



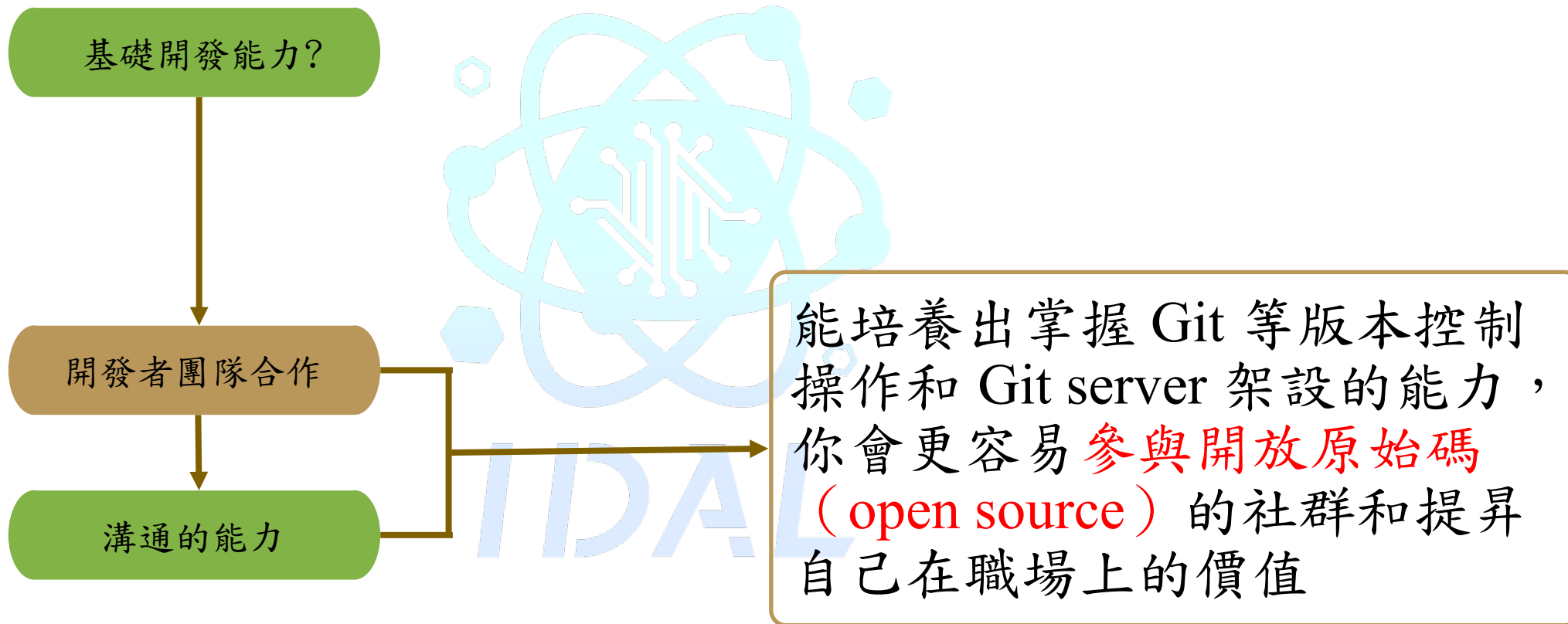
Git 與 Github

IDAL

為甚麼需要學習 Git 與 GitHub?



IDAL



什麼是版本控制系統（Version Control System）？



IDAL

- 一種軟體工程的開發技巧 → 方便同步和維護管理
- 一般在軟體開發中又分為集中式系統（例如：Subversion、CVS 等）與分散式系統（例如：Git、BitKeeper、mercurial 等）

