



政附資訊16th - 0408社課IX

web開發基礎 實戰

社長/教學：王修佑

王修佑 hugo.tw.wang@gmail.com

先來科普一下



網址、URL、網域名稱 (Domain name)

https://www.design-hu.com(.tw)/

通訊協定

主機名稱、子網域
(Subdomain)

自定義網站名稱或
公司名字

網站內容性質識別
(學校、公司等)

申請網址的
國家機構

https://blog.design-hu.com/

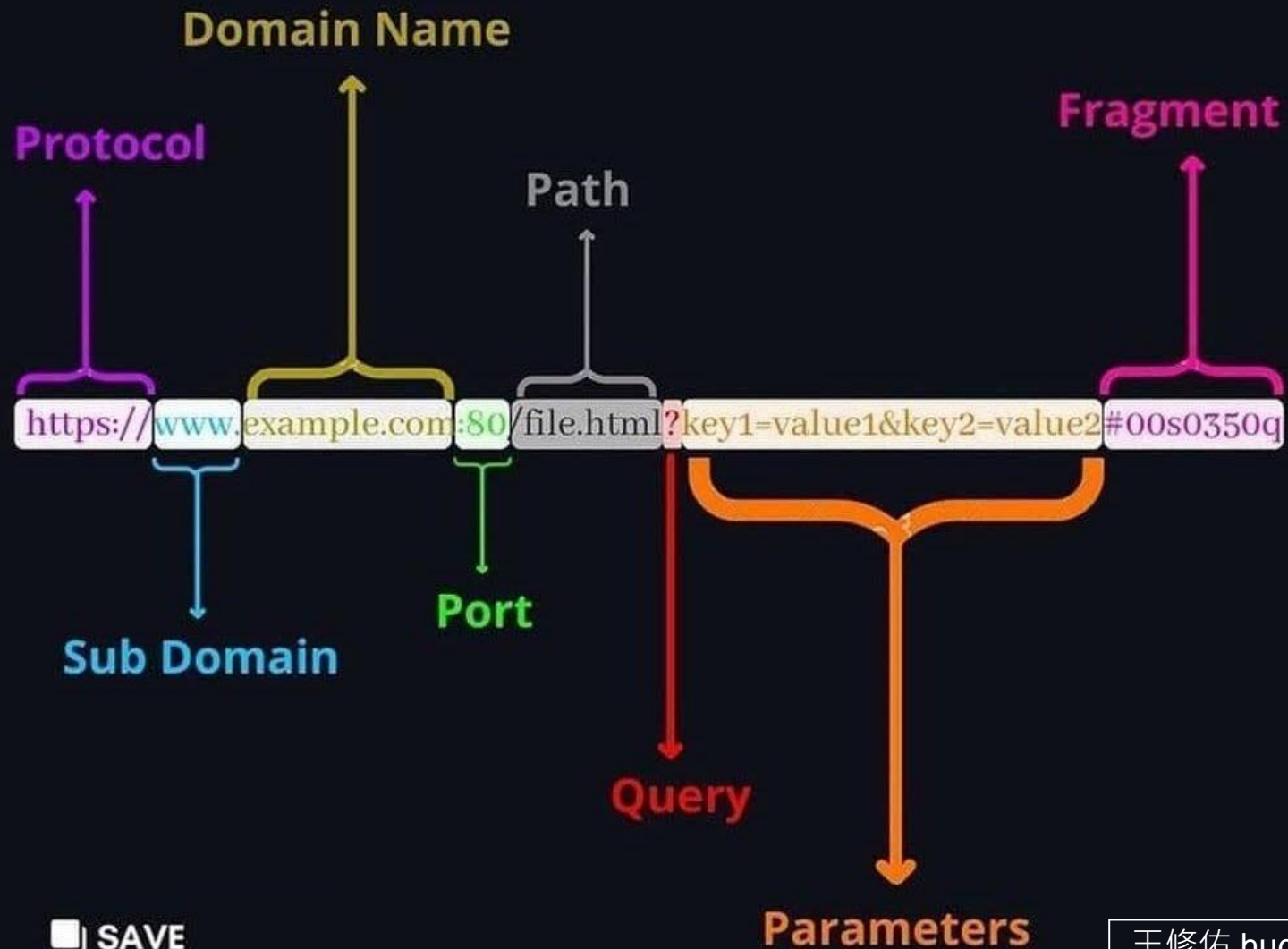
第三層網域

第二層網域

第一層網域



STRUCTURE OF URL



DOMAIN NAMES

.com	Commercial
.net	Network
.edu	Education
.org	Organization
.mil	Military
.gov	Government
.info	Information
.arpa	Infrastructure
.int	International
.mobi	Mobile
.travel	Travel
.jobs	Jobs
.ca	Canada
.coop	Cooperative



