



# 版本控制系統(Version Control System)

## Part 2 Git基本介紹

Wei-Tsung Su  
suwt@au.edu.tw  
564D





# 歷史版本

版本	說明	日期	負責人
v1.0	中文化初版	2019/02/26	蘇維宗





## 參考文件

1. [http://en.wikipedia.org/wiki/Revision\\_control](http://en.wikipedia.org/wiki/Revision_control)
2. <http://homes.cs.washington.edu/~mernst/advice/version-control.html>
3. <http://git-scm.com/book>
4. <https://backlog.com/git-tutorial/tw/>





# 內容

Git的歷史

Git的版本管理方式

安裝Git

檔案狀態

Git的操作(取得Repository、檢查狀態、新增/更新狀態、新增版本、上傳版本)

檢視版本變化

展示操作



# Git的歷史

Linux核心專案早期維護在BitKeeper分散式版本管理系統中。

Linus Torvalds於2005年嘗試開發自己的版本管理系統工具，也就是現在的Git。Linux核心程式碼(與許多開放原始碼專案)目前維護在知名的GitHub平台中。

<https://github.com/torvalds/linux>

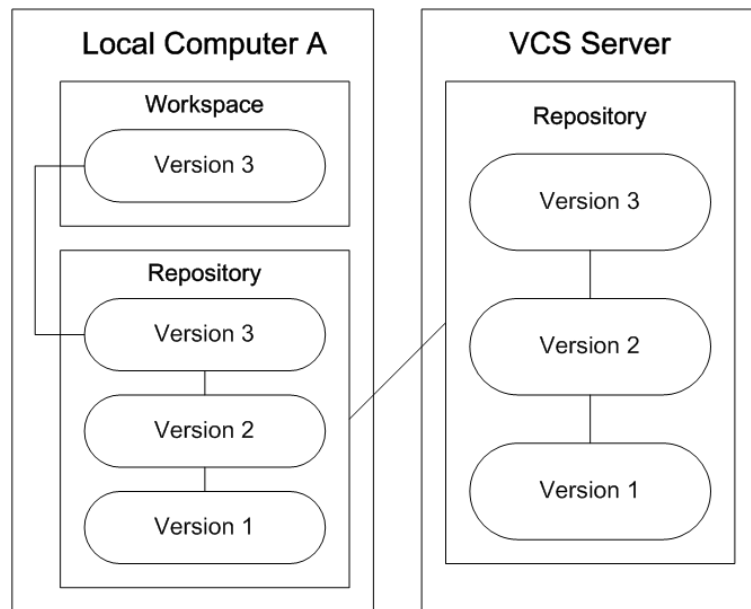


# 為何Git適合用來管理大型專案?

Git會在本機保存專案的所有版本變化，所以可以直接在本機上完成所有專案管理的動作，有需要再上傳到伺服器上。

**優點**是可以在本機進行版本管理

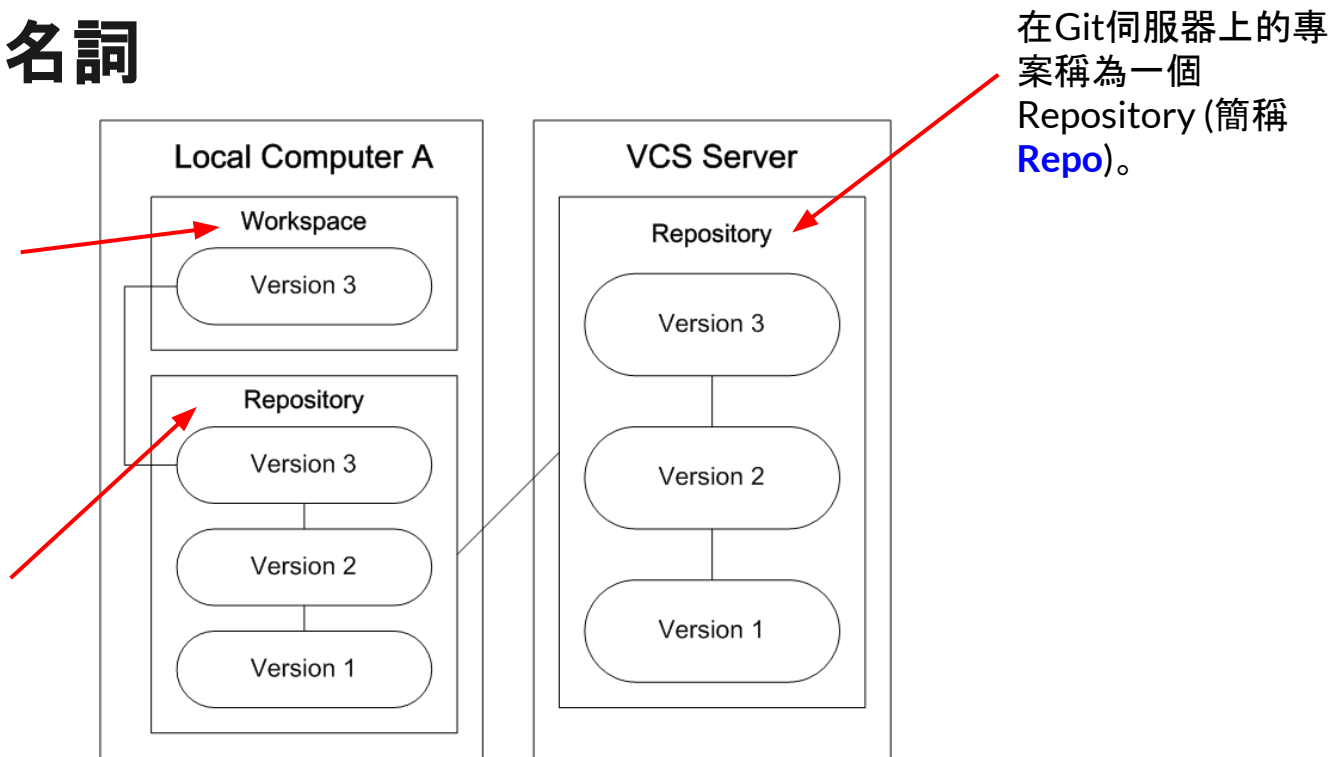
**缺點**是會佔用較多的本機空間



# Git的專有名詞

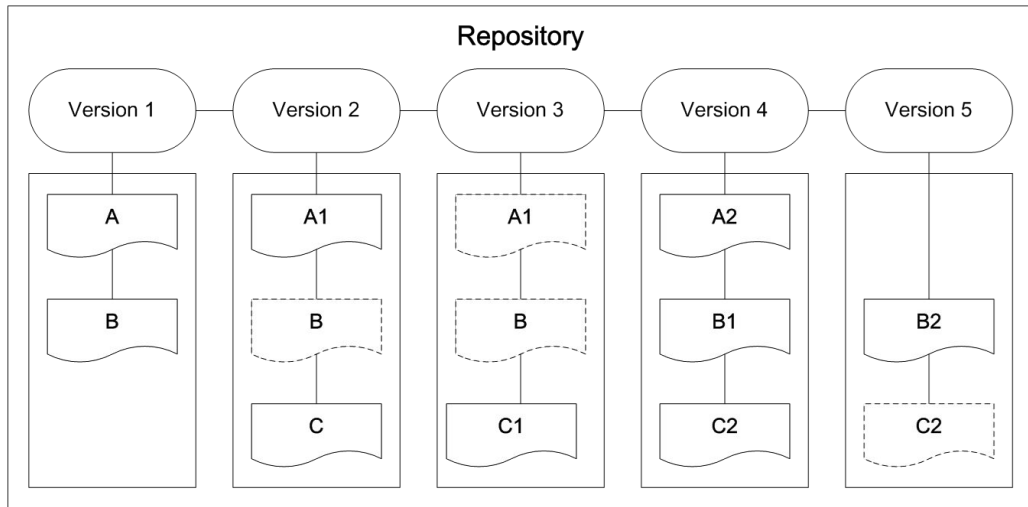
**Workspace**為在本機端目前所使用的版本

可以將Git伺服器上的  
**Repo**完整的下載  
到本機端。



# Git的版本管理方式

Git儲存的是版本快照(snapshots), 而不像SVN儲存是版本間的差異。







# 安裝Git

使用Windows或Mac電腦, 可至下面的網站下載安裝套件

<https://git-scm.com>

使用Linux (CentOS)

```
$ yum install git
```

使用Linux (Ubuntu)

```
$ apt-get install git
```



# 環境設定

Git的環境設定有分三個層級

- Local: 針對專案(**必須在專案目錄中**。設定檔`[repo]/.git/config`)
- Global: 針對使用者(設定檔`~/.gitconfig`)
- System: 針對系統(**需要有系統權限**。設定檔`/etc/gitconfig`)

