

## python 虛擬環境

下載 python

<https://www.python.org/>

安裝套件

pip install virtualenv

建置虛擬環境

開啟 cmd(命令提示字元)

cd 移到適當資料夾位置

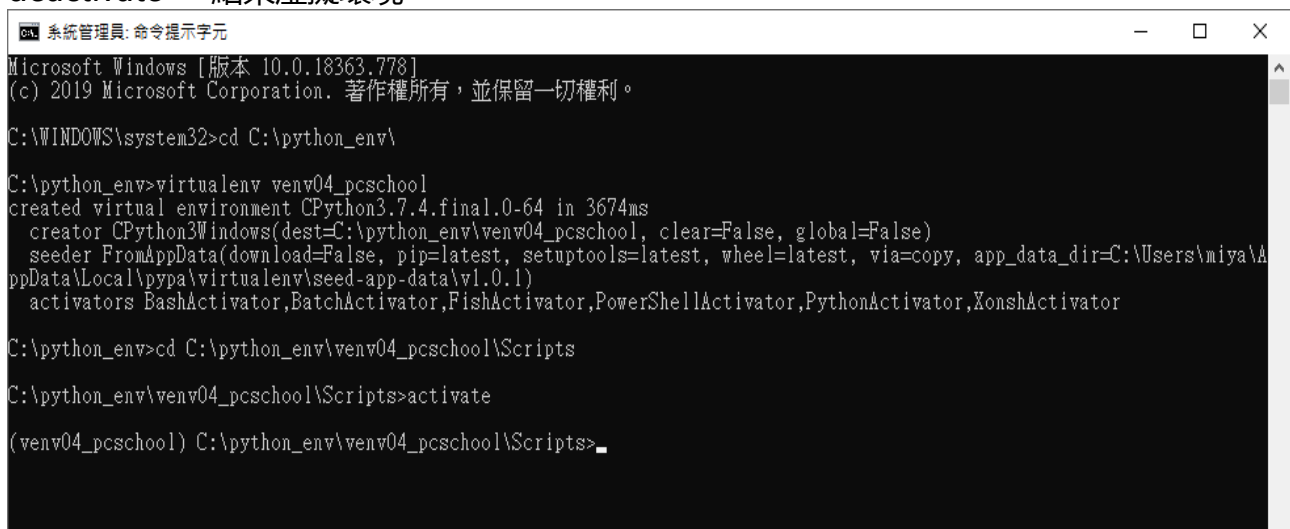
輸入 virtualenv xxx(名字)

產生相關資料後，cd 移到 Scripts 位置

輸入 activate

(虛擬環境名稱)=>即成功建立完成

deactivate=>結束虛擬環境



```
ca 系統管理員: 命令提示字元
Microsoft Windows [版本 10.0.18363.778]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\WINDOWS\system32>cd C:\python_env\

C:\python_env>virtualenv venv04_pcschool
created virtual environment CPython3.7.4.final.0-64 in 3674ms
  creator CPython3Windows(dest=C:\python_env\venv04_pcschool, clear=False, global=False)
  seeder FromAppData(download=False, pip=latest, setuptools=latest, wheel=latest, via=copy, app_data_dir=C:\Users\miya\AppData\Local\pypa\virtualenv\seed-app-data\v1.0.1)
  activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator

C:\python_env>cd C:\python_env\venv04_pcschool\Scripts

C:\python_env\venv04_pcschool\Scripts>activate

(venv04_pcschool) C:\python_env\venv04_pcschool\Scripts>_
```

## VS Code 虛擬環境建立

<https://code.visualstudio.com/>

get-executionpolicy 確認 restricted

set-executionpolicy remotesigned

.\activate=>(建立虛擬) deactivate(結束虛擬)

## DB Browser for SQLite

<https://sqlitebrowser.org/dl/>

---

## Flask 本機端預覽

```
pip install flask
set flask_app=hello.py
set flask_debug=1
flask run
```

---

## 上傳遠端 Heroku 空間

下載 heroku cli

<https://devcenter.heroku.com/articles/heroku-cli>

下載 git

<https://git-scm.com/downloads>

heroku 中 create new app(例如:pcschool01)