基礎概念



CSS

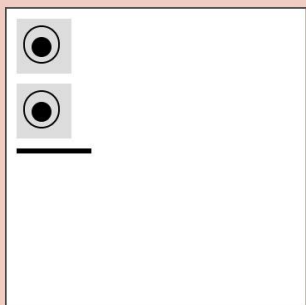
JavaScript



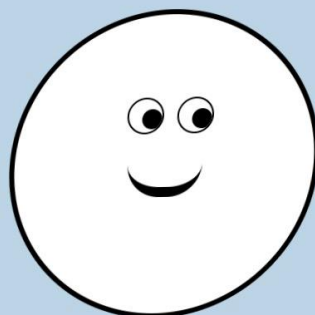
HTML



HTML



HTML + CSS



HTML + CSS + JS

Click here to PLAY!



HTML



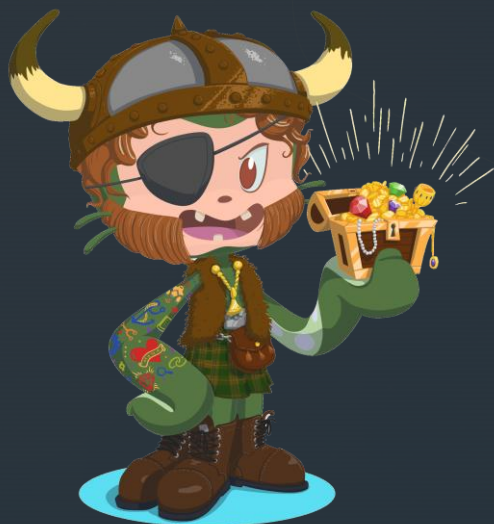
HTML+CSS



王修佑 hugo.tw.wang@gmail.com



開始之前



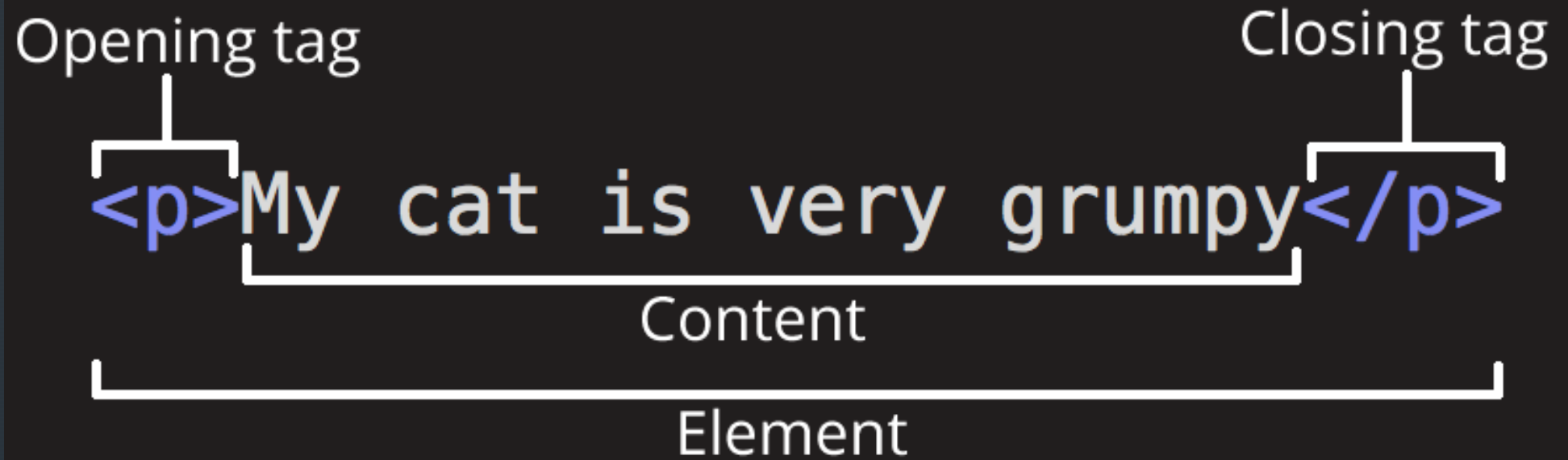
- 你需要……
 - Git & GitHub帳戶
 - Windows 記事本
 - (如果有一個不錯的IDE也許會有不少幫助)