01

集中式系統

- 用戶端必須與伺服器同步程式碼，之後才能建立新版本的程式碼
- 一次只能讓一個開發者進行工作
- 較容易管理

02

分散式系統

- 不同開發者直接在各自的本地檔案庫工作
- 容許多個開發者同時更動同一個檔案
- 每個檔案庫有另外一個合併各個改變的功能
- 沒有網路的情況下也能繼續工作

什麼是 Git？什麼是 Github？



- 分散式版本控制軟體
- 最初目的是為更好地管理 Linux kernel 開發而設計
- 其具備優秀的 merge tracing 合併程式碼的能力

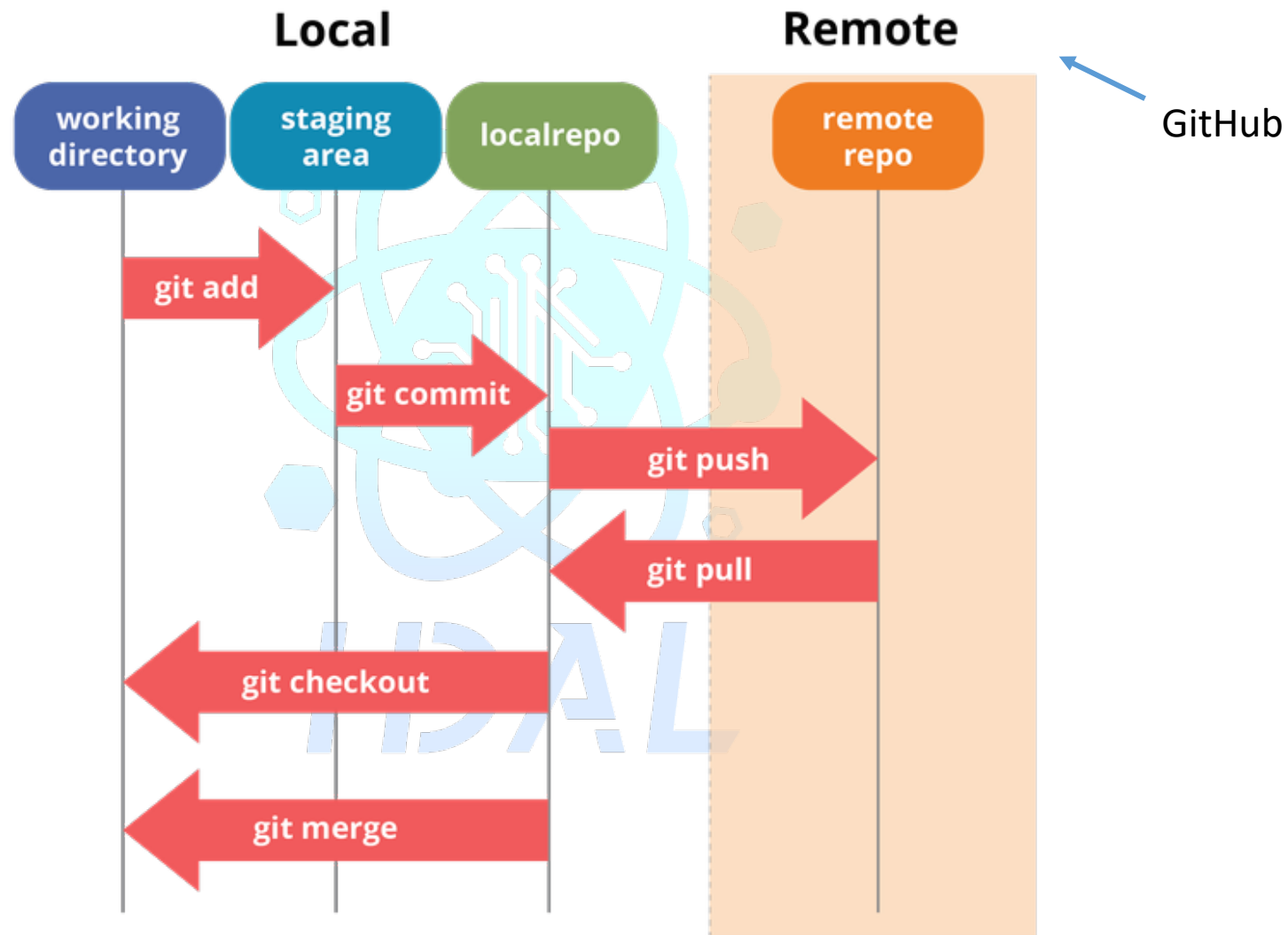


- 支援 git 程式碼存取以及管理專案的平台
- 建議可以熟悉掌握 Git 和 Github 的使用，並建立自己的 Github profile 作品集

Git 流程



IDAL



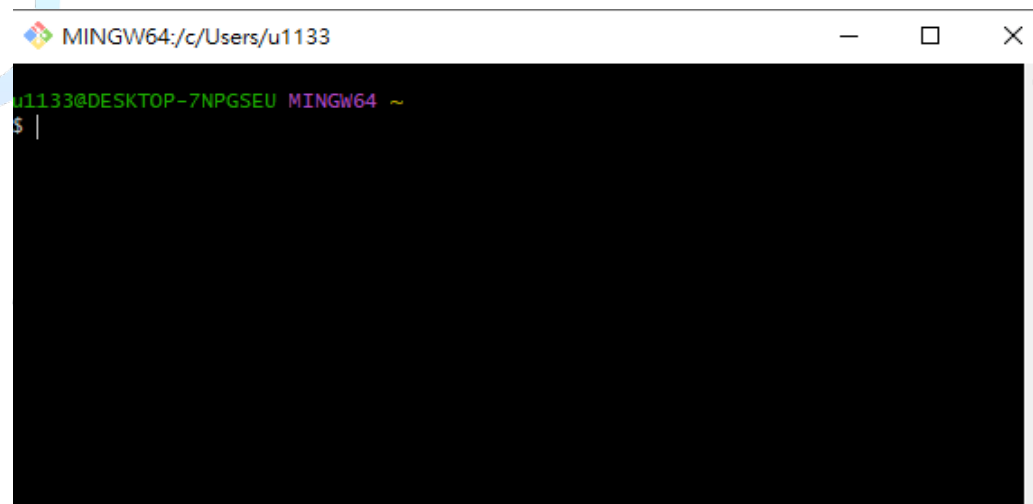
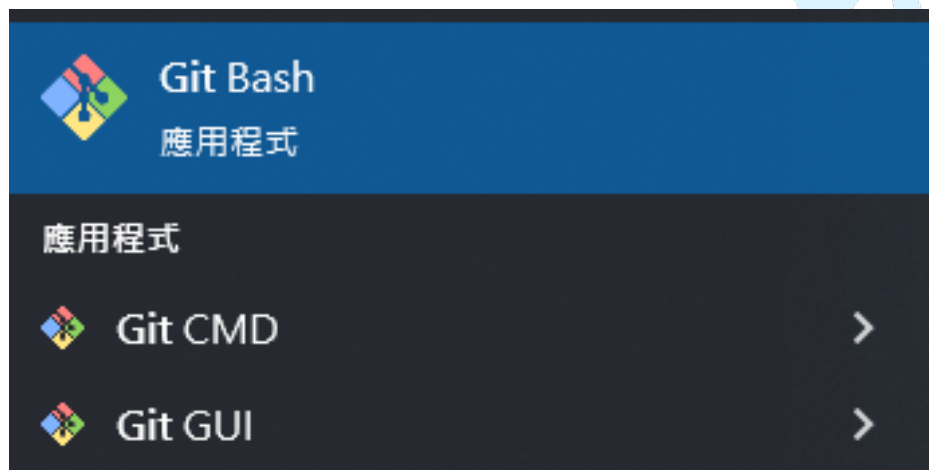
Git 與 Github 實戰操作入門教學



IDAL

1. 安裝並且設定 Git

- Windows 安裝完成後可以透過任何 terminal 程式執行 git 指令
- Linux 內建 terminal 即可執行 git 指令
- Mac 則是可以到 [Git 官方網站](#) 選擇對應作業系統，按照步驟完整下載安裝([中文安裝教學](#))

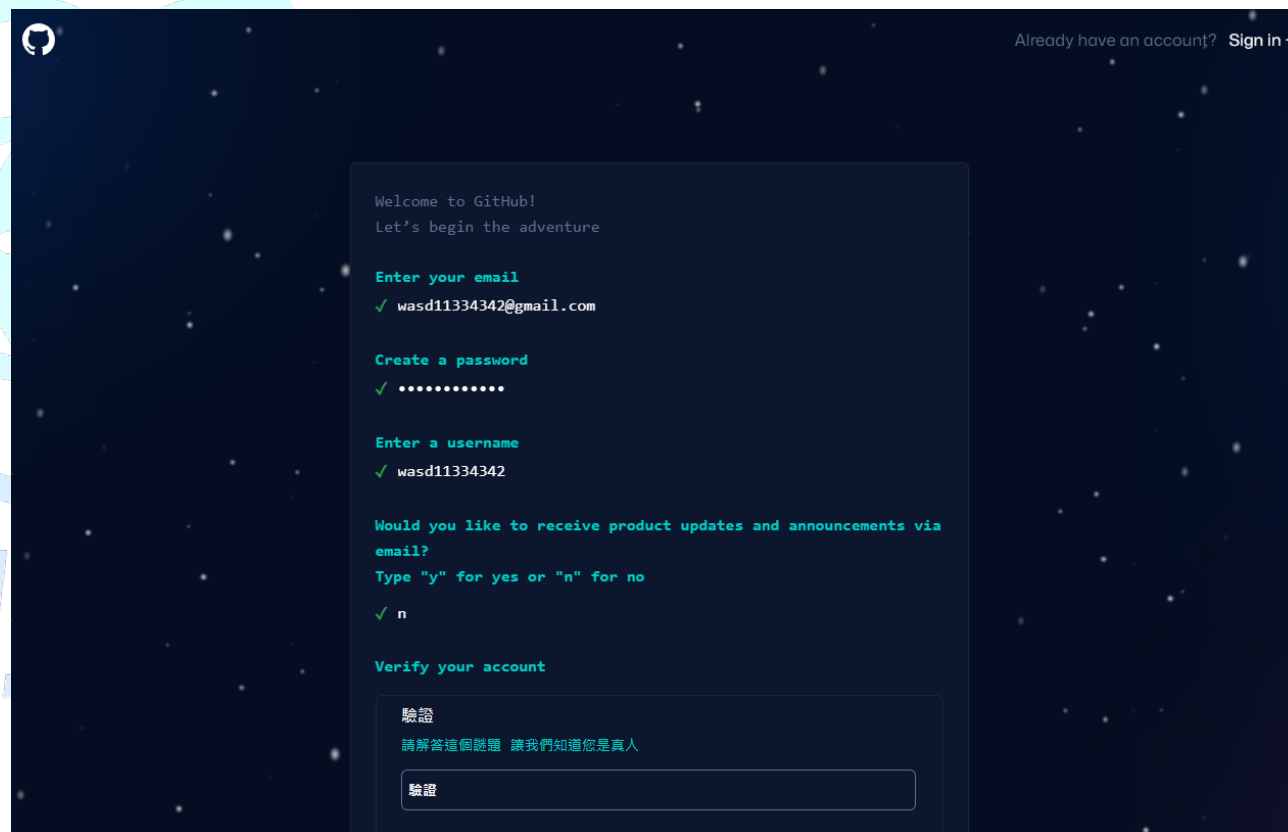


註冊 GitHub 帳號



IDAL

- 先到 GitHub 官網註冊帳號：



The screenshot shows the GitHub registration interface. At the top right, it says "Already have an account? Sign in →". The main content area has a dark blue background with a starry pattern. It starts with "Welcome to GitHub! Let's begin the adventure". The registration steps are as follows:

- Enter your email**: A green checkmark is shown next to the email address "wasd11334342@gmail.com".
- Create a password**: A green checkmark is shown next to a masked password ".....".
- Enter a username**: A green checkmark is shown next to the username "wasd11334342".
- Would you like to receive product updates and announcements via email?**: The instruction says "Type 'y' for yes or 'n' for no". A green checkmark is shown next to the answer "n".
- Verify your account**: This section contains a "驗證" (Verify) button and a text prompt "請解答這個謎題 讓我們知道您是真人" (Solve this puzzle so we know you're real). Below this is a text input field with the placeholder text "驗證".

Github操作



IDAL

- 註冊成功畫面
- 點選右上角來新repository（檔案庫）：

The screenshot shows the GitHub homepage. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Codespaces, Marketplace, and Explore. On the left sidebar, under 'Create your first project', the 'Create repository' button is highlighted with a red box. The main content area features the heading 'The home for all developers — including you.' and a welcome message. Below this are several cards: 'Start writing code' with a 'Start a new repository' section, 'Introduce yourself with a profile README', 'Use tools of the trade' with 'Simplify your development workflow with a GUI' and 'Install a powerful code editor'.

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository Import repository

Recent activity

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

The home for all developers — including you.

Welcome to your personal dashboard, where you can find an introduction to how GitHub works, tools to help you build software, and help merging your first lines of code.

<> Start writing code

Start a new repository

A repository contains all of your project's files, revision history, and collaborator discussion.

wasd11334342 /

☐ Public
Anyone on the internet can see this repository

☒ Private
You choose who can see and commit to this repository

Create a new repository

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

wasd11334342 / README.md **Create**

```
1 - 🙋 Hi, I'm @wasd11334342
2 - 👀 I'm interested in ...
3 - 🌱 I'm currently learning ...
4 - 💬 I'm looking to collaborate on ...
5 - 📫 How to reach me ...
6
```

Latest changes

- 2 days ago
Secret scanning alerts link to providers' docs
- 2 days ago
GitHub Actions: SBOMs now attached to hosted runner image releases for Ubuntu & Windows
- 3 days ago
Secret scanning now push protects custom patterns
- 3 days ago
Secret scanning is now available for free on public repositories

[View changelog →](#)

Use tools of the trade

Simplify your development workflow with a GUI

Install [GitHub Desktop](#) to visualize, commit, and push changes without ever touching the command line.

