

表單



Form

```
<p>GET submit</p>
<form action="get_name" method="GET">
  姓名: <input name="your_name" type='text'>
  <br><br>
  <input type='submit' value='送出表單'>
</form>

<br><br>
<p>POST submit</p>
<form action="post_login" method="POST">
  姓名: <input name="your_name" type='text'><br><br>
  密碼: <input name="your_password" type='password'>
  <br><br>
  <input type='submit' value='送出表單'>
</form>
```

<form action="目的地" method="傳遞方式">

表單內容、說明文字

...

...

表單輸入元素 <input> <option> <textarea>

...

表單按鈕 <button>

送出表單 type='submit'

</form>

0_form.html

GAIS Form Demo

GET submit

姓名：

POST submit

姓名：

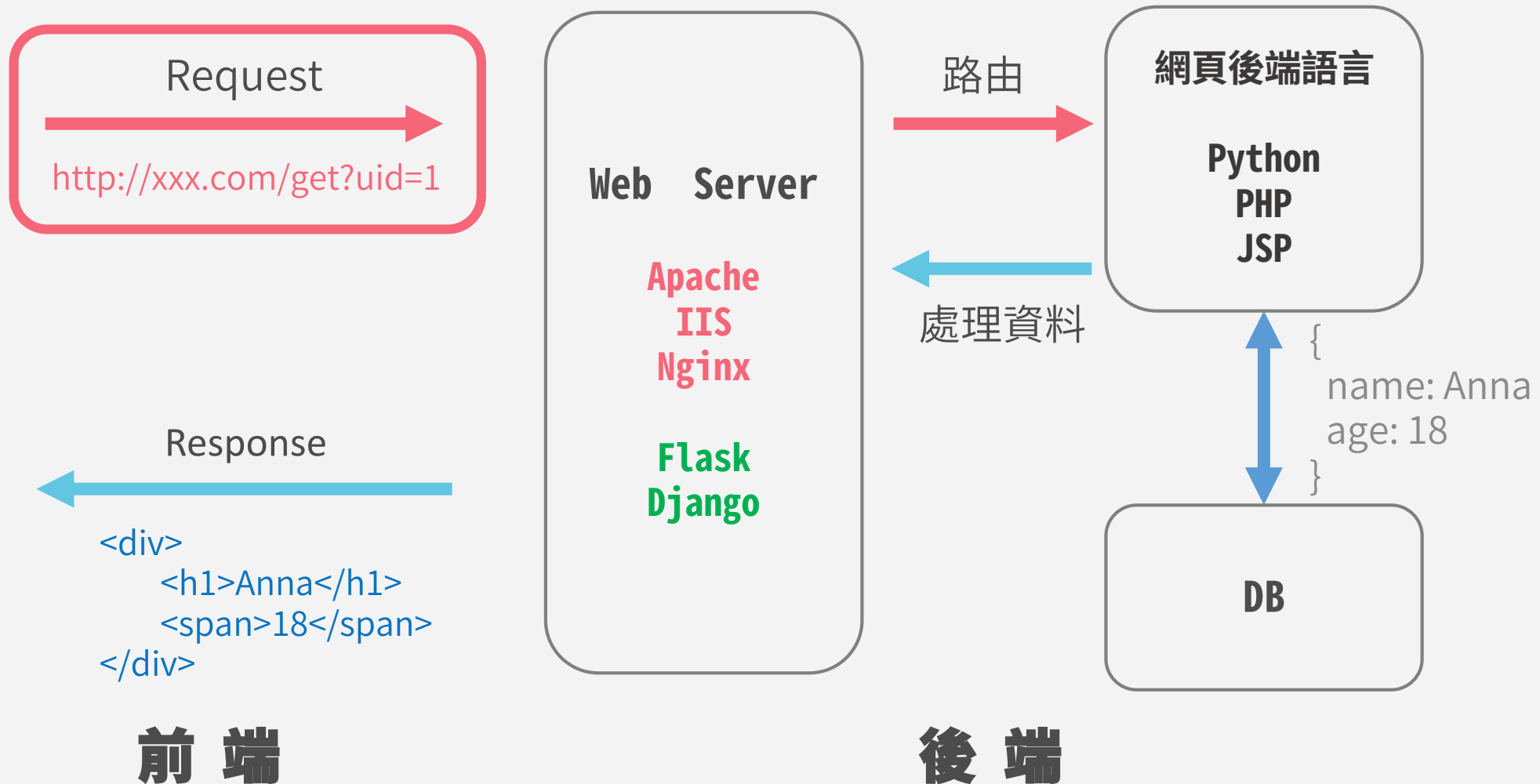
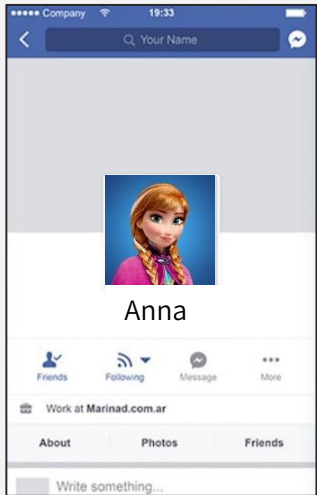
密碼：