什麼是HTML？

- HTML (Hypertext Markup Language) 是一種標記語言 (markup language)，而非一般熟知的程式設計語言；它會告訴瀏覽器該如何呈現你的網頁，組織架構並呈現網頁內容的程式語言。網頁內容的組成，可能包含了段落、清單、圖片或表格...等。





Attribute

`<p class="editor-note">My cat is very grumpy</p>`



巢狀元素

```
<p>My cat is <strong>very</strong> grumpy.</p>
```



HTML基本範例

```
<!DOCTYPE html>
<html>

  <head>
    <title>ZFCSC 16th</title>
  </head>

  <body>
    April 8th, 2022 - course IX
  </body>

</html>
```



什麼是CSS？

- CSS是用來定義網頁的樣式(Style)，像是顏色、字體大小等，還有依據銀幕大小不同，而變動的網頁佈局。

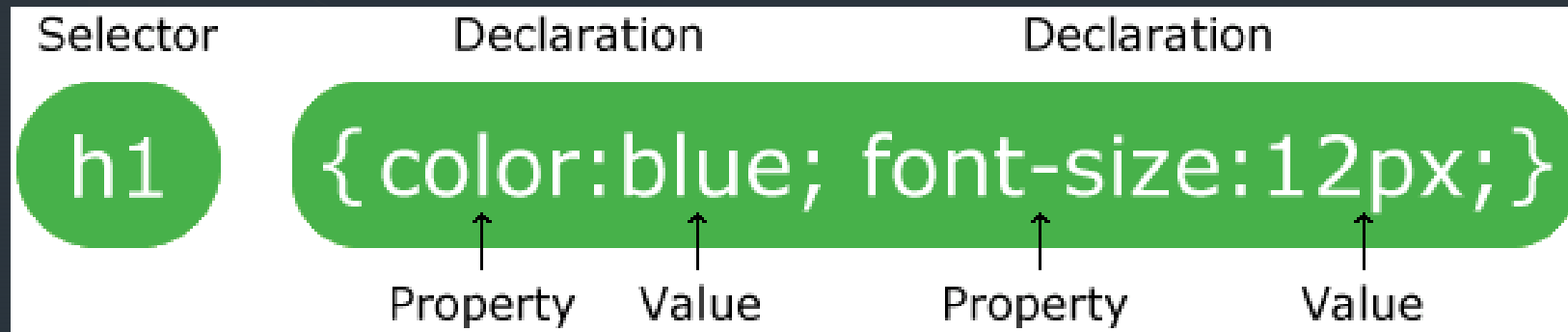


CSS解決了HTML的大問題

- HTML開始的規劃原本是不打算包含網頁格式的標籤(Tag)，他只是用來敘述網頁的內容
- 一直到HTML 3.2版才加入了像是這個標籤，還有顏色屬性。
- 但這才是惡夢的開始，因為標籤和顏色屬性佔據了大部分的網頁原始碼。
- 所以CSS被發明出來，將網頁設計獨立於HTML語法，並且方便使用。



CSS標準語法



- 示範：

https://www.w3schools.com/css/tryit.asp?filename=trycss_syntax1



selector使用標籤、id和class控制

- `<P id="para1" class="center"> 示範標題 </h1>`
- 這三者可以獨立使用，也可以合併使用

Example

```
p {  
  text-align: center;  
  color: red;  
}
```

Example

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

Example

```
.center {  
  text-align: center;  
  color: red;  
}
```

練習網址：

https://www.w3schools.com/css/exercise.asp?filename=exercise_syntax1



多個標籤或多個selector

```
h1 {  
  text-align: center;  
  color: red;  
}  
  
h2 {  
  text-align: center;  
  color: red;  
}  
  
p {  
  text-align: center;  
  color: red;  
}
```