Install a powerful code editor

[Visual Studio Code](#) is a multi-platform code editor optimized for building and debugging modern applications.



• 資料庫命名

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

wasd11334342

Repository name *

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

Description (optional)

☒ 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.](#)

.gitignore template: None

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None

You are creating a public repository in your personal account.

Create repository

• 成功畫面

wasd11334342 / NIU_software_engineering Public

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main

1 branch

0 tags

Go to file

Add file

<> Code

wasd11334342 Initial commit

8ee0811 now

1 commit

README.md

Initial commit

now

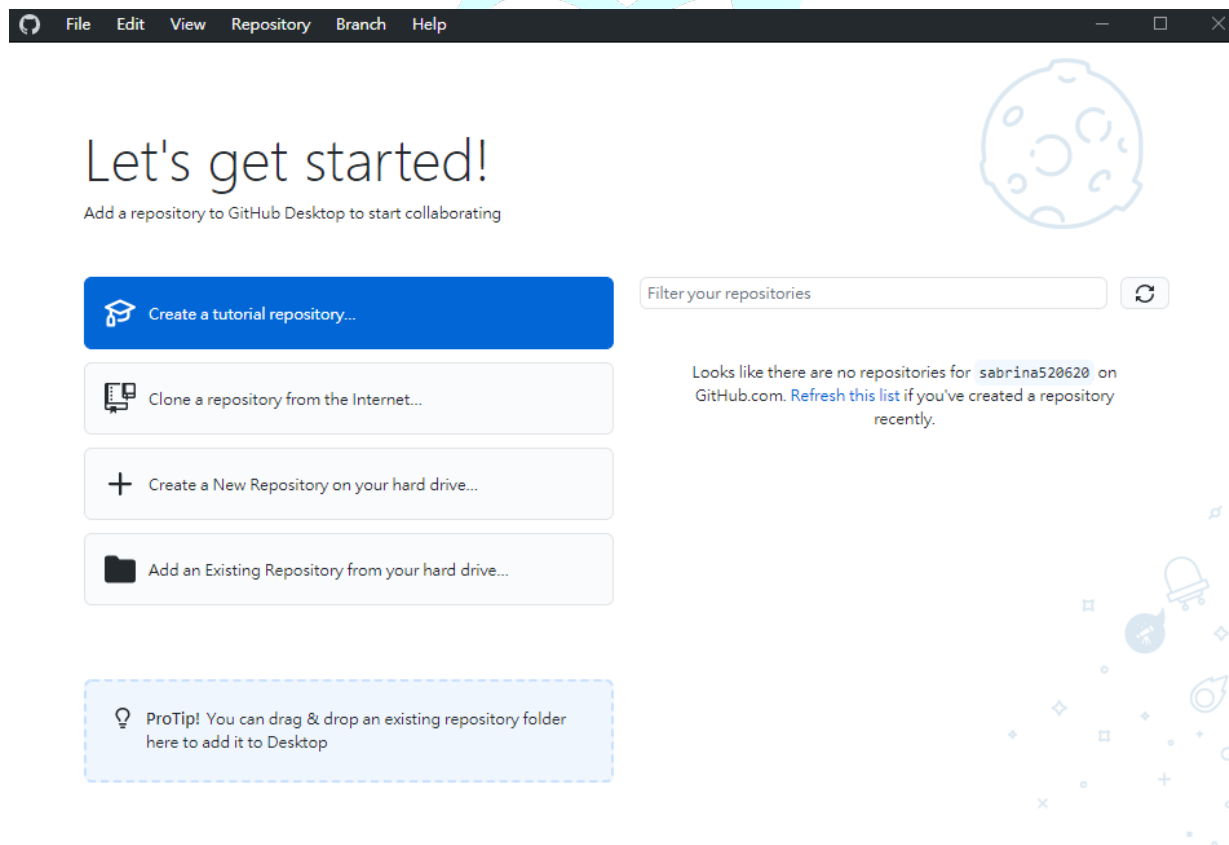
README.md



NIU_software_engineering



- GitHub 也可以根據作業系統安裝 GitHub 桌面版 當作操作工具
- 設定 GitHub 帳號密碼，完成後畫面如下：



基礎Git指令



- \$ cd ~
// cd 是切換當前工作目錄的指令，這條指令是切換到家目錄(~)
- \$ mkdir file
// mkdir: 建立一個名為file 的資料夾
- \$ cd file
// 切換到file
- \$ ls
// 查看當前目錄的所有檔案
- . : 是指當前目錄，cd . 的話，工作目錄不會改變(切換到自己)
- .. : 是指上層目錄，cd .. 的話，會切換到上層目錄
- .git : 這個就是 git 版本控管的資料庫資料夾

Git 基本設定



- 打開終端機（terminal）輸入以下指令

- 查看版本

`git --version`

- 輸入姓名：

`git config --global user.name "wasd11334342"`

- 輸入個人的 email：

`git config --global user.email "wasd11334342@gmail.com"`

- 查詢 git 設定內容：

`git config <key>`

- 指令查詢：

`git help config`

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/GitHub_tutorial
$ git config --global user.name "Morris"

u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/GitHub_tutorial
$ git config --global user.email "wasd11334342@gmail.com"
```

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/GitHub_tutorial
$ git config user.name
Morris

u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/GitHub_tutorial
$ git config user.email
wasd11334342@gmail.com
```

建立一個本機的 repository



IDAL

- repository 就是一個專案，又簡稱 repo
- 以電腦的檔案資料管理來看，我們通常會把同一個專案的資料放到同一個資料夾下，所以我們也可以把 repository 看做一個資料夾

// 建立資料夾

```
mkdir Git_exercise
```

// 移動到資料夾

```
cd Git_exercise
```

// 將專案資料夾建立成 git repository

```
git init
```

// 列出專案資料夾下的檔案和資料夾 (-l 參數為列出詳細資料，-a 為列出隱藏資料夾)

```
ls -la
```

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/GitHub_tutorial/
exercise (master)
$ ls -la
total 4
drwxr-xr-x 1 u1133 197121 0 Dec 19 08:44 ./
drwxr-xr-x 1 u1133 197121 0 Dec 19 08:44 ../
drwxr-xr-x 1 u1133 197121 0 Dec 19 08:44 .git/
```

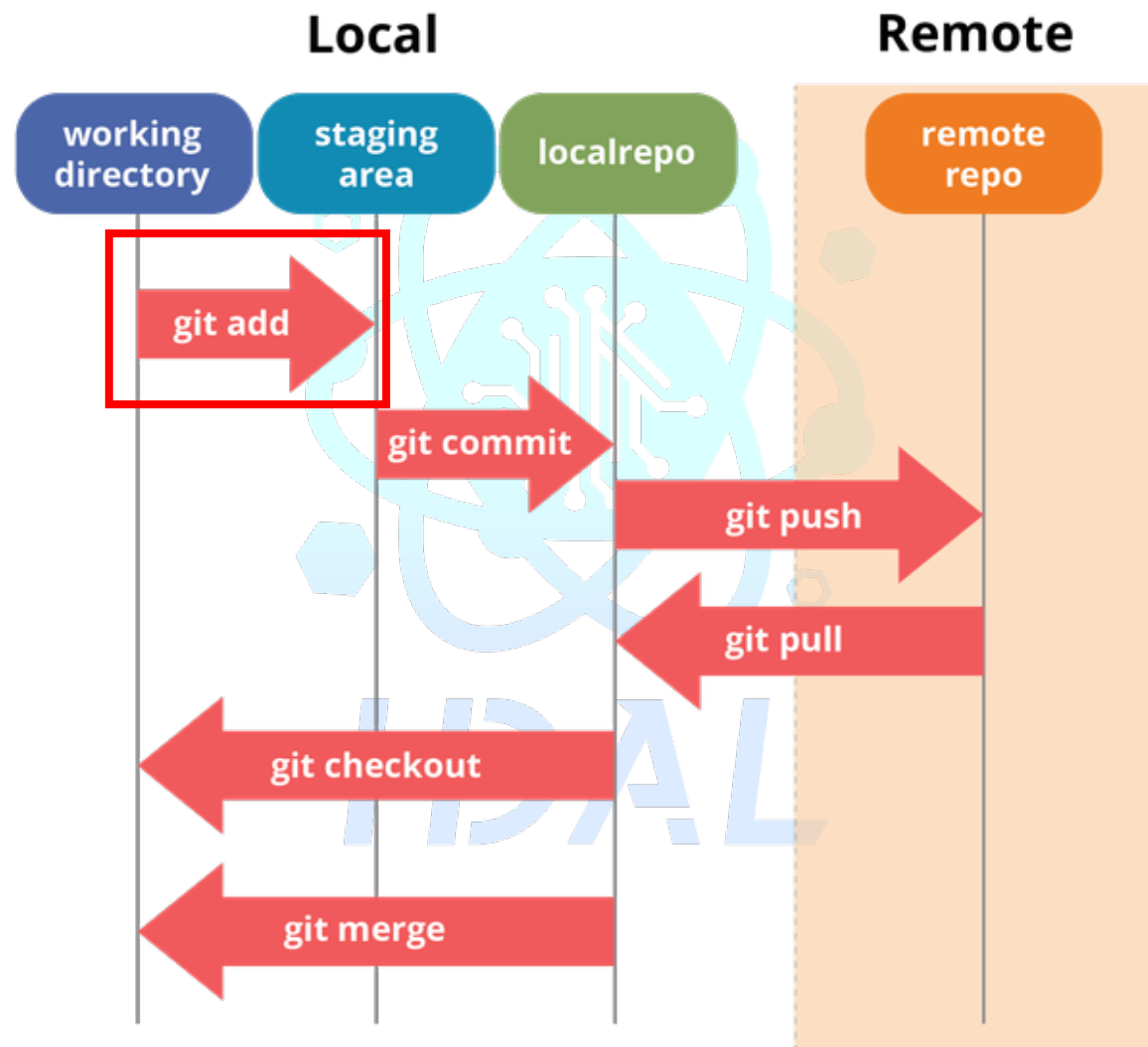


換你們做啦!!



安裝git、註冊GitHub、git 環境設定

IDAL



Add



IDAL

- 接著我們使用文字編輯器，新增一個 code.py 的檔案，程式碼如下：