開啟 cmd(命令提示字元)

<https://devcenter.heroku.com/articles/git>

heroku login

cd 切換到準備上傳到 heroku 的資料夾中

git init

heroku git:remote -a 專案名稱(例如:pcschool01)

git add .

git commit -m "init"

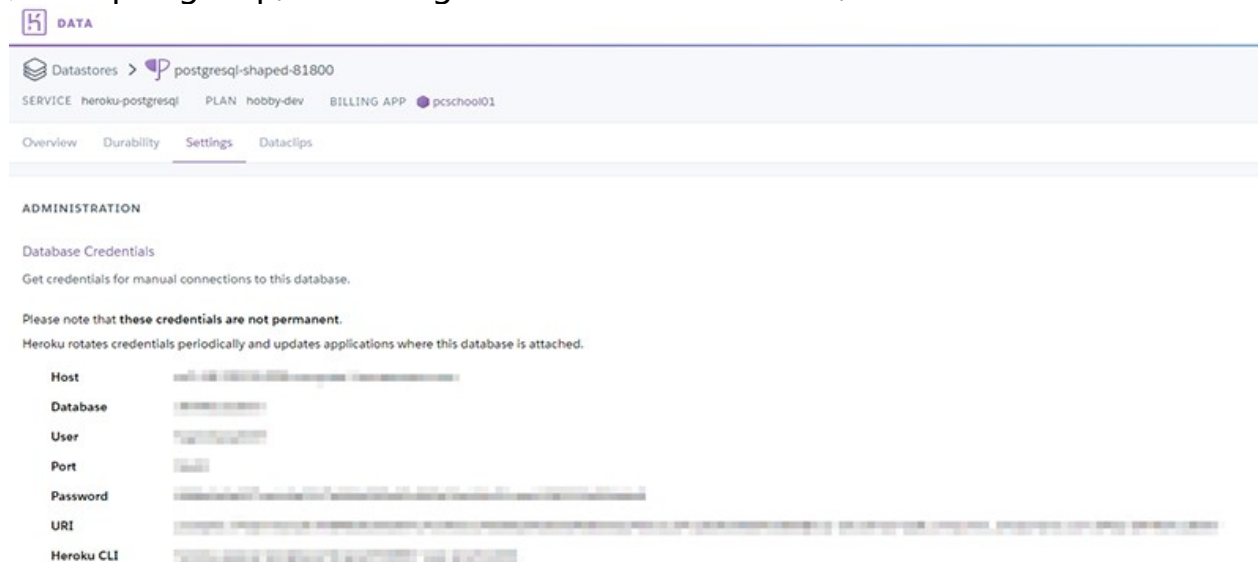
git push heroku master

git logs=>(除錯參考)

## Heroku Postgres 資料庫連線設定

專案下 Installed add-ons , 選擇 Postgres

建立好 postgresql, 於 Settings 下 Database Credentials, 選擇 View Credentials...



安裝 pgadmin

<https://www.pgadmin.org/download/>

---

## Heroku ClearDB MySQL 資料庫連線設定

專案下 Installed add-ons , 選擇 ClearDB

安裝 mysql workbench

<https://dev.mysql.com/downloads/workbench/>

---

## 安裝 ngrok

<https://ngrok.com/download>

## line and ngrok

The image shows a multi-step process for setting up ngrok to connect a local Flask application to the LINE Developers console. It includes the LINE Developers console interface, a terminal window running ngrok, and a code editor showing the Flask application code.

**1** In the terminal, run the command: `ngrok http 5000`

**2** The terminal output shows the ngrok status and the public URL: `https://d642280c.ngrok.io`

**3** In the LINE Developers console, go to the Webhook settings and set the Webhook URL to `https://d642280c.ngrok.io/callback`.

**4** In the code editor, the Flask application code is shown. The `if __name__ == '__main__':` block sets the port to `5000` and runs the app on `host='127.0.0.1'`.

**5** In the terminal, the Flask application is run using `flask run`.