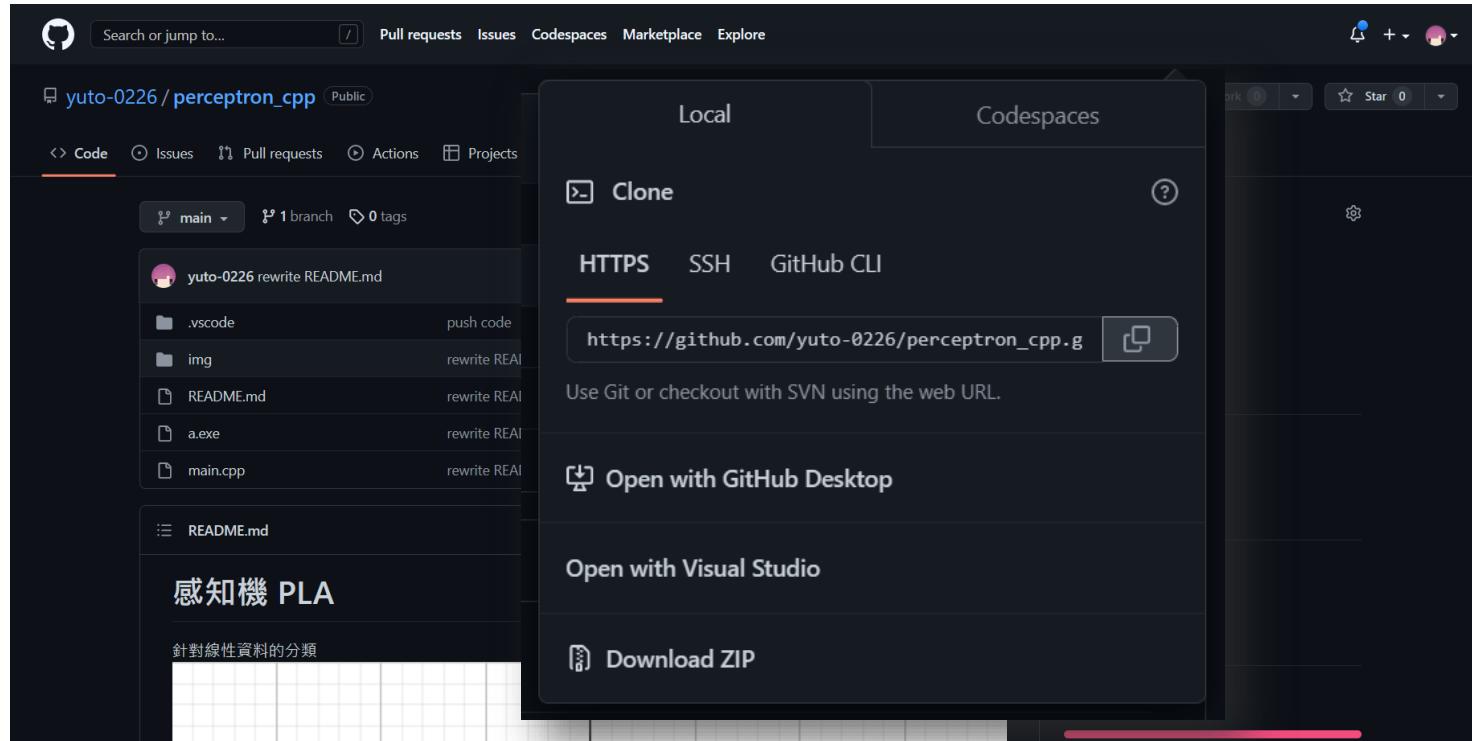
A screenshot of a Windows PowerShell window titled "PowerShell". The terminal shows the following sequence of commands and their outputs:

- pwsh ~\Desktop\my_repo > p master +1
- git add .
- pwsh ~\Desktop\my_repo > p master ✘+1
- git commit -m "add count.c"
- [master a485701] add count.c
1 file changed, 10 insertions(+)
create mode 100644 count.c
- pwsh ~\Desktop\my_repo > p master
- git push origin master
- Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 385 bytes | 385.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/yuto0226/my_repo.git
299ea0f..a485701 master -> master

The terminal window has a dark theme and shows timestamps on the right side of each command line: 14:11:10, 14:12:31, 14:13:24, and 14:14:08.

拷貝別人的Repository

git clone <網址>



加碼大禮包- GitHub pages



- GitHub推出的網頁託管服務
- 可以用於存放靜態網頁
- <ID>.github.io
- 官方文檔

首頁 連結

About Me



羅崧瑋 TTU CSIE

這邊有自介

CSS Styling

網頁設計

網頁設計（英語：Web design）涵蓋了製作和維護網站的許多不同的技能和學科。它包含了網頁圖形設計、**介面設計**、編寫標準化的代碼和專有軟體、**使用者經驗設計**，以及**搜尋引擎最佳化**。通常許多人以團隊協作的方式來工作，各自參與設計過程的各個方面；但有些設計師會參與到任何方面。網頁設計這個術語通常用來描述與網站前端（客戶端）設計相關的設計過程，包括編寫標記式語言。在更廣泛的Web開發領域，網頁設計與**網路工程**（英語：**web engineering**）存在交集。網頁設計師應該對**網路可用性**（英語：**Web usability**）有一定的認識。如果他們的工作涉及到編寫標記式語言，那麼他們也應該遵循網頁親和力的最新指引。

Reference

- [Git](#)
- [w3hexschool](#)
- [為你自己學 Git](#)