Copyright (c) 2019-2022

GAIS Lab 智皓澈女友



網頁怎運作的？



HTTP



GET、POST

GET

語義:

獲取、查詢、檢索

描述:

給我「<http://xx.com/Anna>」這個資源

大部分上網的請求都是GET，
他會把表單的參數置於URI上。

<http://xx.com/hello?name=Anna>

POST

語義:

提交、發布

描述:

提交一份資料表單
新增 or 建立資源 (上傳檔案)

僅次於GET，最常見的用法，
主要參數置於Message body上。

Demo <http://live.ccu.edu.tw/admin/>

PUT、DELETE

PUT

語義:

置換、更新

描述:

以請求payload**更新**特定資源

DELETE

語義:

刪除

描述:

移除 目標中所匹配到的資源

Python

- 物件導向的高階程式語言、直譯語言
- 易讀、易懂、易學
- 跨平台、支援Unicode
- 豐富的函式庫、龐大的交流社群

開發工具 怎麼選



PyCharm



**Anaconda
+ Spyder**



VS code



Vim

大型專案

Web後端開發

資料分析

科學計算

機器學習

快速入門

基礎開發

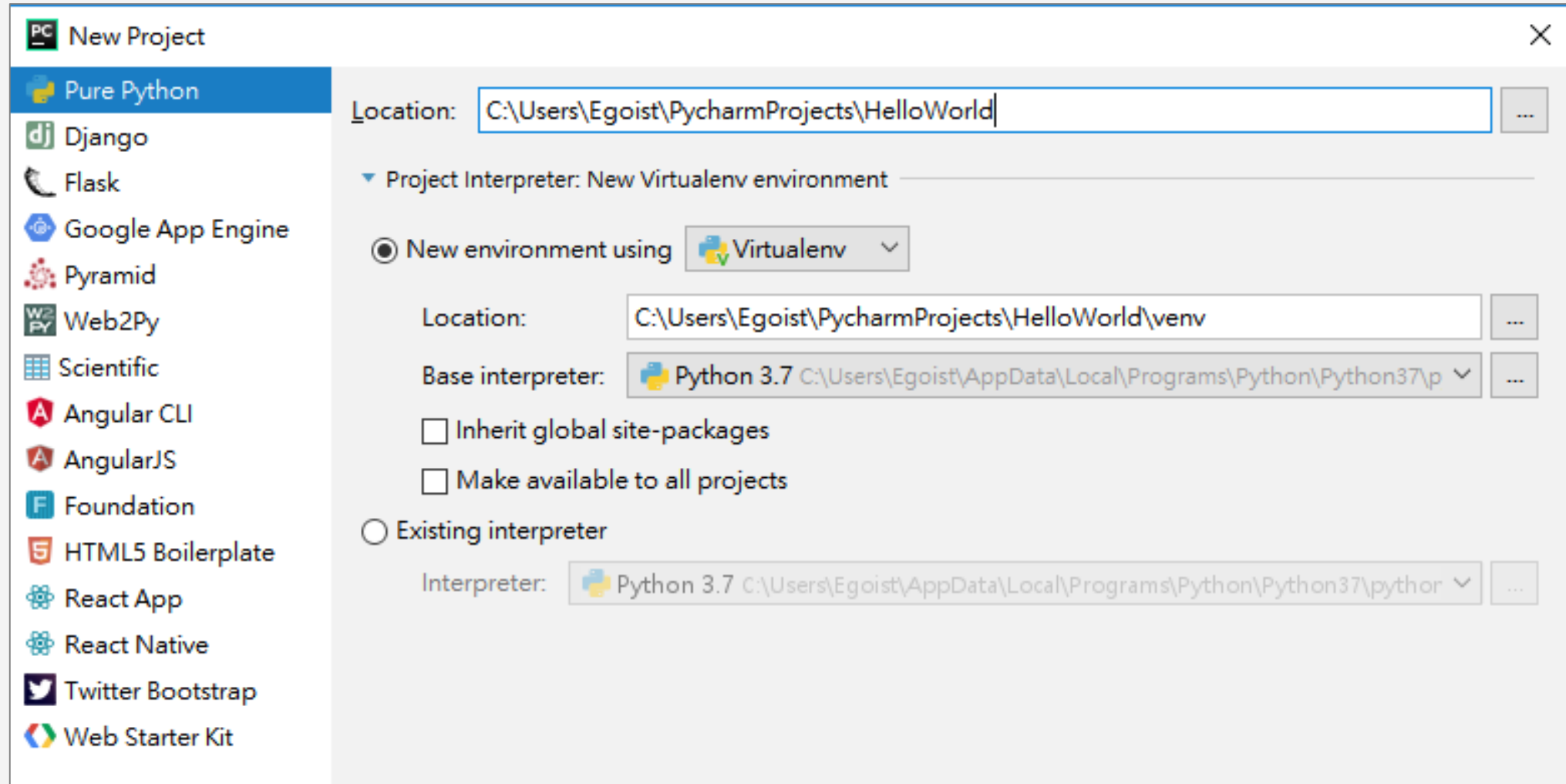
Hacker

Python

Hello world



New Project



The image shows the 'New Project' dialog in PyCharm. On the left is a list of project templates: Pure Python (selected), Django, Flask, Google App Engine, Pyramid, Web2Py, Scientific, Angular CLI, AngularJS, Foundation, HTML5 Boilerplate, React App, React Native, Twitter Bootstrap, and Web Starter Kit. The main area on the right is for configuring the 'Pure Python' project. It has a 'Location' field set to 'C:\Users\Egoist\PycharmProjects\HelloWorld'. Below this, the 'Project Interpreter' section