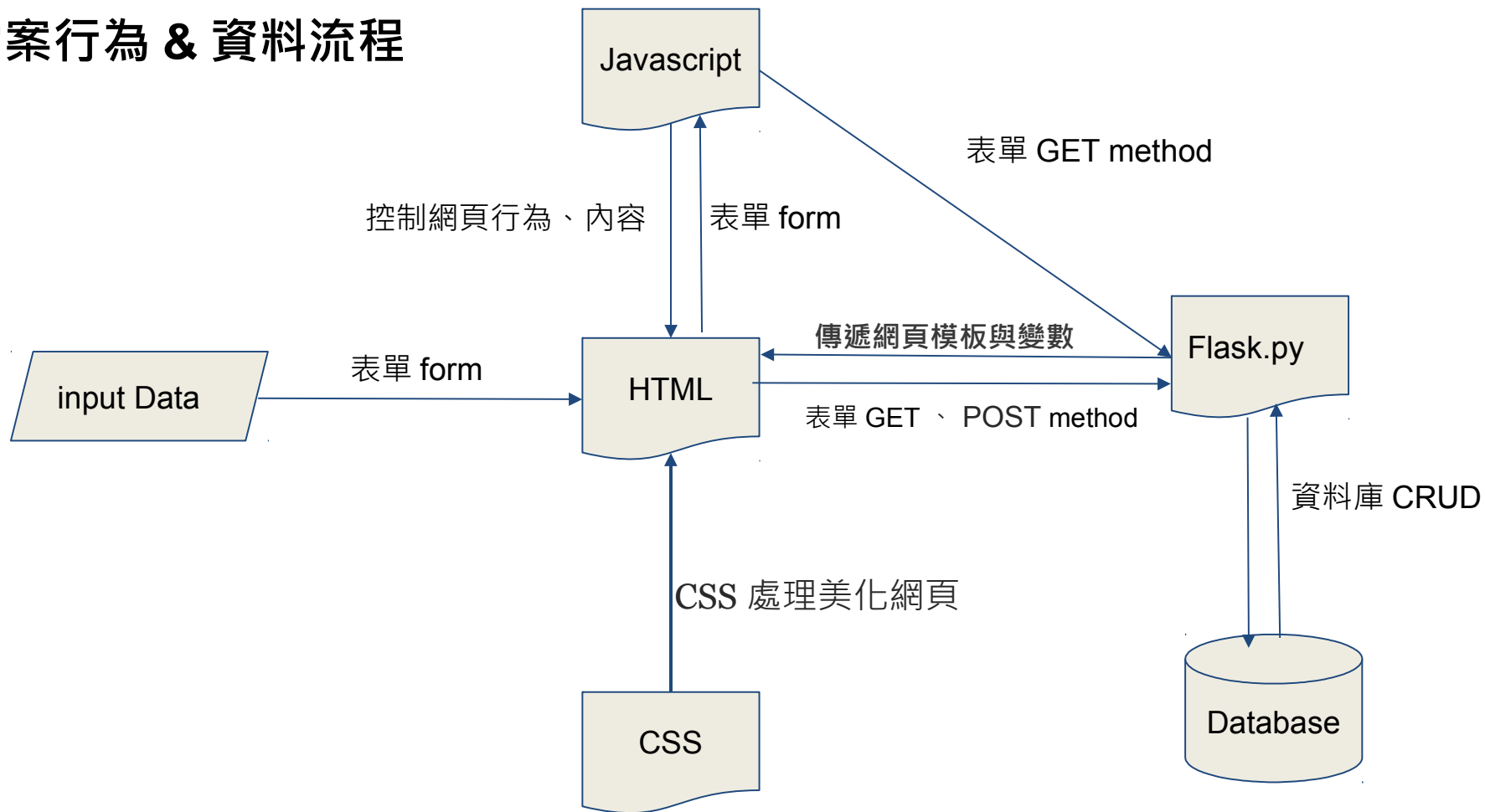


Python Flask 應用

檔案行為 & 資料流程圖



HTML、CSS、JavaScript 介紹

HTML: 網頁的內容與架構

CSS: 網頁的視覺與風格

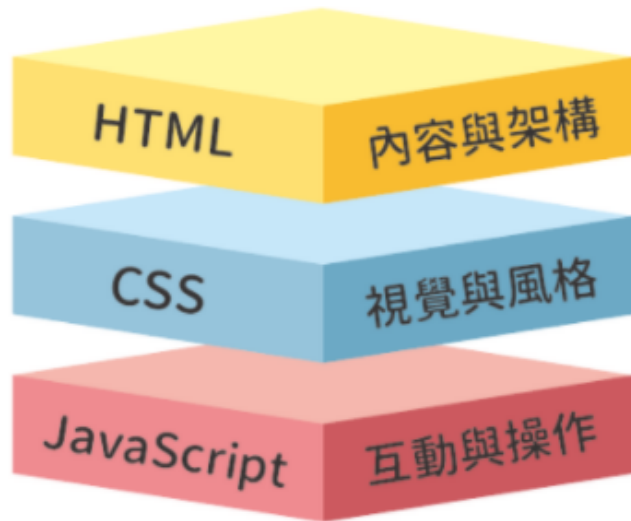
JavaScript: 網頁的互動與操作

教學文章與範例練習：

[網頁三兄弟：HTML、CSS、JavaScript](#)

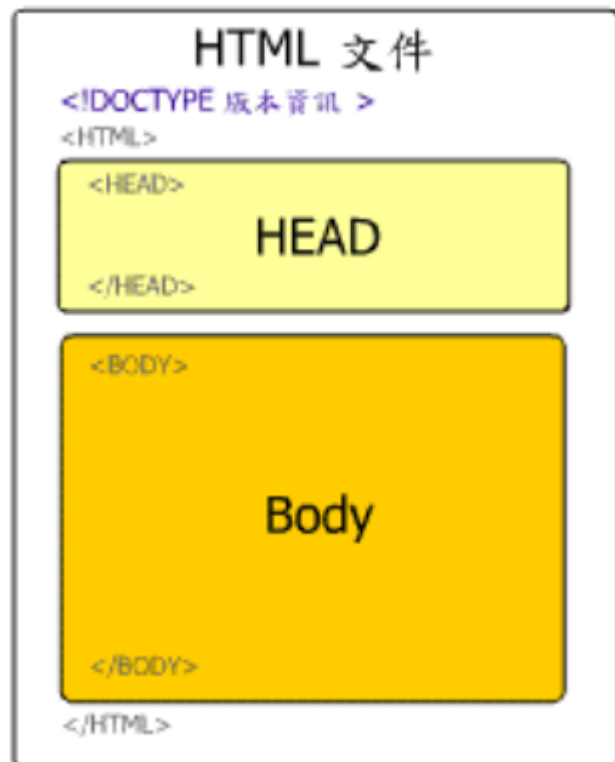
[W3C 表單form input輸入資料範例網址](#)

[W3C JavaScript範例練習網址](#)



[W3C CSS範例練習網址](#)