```
a = int(input("enter a : "))  
b = int(input("enter b : "))  
print(a+b)
```

- 然後在終端機的專案資料夾下輸入git status顯示目前工作環境狀態：

git status

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)  
$ git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
    code.py  
  
nothing added to commit but untracked files present (use "git add" to track)
```

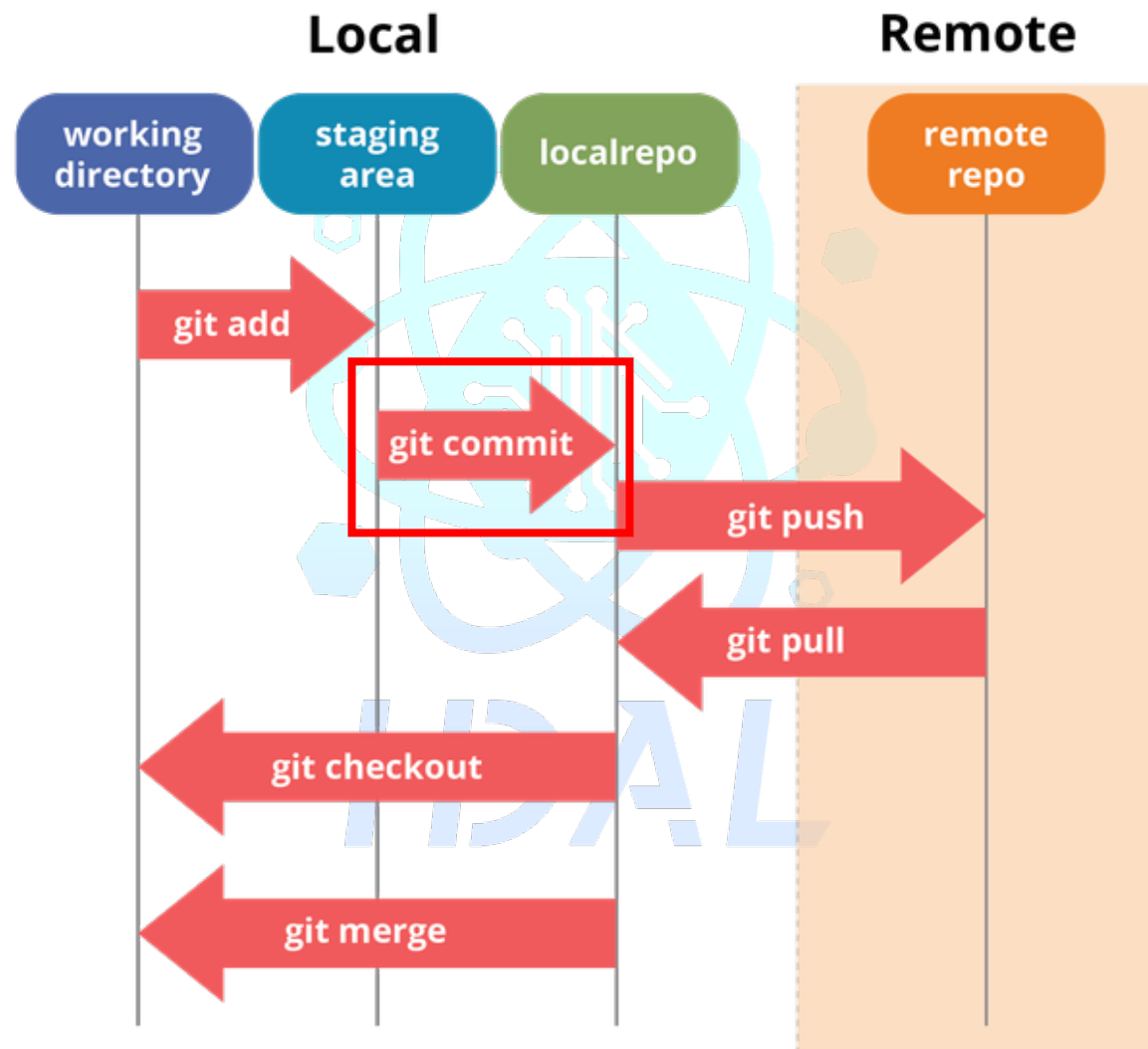
- 因為我們有新增新的檔案，但是還沒進到 git 暫存區，所以我們需要使用 git add code.py加入追蹤，這樣之後檔案有修改就可以追蹤到

git add code.py / git add --all

git status

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)  
$ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   code.py
```



Commit



- 若是確認沒問題我們就準備 commit 進去 repository

- *// -m 為輸入 commit message，也就是說這個 commit 內做了哪些事情*

git commit -m "first commit"

// commit 完成

git status

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)
$ git commit -m "first commit"
[master (root-commit) 99cc945] first commit
1 file changed, 13 insertions(+)
create mode 100644 code.py
```



- 當你追蹤後修改了檔案，例如把code.py的最後一行改成：

```
print(a-b)
```

- 若有檔案修改，記得要再add修改的檔案

// 查看目前工作狀態

```
git status
```

// 查看發生變動的地方

```
git diff
```

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   code.py

no changes added to commit (use "git add" and/or "git commit -a")
```

- commit 這個修改時簡寫會寫成這樣（-a 是 add，-m 為 message 簡寫，後面接訊息資訊）：

```
git commit -a -m "change addition to minus"
```



- 移動檔案(也可以改檔名)

```
git mv code.py minus.py
```

```
git commit -m "change file name"
```

- 查看 commit 紀錄

```
git log
```

- 細看 commit 紀錄

```
git log -p
```

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)
$ git log
commit 966079a33dadb86dfedf30c14d0c605b3b4ee33c (HEAD -> master)
Author: Morris <wasd11334342@gmail.com>
Date:   Mon Dec 19 10:26:14 2022 +0800

    change file name

commit c2cf30ed66c57da3ead4eafa931f6c02c0ca62e5
Author: Morris <wasd11334342@gmail.com>
Date:   Mon Dec 19 09:59:30 2022 +0800

    chane addition to minus

commit 99cc945b2fdbcfb7289a7bf439641d1c33dc2baf
Author: Morris <wasd11334342@gmail.com>
Date:   Mon Dec 19 09:52:48 2022 +0800

    first commit
```



換你們做啦!!