is expanded, showing 'New Virtualenv environment' as the selected option. Under this, 'New environment using' is set to 'Virtualenv'. The 'Location' for the virtual environment is 'C:\Users\Egoist\PycharmProjects\HelloWorld\venv'. The 'Base interpreter' is 'Python 3.7' located at 'C:\Users\Egoist\AppData\Local\Programs\Python\Python37\python.exe'. There are checkboxes for 'Inherit global site-packages' and 'Make available to all projects', both of which are unchecked. At the bottom, the 'Existing interpreter' option is also visible but not selected, with its interpreter path set to 'Python 3.7 C:\Users\Egoist\AppData\Local\Programs\Python\Python37\python.exe'.

PC New Project

Pure Python

Django

Flask

Google App Engine

Pyramid

Web2Py

Scientific

Angular CLI

AngularJS

Foundation

HTML5 Boilerplate

React App

React Native

Twitter Bootstrap

Web Starter Kit

Location: C:\Users\Egoist\PycharmProjects\HelloWorld

Project Interpreter: New Virtualenv environment

☒ New environment using Virtualenv

Location: C:\Users\Egoist\PycharmProjects\HelloWorld\venv

Base interpreter: Python 3.7 C:\Users\Egoist\AppData\Local\Programs\Python\Python37\python.exe