當多個層級皆有環境設定時, 優先順序為

Local > Global > System





## 環境設定(續)

設定貢獻者資訊(當更新版本時, 這些資訊會被記錄)

```
$ git config --[system/global/local] user.name "[name]"  
$ git config --[system/global/local] user.email "[email]"
```

設定預設的文件編輯器

```
$ git config --[system/global/local] core.editor [editor]
```

檢查設定

```
$ git config --[system/global/local] --list
```





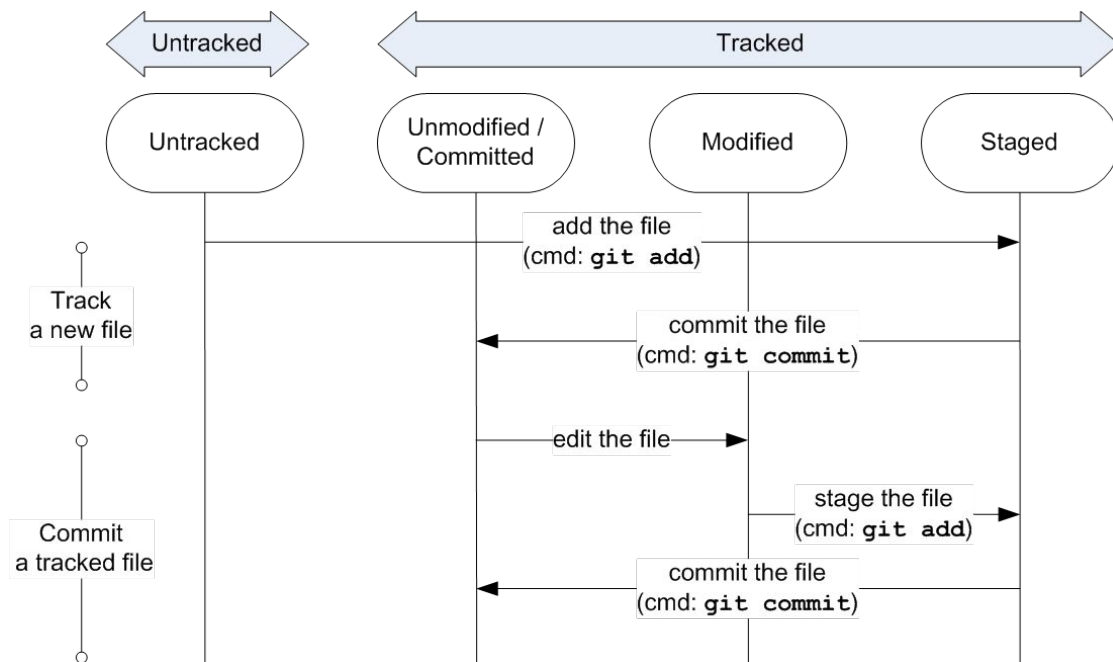
# 檔案狀態

在Git管理下的本機檔案會有幾種不同的狀態

- Untracked: 尚未納入管理的新檔案
- Unmodified: 沒有變動過
- Committed: 變動過且已納入新版本的檔案
- Modified: 變動過但尚未納入新版本的檔案
- Staged: 需要納入新版本的檔案(可能是新檔案也可能是變動過的檔案)



# 檔案狀態的改變





# 基本使用方式

下載Git Repo指令

```
$ git clone [Git Repo URI]
```

**備註:**在下載的專案目錄(以repo代表)中會產生一個.git的隱藏目錄, 裡面包含了該專案所有的版本資訊。





## 基本使用方式(續)

查看專案內檔案的狀態

```
[repo]$ git status
```

檔案狀態

- Nothing to commit, working tree clean: 所有檔案都處於 unmodified 或 committed 的狀態。
- Untracked files: 處於 untracked 狀態的檔案
- Changes to be committed: 處於 staged 狀態的檔案
- Changed not staged for commit: 處於 modified 狀態的檔案