Example

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```



使用CSS控制文字格式

- 控制行距

https://www.w3schools.com/css/tryit.asp?filename=trycss_line-height

- 控制字距

https://www.w3schools.com/css/tryit.asp?filename=trycss_text_word-spacing

- 控制英文單字中字母間的距離

https://www.w3schools.com/css/tryit.asp?filename=trycss_letter-spacing

- 其他

https://www.w3schools.com/css/css_text.asp



認識Git & GitHub

建立與使用github pages server



什麼是Git & Github





- 免費、開源
- 本機端即可操作
- 分散式版本管理系統
- 可離線使用

- 以Git為核心基礎
- 雲端版本控制服務
- *Github pages* 架站
server



Step 1

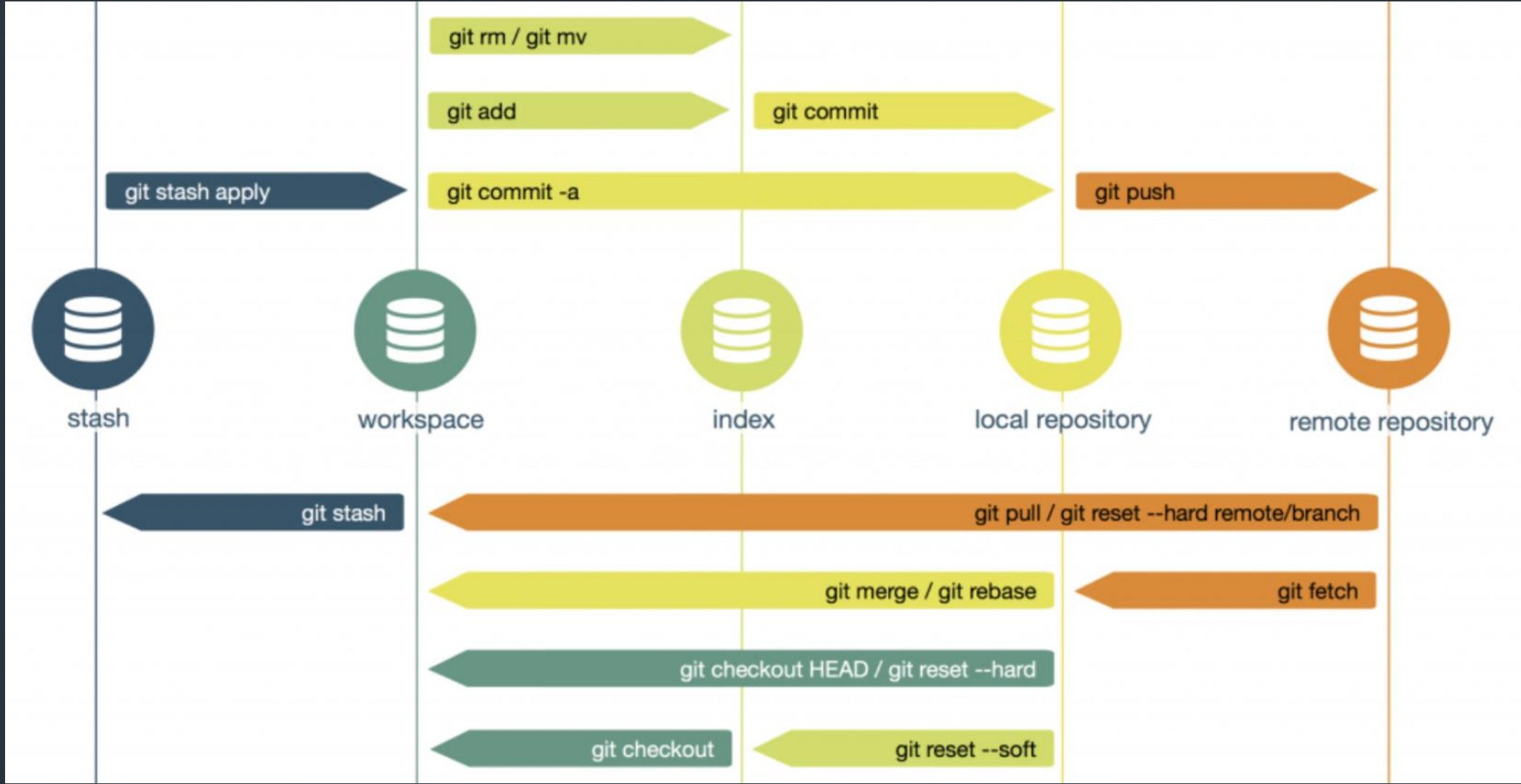
- Download Git 



Git架構

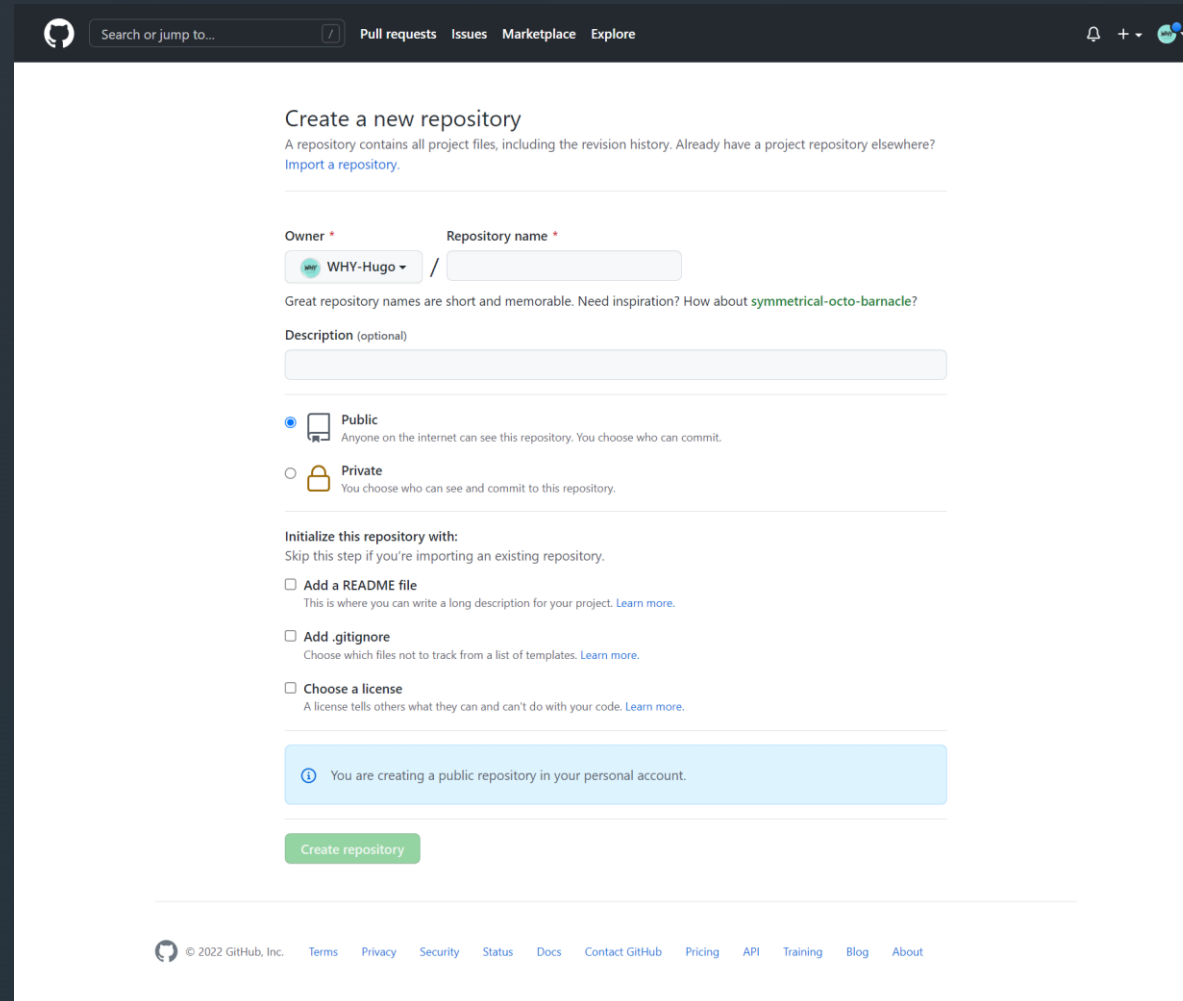
- `git init` 在 Workspace 創建 `.git` 目錄
- `git add` 建立索引
- `git commit` 至本地資料庫 (Local Repository)
- `git push` 至 GitHub 資料庫 (Remote Repository)