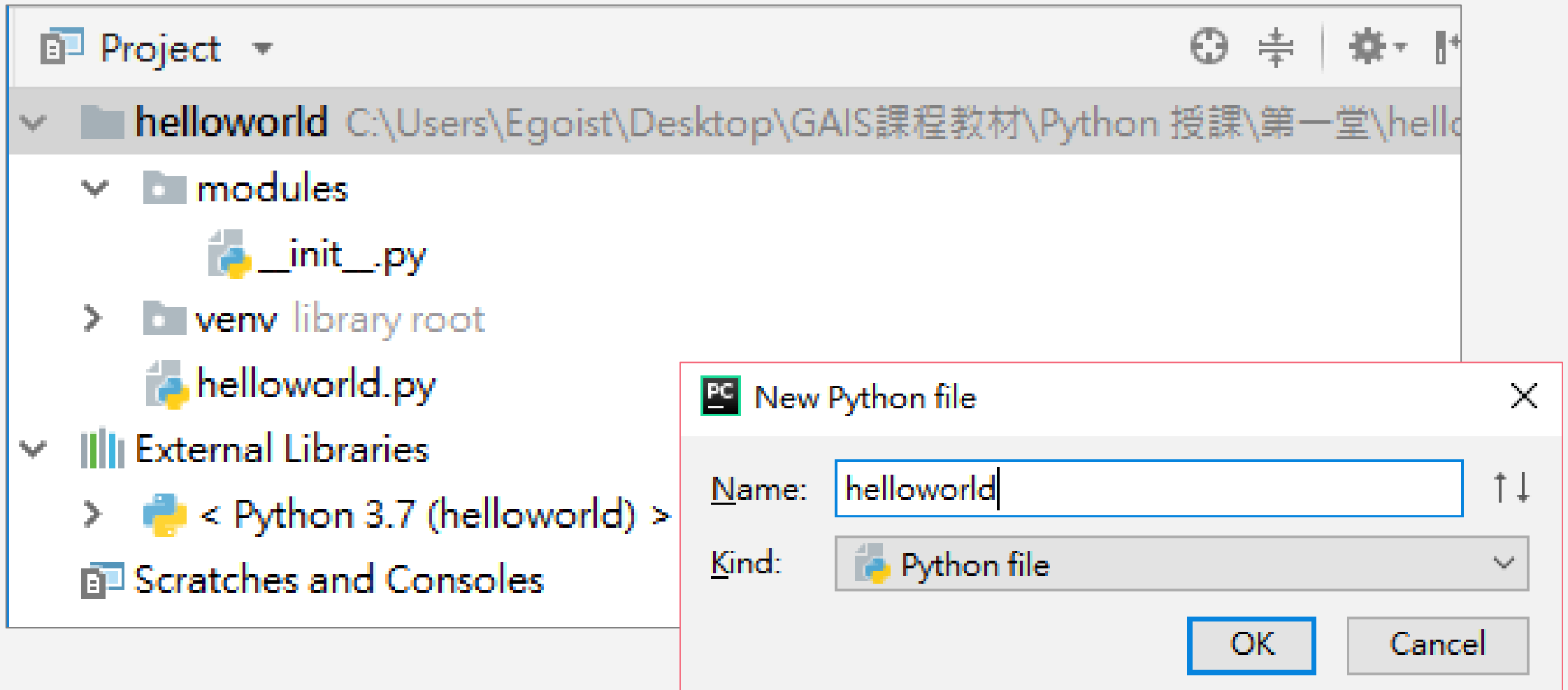
☐ Inherit global site-packages

☐ Make available to all projects

☐ Existing interpreter

Interpreter: Python 3.7 C:\Users\Egoist\AppData\Local\Programs\Python\Python37\python.exe

New Project



Hello World

```
#!/usr/bin/python3
```

指定用什麼直譯器執行

```
hw = "Hello World"
```

宣告字串

```
if __name__ == "__main__":  
    print(hw)
```

程式入口點

程式主體 (注意縮排)



入口點

```
#!/usr/bin/python3
```

```
hw = "Hello World"
```

```
if __name__ == "__main__":  
    print(hw)
```

- 印出 `__name__` 變數
- 試試看 import 其他py文件
- 觀察本體跟import模組的差別
- 嘗試其他內建變數 `__file__`

變數

- 開頭可以大小寫字母或 `_`，不能數字
- 不可跟內建關鍵字同名 (ex. `def`、`if`、`else`)
- 大小寫視為不同變數
- Python3 支援 Unicode (中文可以！)

資料型態

- 整數

int (不限範圍整數數字)

bool (True / False，分別為 1 / 0)

- 浮點數

float (雙精度浮點數)

complex (雙精度浮點數表示複數，複數 z 的實部和虛部用 $z.real$ 和 $z.imag$ 取得)

- 字串

str (字串，可以用index選取字元)

資料容器

- List (列表)

[1, 2, 3, 4, 5, 5]

- Tuple (元組，有序不可更改)