HTML 介紹說明基本架構：



```
1 <!DOCTYPE html>
2
3 <HTML>
4
5 <HEAD><TITLE>瀏覽器標題欄</TITLE></HEAD>
6
7 <BODY>
8   文件內容
9 </BODY>
10
11 </HTML>
```

HTML 教學文章與範例練習：

[HTML筆記文章](#)

[15個常用的HTML標籤與屬性基本介紹](#)

[W3C HTML範例練習網址](#)

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

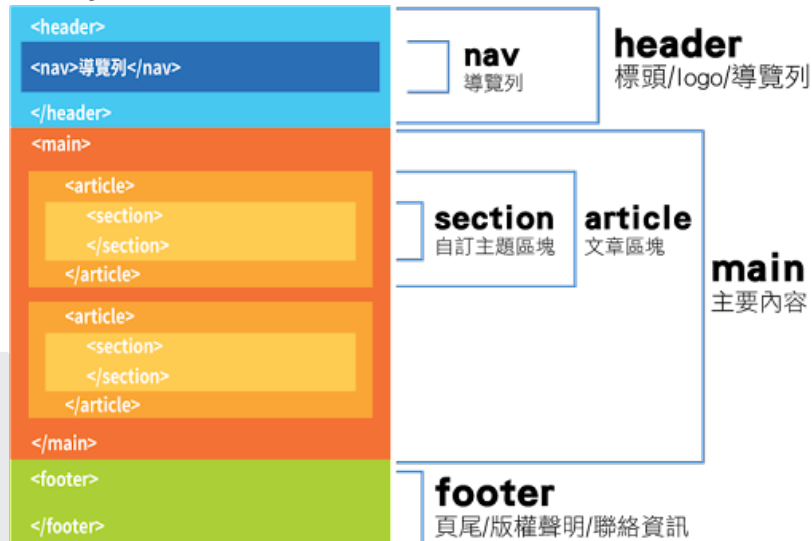
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