## Repo檔案處理(add/rm/mv)

當新增檔案(或修改已存在的檔案)

```
[repo]$ git add [file]
```

刪除已存在的檔案

```
[repo]$ git rm [file]
```

變更已存在檔案的檔名(或檔徑)

```
[repo]$ git mv [src] [dst]
```

**備註:**不管是新增、修改、刪除、或是變更檔名或路徑,最後檔案的狀態都會變成statged。







## 在本機端新增版本(commit)

將目前所有狀態為staged的檔案變成新的版本

```
[repo]$ git commit
```

```
[repo]$ git commit -m '[變更內容的描述]'
```

查詢本機端的版本更新紀錄

```
[repo]$ git log
```

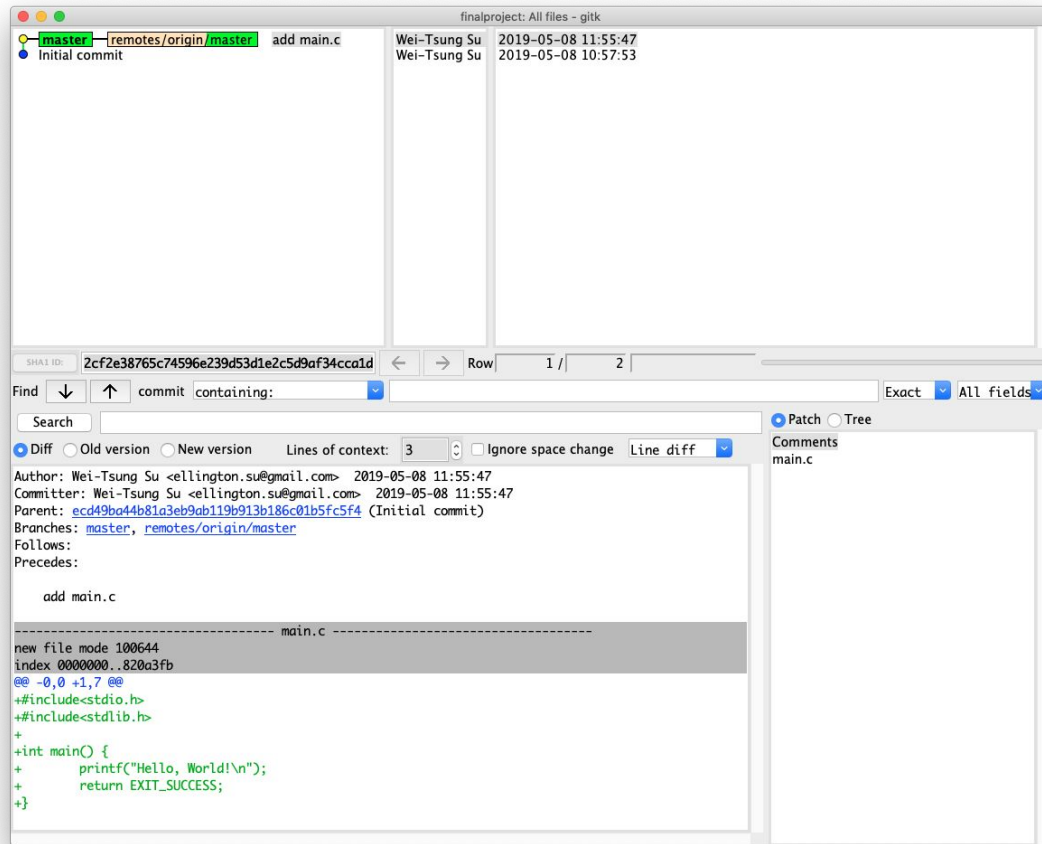
**備註:**這些版本目前都只存在本機端的Git Repo



# 視覺化工具

## 版本紀錄

[repo]\$ gitk





## 將本地端的版本紀錄更新到遠端伺服器(push)

為了與團隊成員分享你對Git Repo進行的修改，你必須透過push指令將本機端上的版本同步到遠端的Git伺服器上。

```
$ git push [upstream] [branch]
```