git commit

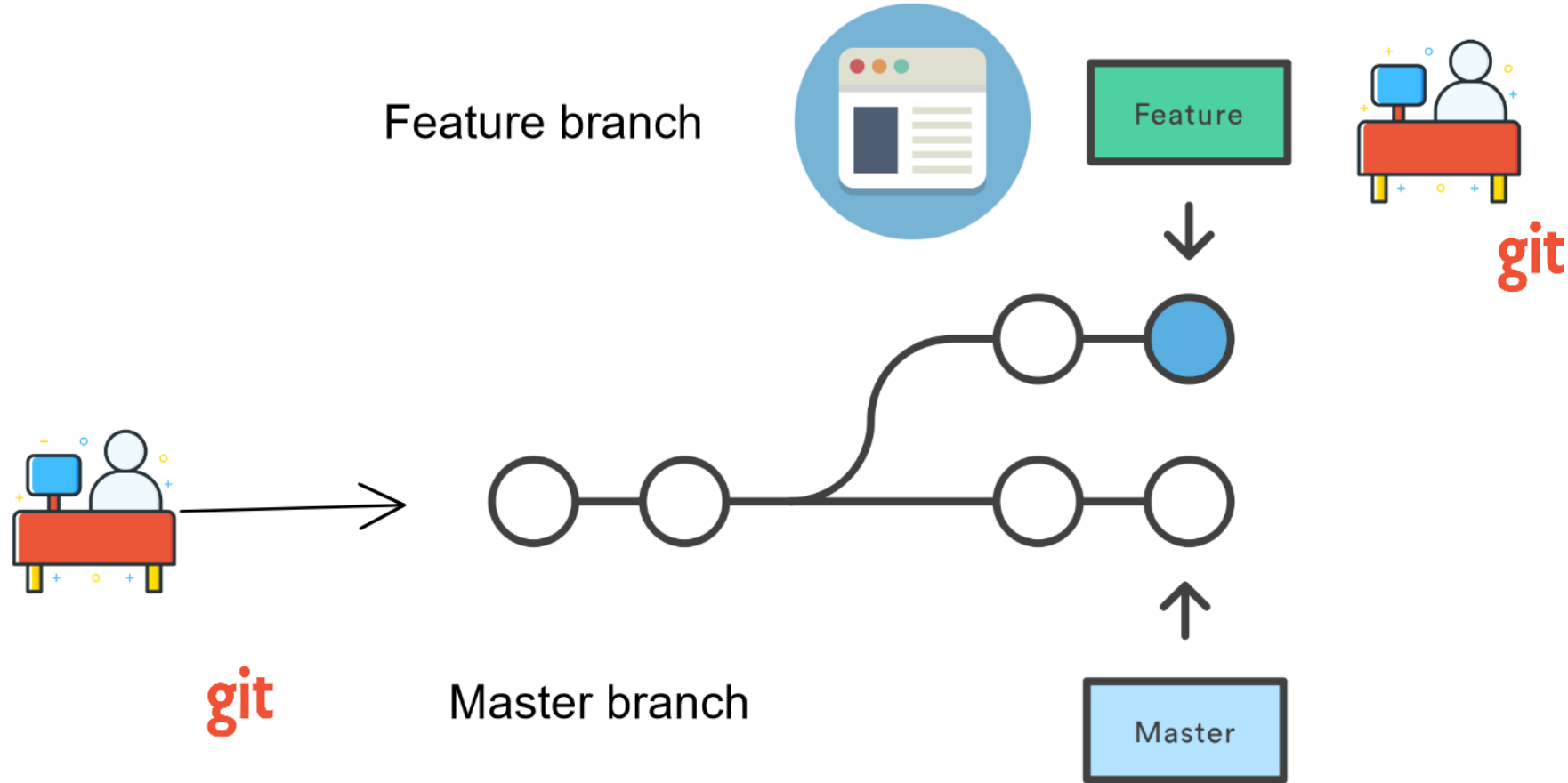
IDAL



Branch



IDAL



Git branching

練習建立一個 branch(分支)



IDAL

- 在實務開發上，因為有些開發版本是屬於「上線穩定版」、「測試版本」、「修改 BUG 版本」，但我們總不能將這些版本混在一起，會導致管理不易。所以 Git 為了解決此問題，開發了分支功能

// 查看所有分支

git branch

// 建立新的分支

git branch First_branch

git branch

// 切換主分支

git switch First_branch

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)
$ git branch
* master

u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)
$ git branch First_branch

u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)
$ git switch First_branch
Switched to branch 'First_branch'
```

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (First_branch)
$ git branch
* First_branch
  master
```


Merge（合併）和刪除 branches



IDAL

- 在分支下的改動只會改變該分支的內容，程式測試完成後若想合併到主分支則需使用merge指令

➤ 將程式碼做任意改動並commit

// 將分支合併

```
git switch master
```

// 將分支合併(把First_branch合併到master)

```
git merge -m “first merge” First_branch
```

// 刪除分支

```
git branch -d First_branch
```

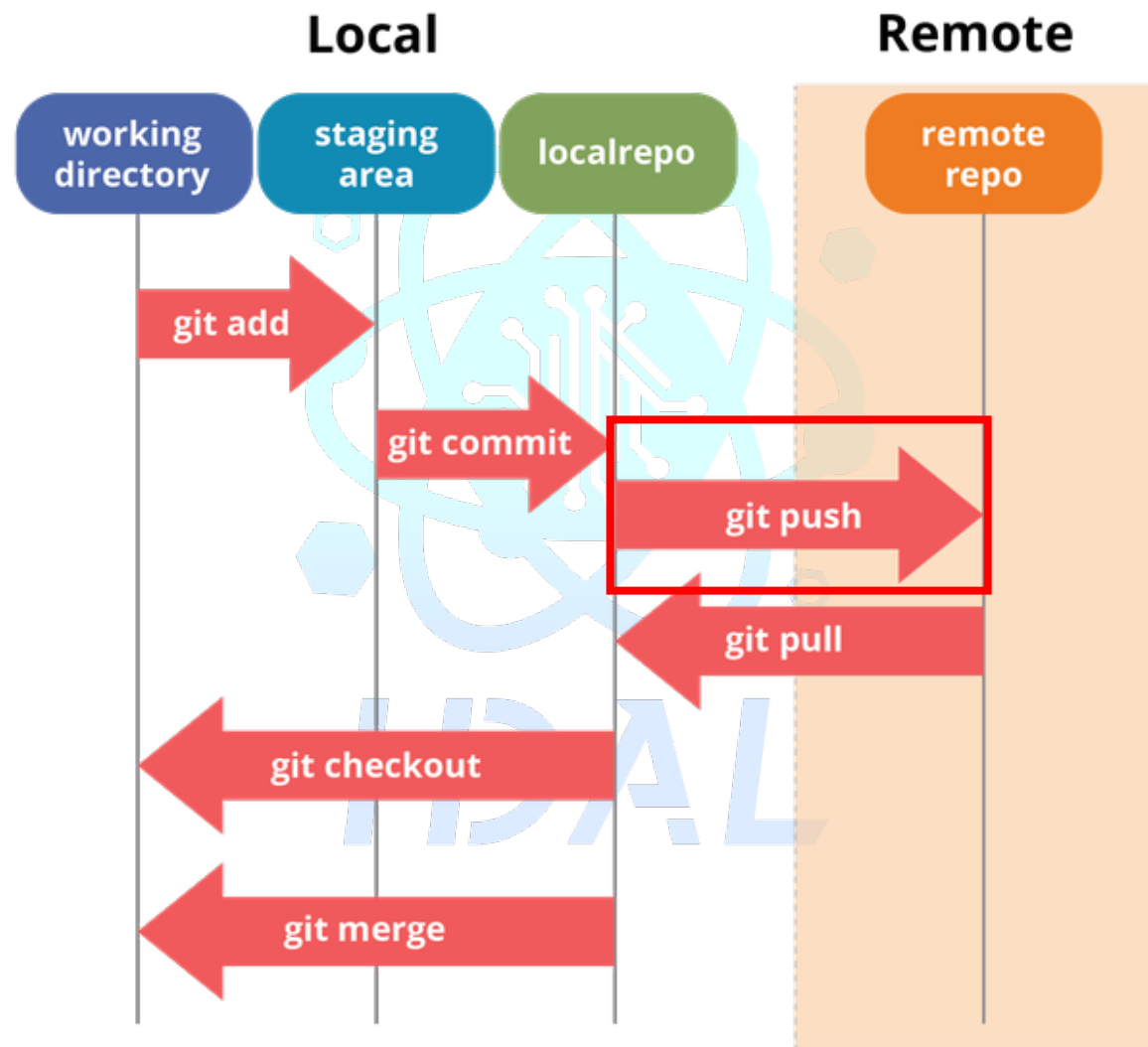


換你們做啦!!