(1, 2, 3, 4, 5, 5)

- Set (集合，無序不重複)

{ 1, 2, 3, 4, 5 }

資料容器

- Dict (字典，無序)

```
{  
    'name': '柴魚',  
    'age': 25,  
    'man': True,  
    'birth': [1994, 7]  
}
```

數值運算

+ - * /

加減乘除

次方
23(2的3次方)**

//

除法只留整數
(不會四捨五入)

%

取餘數

字串

- `S = 'TEST' 、 "TEST" 、 '''TEST'''`
- `print(S[1])` → `E`
- `print(S + "OK")` → `TESTOK`
- `print(S[0:3])` → `TES`
- `print(S[0:3:2])` → `TS`
- `print(S[-2:])` → `ST`

條件分支

比較運算符

- `<`、`>`、`<=`、`>=`、`==`、`!=`

邏輯運算符

- `and`、`or`、`not`

條件運算符

- 成立 `if` 條件式 `else` 不成立
x = True `if` a=0 `else` False

條件式

`if` 布林條件式:

...

`elif` 布林條件式:

...

`else`:

...

Function

- 用 `def` 定義函式
- 參數可以給定預值、不定參數值
- 不限返回型態
- 可以返回多值

```
def MinMax(n1, n2=99):
```

```
...
```

```
    return min, max
```

```
def Sum(*n3):
```

```
...
```

```
    return total
```

```
A, B = MinMax(5)
```

```
C = Sum(1, 2, 3, 4, 5)
```

小練習

- 嘗試寫一個擷取部分字串的函式
- 輸入參數
(原字串, 開始pattern, 結尾pattern)
- 擷取開始與結尾中間的文字輸出

```
hw = "test<p>Hello World</p>end"
```

```
def mid (source, start, end):  
    r = " "  
    ...  
    return r
```

```
if __name__ == "__main__":  
    print(mid(hw, "<p>", "</p>"))
```

Flask



介紹&安裝

- 內建輕量級web server、Jinja2模板引擎
- 很多插件來實現具體功能
- `pip install flask` (`pip3 install flask`)
- 練習從 PyCharm 設定中管理套件

Hello World

```
from flask import Flask  
app = Flask(__name__)
```

```
@app.route('/hello')  
def hello_world():  
    return 'Hello World!'
```

```
if __name__ == '__main__':  
    app.run(host='0.0.0.0', port=8080)
```

引入 flask 包
實例一個網頁應用 app

定義路由規則

處理請求函式

啟動 Web server

有參數的route

```
@app.route('/hello/<id>')  
def hello_world(id):  
    return 'Hello %s' % id
```

試試看

```
@app.route('/hello/<int:id>')  
@app.route('/hello', methods=['GET', 'POST'])
```

- 型別轉換器支援
 - (空) ，字串但裡面不能有 /
 - int ，整數
 - float ，浮點數
 - path ，字串 可以有 /

有參數的route

① 127.0.0.1:8080/get_name?your_name=Egoist

```
@app.route( '/get_name' , methods=[ 'GET' ])  
def get_name():  
    name = request.args.get( "your_name" )  
    return 'Hello %s' % name
```

TODO request.args 裡面會保存 GET 傳進來的參數 可以用 request.args.get('名稱') 取值 print(request.args)

有參數的route

```
<form action="post_login" method="POST">
    姓名:<input name="your_name" type='text'><br><br>
    密碼:<input name="your_password" type='password'><br>
    <p><input type='submit' value='送出表單'></p>
</form>
```

```
# 取得 POST 表單
@app.route('/post_login', methods=['POST'])
def post_hello():
    # TODO request.form 會保存 POST 表單    print(request.form)

    name = request.form.get("your_name")
    password = request.form.get("your_password")

    return "<h1> Hello %s  <br> <p>your password = %s </p> </h1>" % (name, password)
```

Jinja 2



Jinja 語法

- 印出變數
 - {{傳過來的參數}}
- 判斷式
 - {% - if foo -%}...{% endif %}
 - {% if 參數 is 字串 %}
 - {% if 參數 is in ... %}
- 迴圈
 - {% for i in ...%}你的程式{% end for %}
- 邏輯運算
 - python 支援的運算幾乎都支援
- 設定變數
 - {% set 變數 = 你要設定的值 %}