通常，upstream與branch預設為origin與master。如果，沒有指定branch，則未將目前的branch同步到伺服器上。





## 更新遠端版本(Push)

為了與團隊成員分享你對Git Repo進行的修改，你必須透過push指令將本機端上的版本同步到遠端的Git伺服器上。

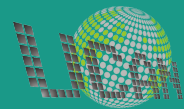
```
[repo]$ git push [upstream] [branch]
```

通常，upstream與branch預設為origin與master。如果，沒有指定branch，則未將目前的branch同步到伺服器上。



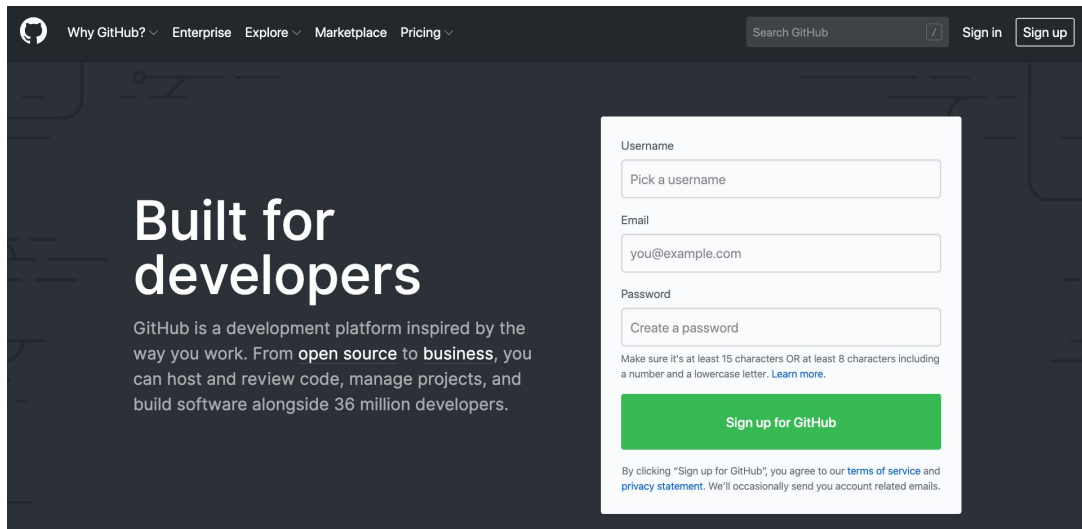
---

# 實際操作



# 申請GitHub帳號

到GitHub官方網站(<https://github.com/>)申請帳號



The screenshot shows the GitHub sign-up page. The header includes the GitHub logo, navigation links (Why GitHub?, Enterprise, Explore, Marketplace, Pricing), a search bar, and 'Sign in' and 'Sign up' buttons. The main content area features the text 'Built for developers' and a description of GitHub as a development platform. On the right, there is a sign-up form with fields for Username, Email, and Password, followed by a green 'Sign up for GitHub' button. Below the button, there is a disclaimer about agreeing to terms of service and privacy statement.

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub / Sign in 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 36 million developers.

Username

Pick a username

Email

you@example.com

Password

Create a password

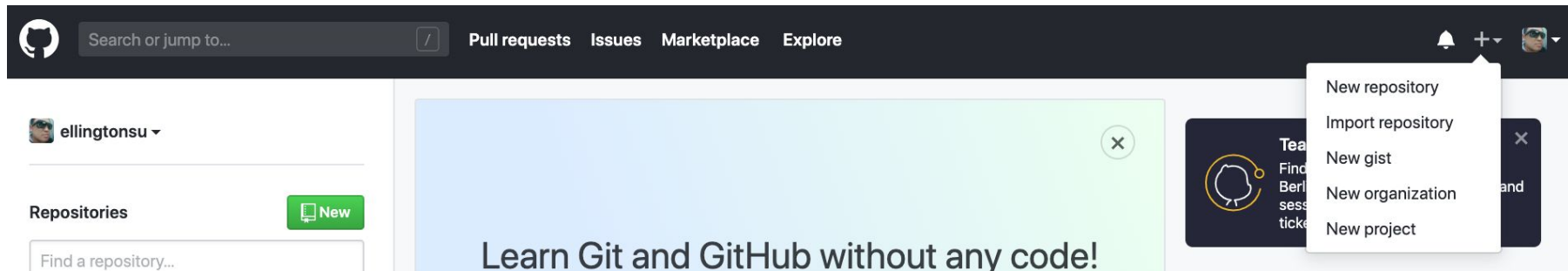
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