git branch



IDAL



將 repository 做本機和遠端的連結



IDAL

// 將本地專案連結到對應遠端網址

git remote add origin <remote 網址>

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (master)
$ git remote add origin https://github.com/wasd11334342/NIU_software_engineering.git
```

- 接著準備將本地端程式 push 到遠端檔案庫：

// 更改分支名稱

git branch -M main

// 觀看情況

git status

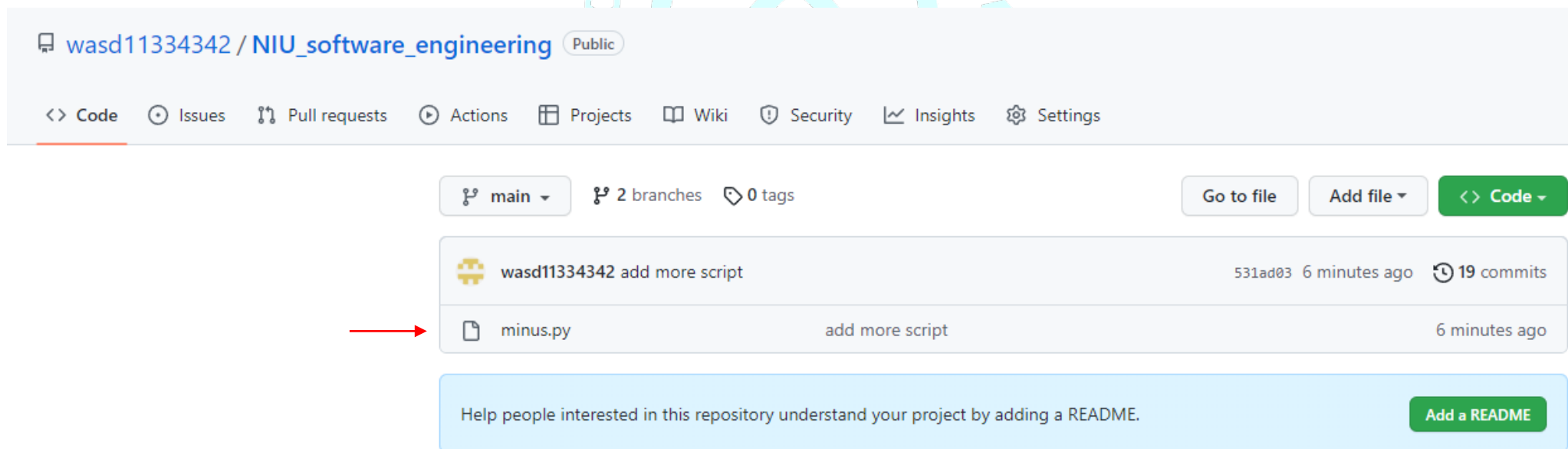
// 將本地端程式 push 到遠端檔案庫

git push -u origin master

```
u1133@DESKTOP-7NPGSEU MINGW64 ~/Desktop/git_exercise (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 334 bytes | 334.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/wasd11334342/NIU_software_engineering.git
   0920f01..531ad03  main -> main
branch 'main' set up to track 'origin/main'.
```



- Push 成功就可以在 github 上看到檔案





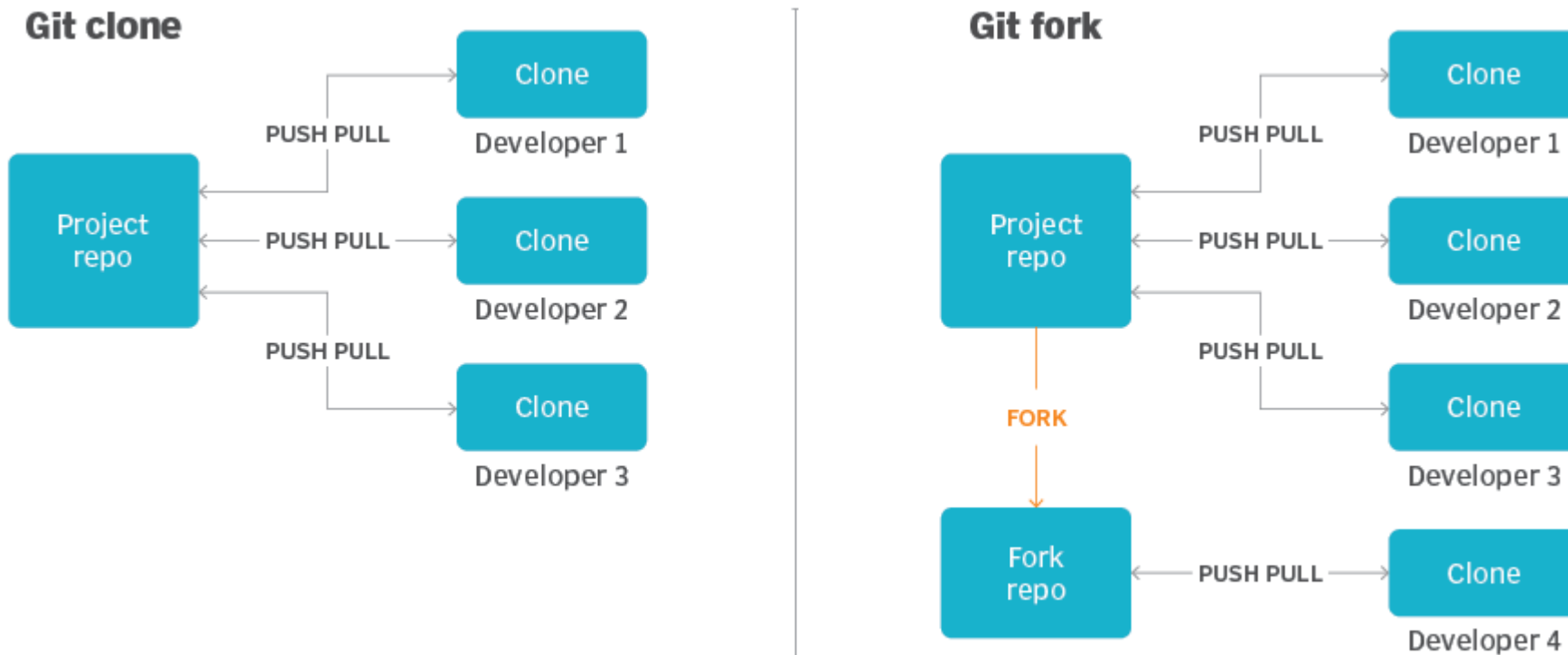
換你們做啦!!



建立local repo與GitHub repo的連結，並上傳資料

IDAL

fork 和 clone



fork 和 clone



IDAL

- **fork**：在github頁面，點擊fork按鈕。將別人的倉庫複製一份到自己的倉庫。
- **clone**：將 github 中的倉庫完整複製到自己本地電腦中。
- 點選左上角 github icon 回到首頁，我們從上面搜尋欄搜尋 **yolov5**：

The screenshot shows the GitHub search interface with the search term 'yolov5' entered in the top bar. The search results are displayed in a grid layout. On the left, there is a sidebar with navigation links: Repositories (6K), Code (575K), Commits (58K), Issues (23K), Discussions (711), Packages (6), Marketplace (0), Topics (34), Wikis (254), and Users (9). Below this, there is a 'Languages' section showing Python (3,495), Jupyter Notebook (1,503), C++ (340), and Java (41). The main content area shows '6,959 repository results' with a 'Sort: Best match' dropdown. The top three results are:

- ultralytics/yolov5**: YOLOv5 in PyTorch > ONNX > CoreML > TFLite. It has 33.8k stars, is Python-based, and has a GPL-3.0 license. It was updated 1 hour ago and has 2 issues needing help. Tags include: pytorch, deep-learning, machine-learning, ios, ml, yolo, object-detection, coreml, onnx, tflite.
- mikel-brostrom/Yolov5_StrongSORT_OSNet**: A collection of real-time multi-camera multi-object SOTA trackers using YOLOv5. It has 3.6k stars, is Python-based, and has a GPL-3.0 license. It was updated 6 hours ago. Tags include: deep-learning, segmentation, tensorrt, tracking-by-detection, yolov5, osnet, bytetrack, strongsort, ocsort.
- ppogg/YOLOv5-Lite**: YOLOv5-Lite: lighter, faster and easier to deploy. Evolved from yolov5 and the size of model is only 930+kb (int8)... It has 1.5k stars, is Python-based, and has a GPL-3.0 license. It was updated 3 weeks ago and has 1 issue needing help. Tags include: pytorch, transformer, android-app, tensorrt, mnn, mobilenet, ncnn, tflite, shufflenetv2, onnxruntime, yolov5, repvgg, pplnet, openvino, picodet.

fork



IDAL

- 點選右上角 fork 按鈕，複製一份專案到我們這：

ultralytics / yolov5 Public

Sponsor Watch 338 Fork 12.3k Star 33.8k

<> Code Issues 275 Pull requests 45 Discussions Actions Projects 1 Wiki Security Insights

master 12 branches 10 tags

Go to file Add file <> Code

developer0hye and glenn-jocher Set a seed of generator with an option for ... 10e93d2 4 hours ago 2,525 commits

github	Fix Chinese README (#10465)	5 days ago
classify	Update train.py (#10485)	2 days ago
data	YOLOv5 v7.0 release updates (#10245)	last month
models	Apply make_divisible for ONNX models in Autoshape (#10172)	last month
segment	Update Comet hyperlinks (#10500)	4 days ago
utils	Set a seed of generator with an option for more randomness when train...	4 hours ago
.dockerignore	Add .git to .dockerignore (#8815)	4 months ago
.gitattributes	git attrib	2 years ago
.gitignore	Ignore *_openvino_model/ dir (#6180)	last year
.pre-commit-config.yaml	Automatic README translation to Simplified Chinese (#10445)	last week
CITATION.cff	Update CITATION.cff to version: v7.0 (#10389)	2 weeks ago
CONTRIBUTING.md	Delete rebase.yml (#9202)	3 months ago
LICENSE	Add pre-commit CI actions (#4982)	last year
README.md	Update Comet hyperlinks (#10500)	4 days ago

About

YOLOv5 in PyTorch > ONNX > CoreML > TFLite

ultralytics.com

ios machine-learning deep-learning ml pytorch yolo object-detection coreml onnx tflite yolov3 yolov4 yolov5

Readme GPL-3.0 license Code of conduct Security policy Cite this repository 33.8k stars 338 watching 12.3k forks

Releases 10

v7.0 - YOLOv5 SOTA Realtime In... Latest

複製 HTTP 網址



IDAL

- 點選右邊綠色按鈕複製 HTTP 網址：

The screenshot shows the GitHub interface for a repository named 'wasd11334342 / yolov5', which is a public fork of 'ultralytics/yolov5'. The repository page includes a search bar, navigation links (Pull requests, Issues, Codespaces, Marketplace, Explore), and a header with repository details (Pin, Watch, Fork, Star). Below the header, there are tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Code' tab is selected, and the 'Code' button is highlighted with a red box. The repository content shows a list of files and folders, including .github, classify, data, models, and segment. The right sidebar contains links to the repository's website (ultralytics.com), README, license (GPL-3.0), code of conduct, security policy, and citation information.

clone



// 複製到本地端