[Try it Yourself »](#)

html5架構

body



Flask 簡介

Flask 是一個使用 Python 撰寫的輕量級 Web 應用程式框架，由於其輕量特性，也稱為 micro-framework（微框架）。Micro 輕量化的設計理念，在架設 Flask 時就像是堆積木，可以自己決定要使用什麼積木（擴充套件），不會有多餘的積木，達到簡單、輕量、高擴充性的架構，Flask 給予開發者非常大的彈性（當然你也可以說是需要思考更多事情）。

前置作業 :python 環境的安裝 (WIN10 版)

關於環境安裝的文章：

[1 python安裝](#)

[2-0 vs code程式編輯器安裝](#)

[2-1 vs code python延伸套件](#)

[2-2 環境變數設,中文亂碼設定](#)

撰寫你的第一個 Flask 網站

印出 Hello World!

1 先打開 CMD 開始安裝基本套件 `pip install flask`，然後開始撰寫你的第一個 Flask 網站。

先 `pip install flask`

```
C:\ironman>pip install flask
Collecting flask
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec85276964ff8f43bbb1d3b/Flask-1.1.1-py2.py3-none-any.whl (94kB)
    100% |#####| 102kB 793kB/s
Collecting Jinja2>=2.10.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/1d/e7/fd8b501e7a6dfe492a433deb7b9d833439ca74916fa8bc63d41a4947a671/Jinja2-2.10.1-py2.py3-none-any.whl (124kB)
    100% |#####| 133kB 1.3MB/s
Collecting itsdangerous>=0.24 (from flask)
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b25366fde317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting Werkzeug>=0.15 (from flask)
  Downloading https://files.pythonhosted.org/packages/d1/ab/43bce46b92042672824decc7aad68877badf18aec1571045840ad495643f/Werkzeug-0.15.5-py2.py3-none-any.whl (328kB)
    100% |#####| 337kB 3.1MB/s
Collecting click>=5.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abb630d7257104c434fe0b07e5a195a6847506c074527aa599ec/Click-7.0-py2.py3-none-any.whl (81kB)
    100% |#####| 81kB 446kB/s
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10.1->flask)
  Downloading https://files.pythonhosted.org/packages/5b/44/1deb3c5dc3714fb160c7e21166c6dfff36a063d9156a9328cce54ef35cc52/MarkupSafe-1.1.1-cp37-cp37a-win32.whl
Installing collected packages: MarkupSafe, Jinja2, itsdangerous, Werkzeug, click, flask
Successfully installed Jinja2-2.10.1 MarkupSafe-1.1.1 Werkzeug-0.15.5 click-7.0 flask-1.1.1 itsdangerous-1.1.0
```

2 新增 py 檔，輸入代碼如下

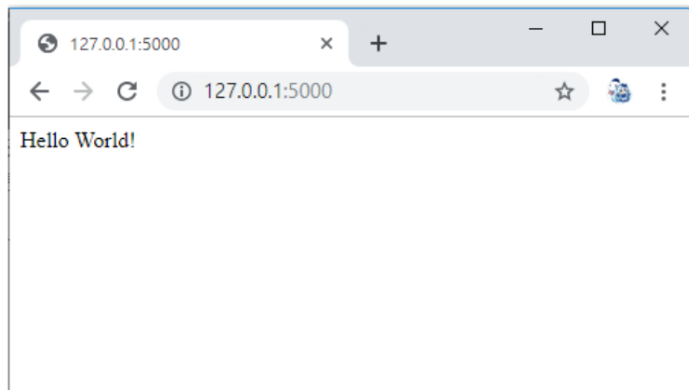
```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def test():
    return "Hello World!"

if __name__ == "__main__":
    app.run(debug=True)
```

印出 Hello World!



3 執行程式，點擊網址列，或是在瀏覽器網址列輸入 <http://127.0.0.1:5000/>

```
D:\ironman>python flask_test_1.py
* Serving Flask app "flask_test_1" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 128-853-007
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```


Flask 參考連結

1 Flask 安裝

[iT幫幫忙flask安裝教學](#)

2 Flask 前端與後端變數的傳遞-GET、POST 傳送資料

[medium 參考文章](#)

3 Flask 前端檔案HTML/CSS/JS 引用

[HTML CSS JS的引用](#)

Python 後端與 MySQL 資料庫連線 CRUD 應用

前置作業 . 關於資料庫安裝的文

章 :

[MySQL、HeidiSQL 下載及安裝教學](#)

Python MySQL 連線與 CRUD 語法文章

[MySQL+Python連接和操作](#)

1 先打開 CMD 開始安裝 MySQL 連線套件 `pip install pymysql`

SQL 語法 : [CRUD基本語法](#)

查詢資料語法 : `SELECT * FROM order_header WHERE order_status = '完工'`

新增資料語法

`INSERT INTO order_header (order_no, order_type, order_status, order_date, createdon) VALUES ('2018102109', 2, '下達', sysdate(), sysdate())`

更新資料語法

`UPDATE order_header SET order_no = '112109' WHERE order_no = '2018102109'`

刪除資料語法

`DELETE FROM order_header WHERE order_no = '112109'`

Python 後端套用 io_database.py 檔案與 MySQL 資料庫連線 CRUD 應用

製作連線物件：