**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.

# 新增Git Repo

選擇New repository來新增一個Git Repo



# 新增Git Repo (續)

Owner (專案擁有人)

Repository name (專案名稱)

Description (描述)

Public / Private (公開 / 私人專案)

另外, 可選擇是否新增

專案說明(README)

排除規則(.gitignore)

專案授權(license)

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner

 coachsu ▼

Repository name \*

finalproject ✓

Great repository names are short and memorable. Need inspiration? How about **legendary-carnival**?

Description (optional)

Final Project for Programming Class



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: C ▼

Add a license: None ▼



Create repository





# Git Repo主頁

多少人喜歡  
這個專案?

多少人想貢  
獻這個專案?

coachsu / finalproject

Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Final Project for Programming Class Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

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

ellingtonsu Initial commit Latest commit ecd49ba a minute ago

.gitignore Initial commit a minute ago

README.md Initial commit a minute ago

README.md

finalproject

Final Project for Programming Class

最後更新資訊

專案說明文件

取得專案URI





# Git Repo開發

## 下載Git Repo

```
$ git clone [Git Repo URI]
```

## 新增檔案(例如, main.c)並追蹤

```
[repo]$ git add *
```

## 建立新版本

```
[repo]$ git commit -m 'add main.c'
```

## 查看Git Repo內檔案的狀態

```
[repo]$ git status
```

## 查詢目前的版本更新紀錄

```
[repo]$ git log
```

## 更新遠端版本

```
[repo]$ git push
```



# Git Repo主頁

coachsu / finalproject

Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Final Project for Programming Class Edit

[Manage topics](#)

2 commits 1 branch 0 releases 1 contributor

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

ellingtonsu add main.c Latest commit 2cf2e38 24 seconds ago

.gitignore	Initial commit	an hour ago
README.md	Initial commit	an hour ago
main.c	add main.c	24 seconds ago

README.md

## finalproject

Final Project for Programming Class

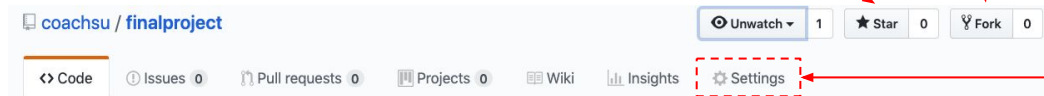
剛剛更新的檔案



# 新增Git Repo協作者

多少人喜歡  
這個專案?

多少人想貢  
獻這個專案?



設定專案

Final Project for Programming Class

Edit

Manage topics

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

取得專案URI

最後更新資訊

ellingtonsu Initial commit

Latest commit ecd49ba a minute ago

.gitignore

Initial commit

a minute ago

README.md

Initial commit

a minute ago

專案說明文件

README.md

finalproject

Final Project for Programming Class



# 新增Git Repo 協作者

coachsu / finalproject

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Options

**Collaborators & teams**

Branches

Webhooks

Notifications

Integrations & services

Deploy keys

Vulnerability alerts

Moderation

Interaction limits

**Default repository permission**

The coachsu organization has their default repository permission set to read. This means that every member of this organization has read access to this repository, regardless of the team and collaborator access specified below.

You can change or remove the default repository permission setting on this organization's [member privileges page](#).

**Teams** [+ Create new team](#)

No teams have been given access to this repository yet.

**Collaborators**

Awaiting Leavatein's response [Write](#) [Copy invite link](#) [Cancel invite](#)

**Search by username, full name or email address**

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

[Add collaborator](#)

協作者權限

新增專案協作者



# Git Repo協作練習

請兩個人一組 (假設為Alice與Bob)

1. Alice邀請Bob到自己的Git Repo
2. Bob下載Alice的Git Repo到本機端
3. Bob在Alice的Git Repo中新增檔案並更新到GitHub上
4. 確認Git Repo在GitHub上的更新狀態

# Q & A

---



Computer History Museum, Mt. View, CA