```
git clone https://github.com/wasd11334342/yolov5.git
```

// 移動到 clone 下來的資料夾

```
cd yolov5
```

```
git add .
```

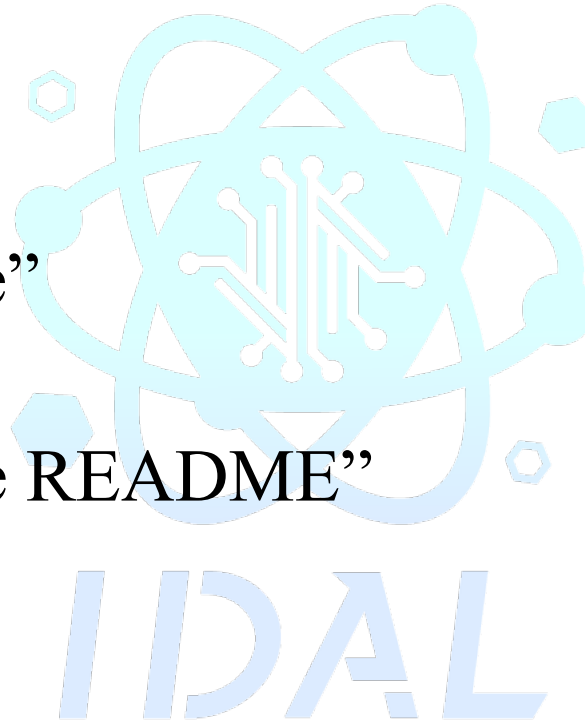
```
git commit -m "clone file"
```

// 做一些 README.md 檔案修改，然後 commit

```
git commit -a -m "Update README"
```

// 上傳到 github

```
git push origin master
```





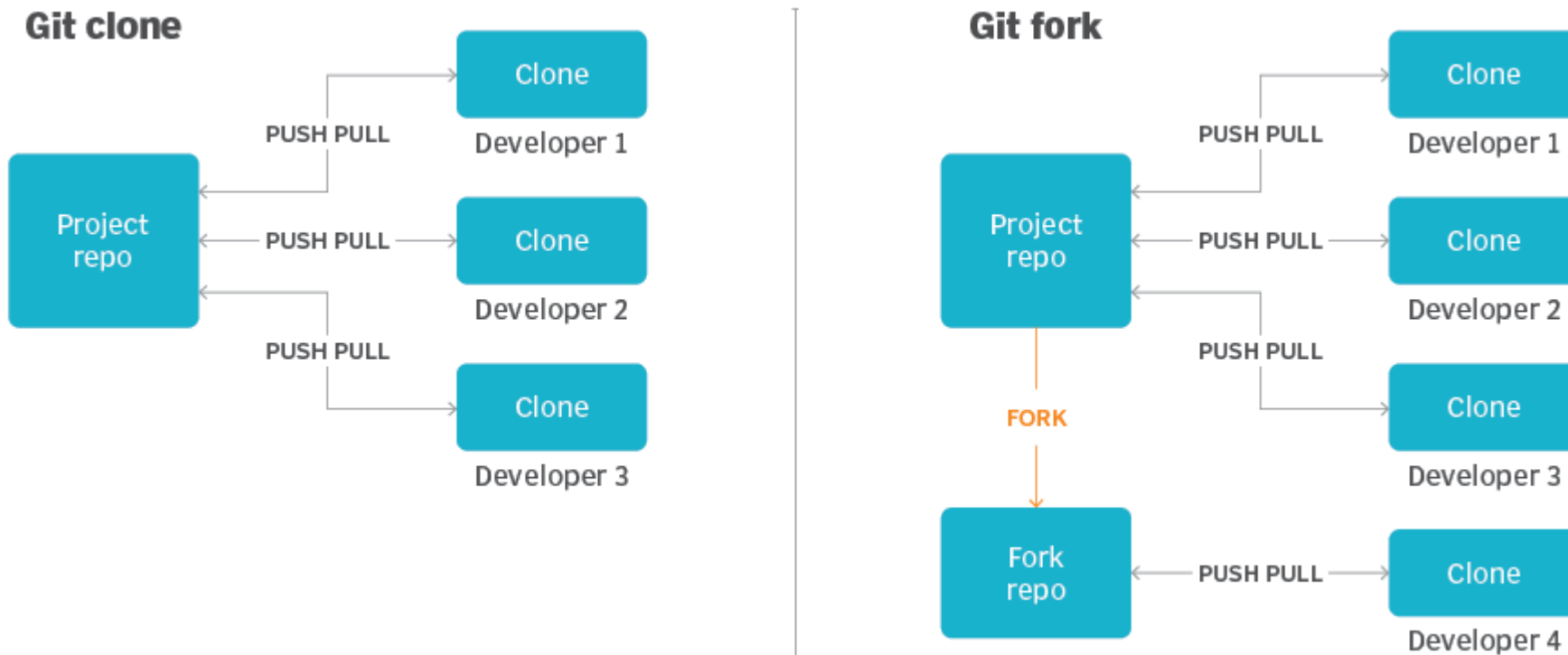
換你們做啦!!



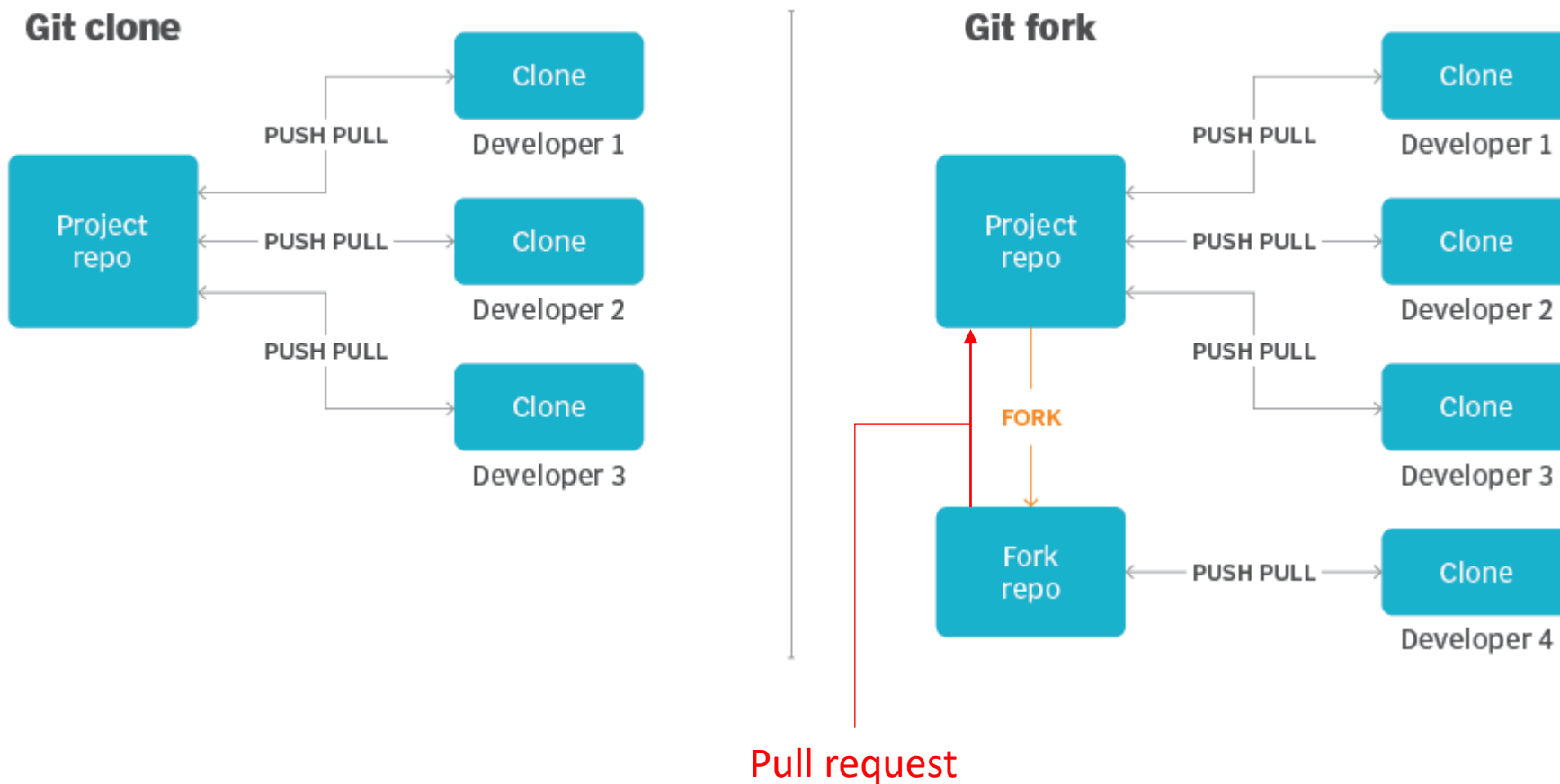
找一個repo並fork&push

IDAL

Pull request



Pull request

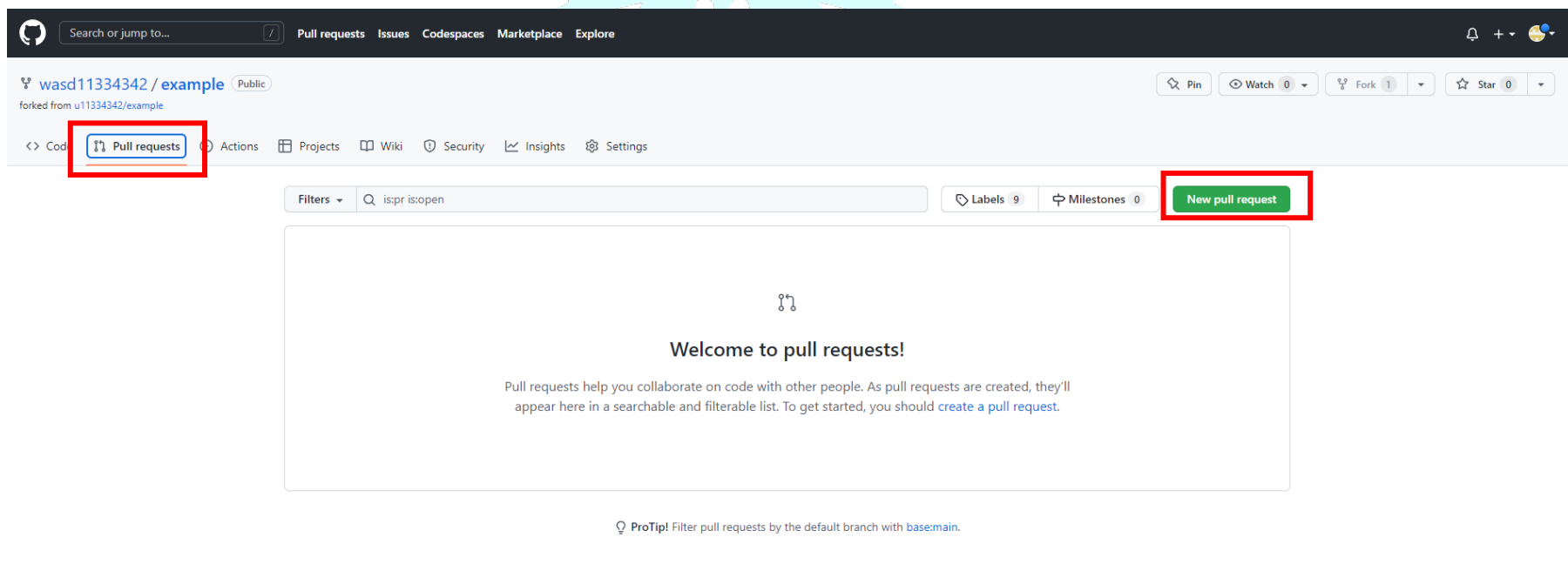


開放原始碼做出第一步貢獻



IDAL

- 你可以透過將自己的修改 commit 到自己 fork 過來的專案，然後到原始專案頁面點選 new pull request 按鈕發 pull request。
- 若對方 review 完後接受就可以將自己的程式碼合併到原始專案中



Pull request 畫面




UDAL

Fork端的pull
request畫面

example / minus.py in main Cancel changes

<> Edit file Preview changes Spaces 2 No wrap