```
1  from io_database import database
2
3  host = "127.0.0.1"           #主機名稱 (MYSQL_HOST)
4  user = "root"                #用戶名稱 (MYSQL_USER)
5  password = "123456"          #資料庫密碼(MYSQL_PASS)
6  PHP_database = "affiliates"  #資料庫名稱(MYSQL_DATABASE)
7  port = 3307                  #連接埠號(MYSQL_PORT)
8  #輸入連線參數,用database類別作個連線物件|
9  objIO_DB_PHP = database (host,user,password,PHP_database,port )
10 print (objIO_DB_PHP )
```

io_database.py CRUD SQL 語法

讀取資料表，欄位資料：

```
sqlstring = f'SELECT {star} FROM {tableName}'
```

新增資料：

```
sqlstring = f'UPDATE {tableName} set {KeyValuestring}  where id = {id}'
```

更新資料：

```
sqlstring = f'UPDATE {tableName} set {KeyValuestring}  where id = {id}'
```

刪除資料表所有資料：

```
sqlstring = f'delete FROM {tableName}'
```

清空資料表：

```
sqlstring = f'TRUNCATE TABLE {tableName}'
```

運用物件新增資料：

```
#新增資料到資料庫
#要新增資料的字典(key:value) 資料對映(欄位:資料)
dictCustomer = {
    "aid": "001",
    "user_id": "111",
    "username": "customer333",
    "add_time": "2021-08-05 00:00:00",
    "created_at": "2021-08-05 00:00:00",
    "updated_at": "2021-08-05 00:00:00"}
#藉由 資料表名稱 與 字典(key:value) 資料對映(欄位:資料) 新增一筆資料
insertData = objIO_DB_PHP.insertData(tableName,dictCustomer)
print(insertData)
#新增資料後再讀取資料庫新增資料狀況
readDataDF = objIO_DB_PHP.readData(tableName,columnList)
print (readDataDF)
```

運用物件更新資料

```
37  #更新資料庫
38  #要更新資料的字典參數,更新位置為 id = 1 的那一筆資料
39  dictCustomer = {
40      "id":"1",
41      "user_id":"111",
42      "username":"customer888",
43      "add_time":"2021-08-05 00:00:00",
44      "created_at":"2021-08-05 00:00:00",
45      "updated_at":"2021-08-05 00:00:00"
46  }
47  #藉由 資料表名稱 與 字典(key:value) 資料對映(欄位:資料) 更新一筆資料
48  updateData = objIO_DB_PHP.updateData(tableName,dictCustomer)
49  #更新資料後再讀取資料庫新增資料狀況
50  readDataDF = objIO_DB_PHP.readData(tableName,columnList)
51  print (readDataDF)
```

運用物件讀取、刪除、清空資料表：[Python_MySQL連線工具檔案下載](#)

```
12  #讀取資料庫
13  #要讀取的資料表名稱
14  tableName='aff_customer'
15  #要讀取的欄位名稱,'*'表示讀取全部欄位
16  columnList=['*']
17  #將返回的DataFrame存入readDataDF變數
18  readDataDF = objIO_DB_PHP.readData(tableName,columnList)
19  print (readDataDF)
```

```
55  #刪除資料表所有資料
56  #刪除資料表所有資料,主鍵ID AUTO_INCREMENT序號記憶不會刪除,if刪除前ID序號是11再新增資料ID會是12
57  deleteData = objIO_DB_PHP.deleteData(tableName)
58  #刪除資料表所有資料後再讀取資料庫
59  readDataDF = objIO_DB_PHP.readData(tableName,columnList)
60  print (readDataDF)
```

```
62  #清空資料表
63  truncateTable = objIO_DB_PHP.truncateTable(tableName)
64  #清空資料表後再讀取資料庫
65  readDataDF = objIO_DB_PHP.readData(tableName,columnList)
66  print (readDataDF)
```