Step 2

■ <Github> Create a new repository



The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the main heading is 'Create a new repository', followed by a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'

The form fields include:

- Owner ***: A dropdown menu showing 'WHY-Hugo'.
- Repository name ***: An empty text input field.
- Description (optional)**: An empty text input field.
- Visibility**: Two radio buttons. 'Public' is selected, with the description 'Anyone on the internet can see this repository. You choose who can commit.' 'Private' is unselected, with the description 'You choose who can see and commit to this repository.'
- Initialize this repository with:** A section with three checkboxes:
 - ☐ **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.](#)'
 - ☐ **Choose a license**: 'A license tells others what they can and can't do with your code. [Learn more.](#)'

Below the form, there's a light blue box with an information icon and the text: 'You are creating a public repository in your personal account.'

At the bottom of the form is a green 'Create repository' button.

The footer of the page includes the GitHub logo, copyright information '© 2022 GitHub, Inc.', and a series of links: 'Terms', 'Privacy', 'Security', 'Status', 'Docs', 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.



[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)[WHY-Hugo](#) / [zfcsc_0408](#) Public[Pin](#)[Unwatch](#) 1[Fork](#) 0[Star](#) 0[Code](#)[Issues](#)[Pull requests](#)[Actions](#)[Projects](#)[Wiki](#)[Security](#)[Insights](#)[Settings](#)

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

[Set up in Desktop](#)

or

[HTTPS](#) [SSH](#)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# zfcsc_0408" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/WHY-Hugo/zfcsc_0408.git
git push -u origin main
```



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

```
git remote add origin https://github.com/WHY-Hugo/zfcsc_0408.git
git branch -M main
git push -u origin main
```



...or import code from another repository

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

[Import code](#)

ProTip! Use the URL for this page when adding GitHub as a remote.



© 2022 GitHub, Inc.

[Terms](#)[Privacy](#)[Security](#)[Status](#)[Docs](#)[Contact GitHub](#)[Pricing](#)[API](#)[Training](#)[Blog](#)[About](#)

王修佑 hugo.tw.wang@gmail.com



Step 3

- 開啟Command Prompt

 + R >> 輸入cmd

- 進入Working Directory

- 切換記憶體位置(C、D槽)：直接輸入 (例如 `$ D:`)
- 進入下一層資料夾：`$ cd folder_name`
- 退一層資料夾：`$ cd..`

*補充：按F3可自動輸入前一次指令



Step 4

- Workspace 創建 .git 目錄

```
$ git init
```

- 初次於local端使用需設定git config 環境

```
$ git config --global user.name "username"
```

```
$ git config --global user.email "email"
```



Step 5

進行github setup

- 方法A

- 從 Github 下載 Remote Repository to Workspace

```
$ git clone <repo URL>
```

- 從 Github 下載 Remote Repository，並且指定分支

```
$ git clone <repo URL> -b <branch name>
```

- 從 Github 下載 Remote Repository，並且指定下載名稱

```
$ git clone <repo URL> <folder name/path>
```



進行github setup

- 方法B

```
echo "# Repo name" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/user/Repo name.git
git push -u origin main
```



Step 6

Git 傳送檔案

- 上傳三步驟

- 工作目錄 (Working Directory) 內所有檔案建立索引

```
$ git add *
```

- 將建立索引的檔案提交至本地資料庫 (Local Repository)

```
$ git commit -m "annotation"
```

- git push 至 GitHub

```
$ git push origin
```



Git常用指令

- 查看 Git config 設定

```
$ git config --list
```

- Working Directory檔案狀態

```
$ git status
```

- <Git commit 記錄>顯示分支的提交詳細資訊

```
$ git log
```

- <Git commit 記錄>commit 內容

```
$ git reflog
```

- 查看目前 Remote 連線位置

```
$ git remote -v
```



Step 7

- 進入Repository >> Setting
- >> GitHub Pages

GitHub Pages

Pages settings now has its own dedicated tab! [Check it out here!](#)

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at <https://why-hugo.github.io/>

Source

Your GitHub Pages site is currently being built from the master branch. [Learn more.](#)

🔗 Branch: master ▼

📁 / (root) ▼

Save

王修佑 hugo.tw.wang@gmail.com





國立政大附中資訊社

AHSNCCU Computer Science Club

王修佑 hugo.tw.wang@gmail.com