```
1 a = int(input("enter a : "))
2 b = int(input("enter b : "))
3
4 print("sum = ", a+b)
5 print("minus = ", a-c)
6 print("multiplication = ", a*c)
7 print("division = ", a/c)
8 print("a square = ", a*a)
```

 **Commit changes**

Update minus.py

Add an optional extended description...

☒ Commit directly to the main branch.

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

Commit changes Cancel


Pull request 畫面






IDAL

Main 端(被fork端)的pull request畫面

Add more commits by pushing to the **main** branch on **wasd11334342/example**.



-  **Require approval from specific reviewers before merging**
Branch protection rules ensure specific people approve pull requests before they're merged. Add rule ×
-  **Continuous integration has not been set up**
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.
-  **This branch has no conflicts with the base branch**
Merging can be performed automatically.

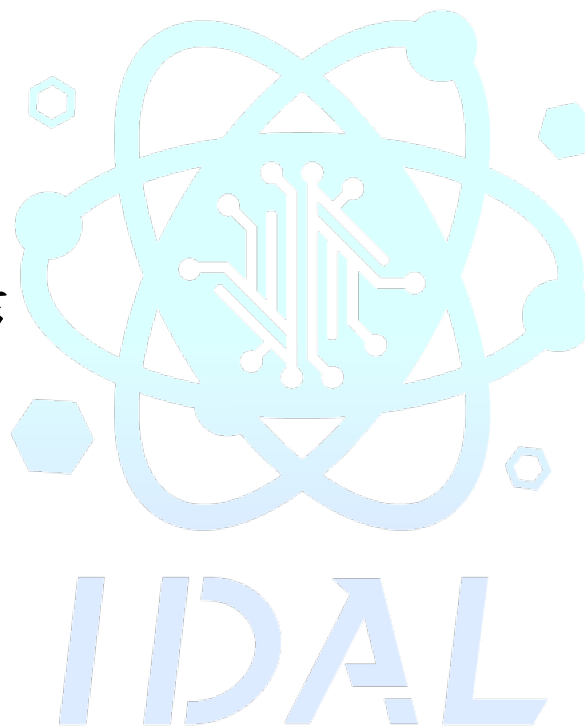
Merge pull request ▼ You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

總結



IDAL

- 安裝 Git & 註冊 GitHub 帳號
- 建立 GitHub repository
- 建立 Local repository
- Add & Commit
- 建立本機和遠端的連結
- Fork & Clone
- Pull request



課堂作業

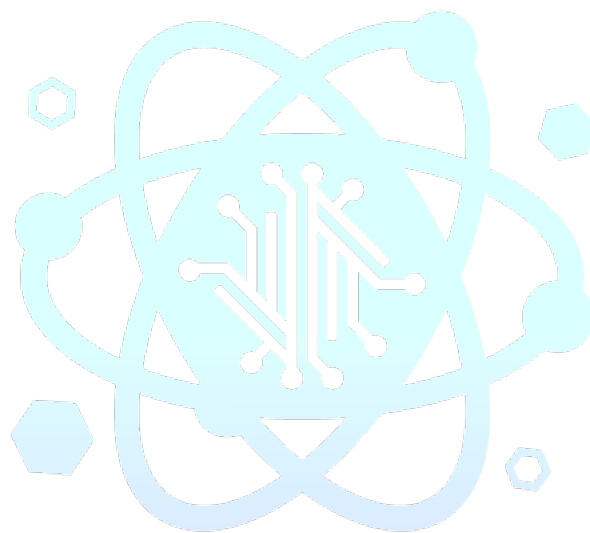


- 兩兩一組，把對方的repo fork 下來
- 對資料做任意改動，並截圖 commit 的紀錄
- 將資料 push 回 repo，並發送pull request
- 截圖最新的commit畫面

流程：

1. 於 GitHub fork repo
2. git clone 到本地端
3. 改動後截圖 git log 結果
4. git push 到fork端，截圖網頁發送pull request的結果
5. 在GitHub中截圖main端的結果

謝謝聆聽



IDAL



IDAL