第9章 WiFi上網：urequests物件+JSON處理(Open Data)

- 9-1 連接WiFi基地台
- 9-2 認識HTTP請求
- 9-3 使用urequests送出HTTP請求
- 9-4 取得和剖析JSON資料
- 9-5 整合應用：Google圖書查詢的Web API
- 9-6 整合應用：OpenWeatherMap天氣資訊指示燈

9-1 連接WiFi基地台

- 在ESP8266已經整合WiFi網路晶片，可以使用三種工作模式來連線WiFi，如下所示：
  - STA模式（Station）：ESP8266開發板如同一張WiFi無線網路卡，可以連線至可用的WiFi基地台。
  - AP模式（Access Point）：將ESP8266開發板作為熱點的WiFi基地台，可以讓其他裝置連線至 ESP8266開發板，例如：智慧型手機。
  - STA+AP模式：同時啟用STA與AP模式的混和模式。
- MicroPython 網路功能是network模組，使用ESP8266開發板連接WiFi基地台需要匯入network模組，如下所示：

```
import network
```

9-1 連接WiFi基地台
啟用WiFi和掃瞄WiFi基地台：ch9-1.py

- MicroPython程式需要先啟用WiFi的STA模式後，才能掃瞄找出可連線的WiFI基地台清單，如下所示：

```
import network

sta = network.WLAN(network.STA_IF)
sta.active(True)
print(sta.isconnected())
print(sta.scan())
```

```
False
[(b'ESSID_MyNetwork', b'\\\x92^\x1bz\x9c', 11, -69, 3, 0), (b'A
SUS-home', b',\xfd\xa1`z\xfc', 2, -81, 3, 0), (b'KI LIN', b'\x9
8\rg\xa5\xbdO', 11, -89, 3, 0)]
```

9-1 連接WiFi基地台

顯示可連線WiFi基地台的MAC地址：ch9-1a.py

- 我們可以使用ubinascii模組將二進位值的MAC地址轉換成十六進位值的字元，如下所示：

```
import network
import ubinascii

sta = network.WLAN(network.STA_IF)
sta.active(True)
print(sta.isconnected())
aps = sta.scan()
for ap in aps:
    ssid = ap[0].decode()
    mac = ubinascii.hexlify(ap[1], ":").decode()
    print(ssid, mac)
```

```
False
ASUS-home 2c:fd:a1:60:7a:fc
ESSID_MyNetwork 5c:92:5e:1b:7a:9c
KI LIN 98:0d:67:a5:bd:4f
```

9-1 連接WiFi基地台

在成功連線WiFi基地台後馬上中斷連線：ch9-1b.py

- 在了解如何掃瞄基地台後，我們就可以連線WiFi基地台後，馬上中斷WiFi連線，如下所示：

```
import network

sta = network.WLAN(network.STA_IF)
sta.active(True)
if not sta.isconnected():
    print("Connecting to network...")
    sta.connect('<WiFi名稱>', '<WiFi密碼>')
    while not sta.isconnected():
        pass
print("network config:", sta.ifconfig())
sta.disconnect()
print(sta.isconnected())
```

9-1 連接WiFi基地台

建立函式來連線WiFi基地台：ch9-1c.py

- 為了方便MicroPython程式連線WiFi，我們可以建立connect_wifi()函式來連線WiFi，函式只需傳入SSID名稱和連線密碼，就可以進行WiFi連線，如下所示：

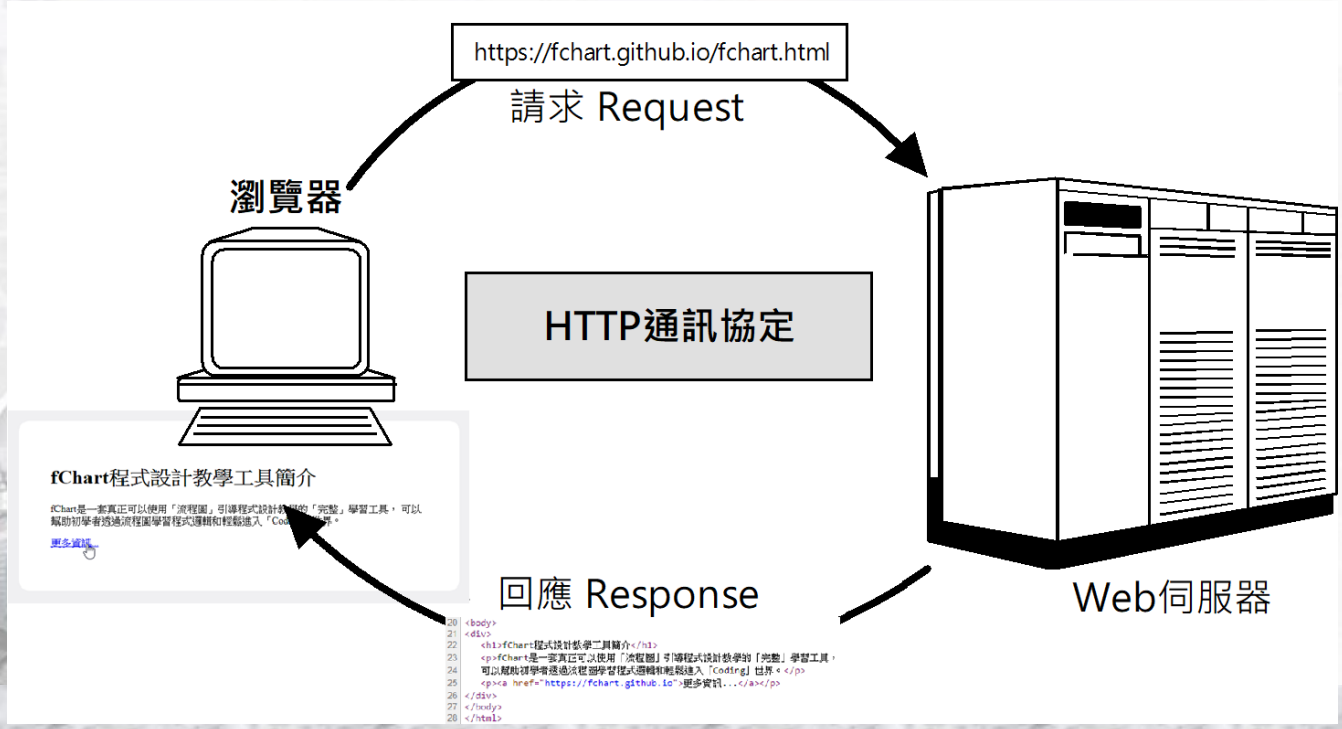
```
...
def connect_wifi(ssid, passwd):
    sta = network.WLAN(network.STA_IF)
    sta.active(True)
    if not sta.isconnected():
        print("Connecting to network...")
        sta.connect(ssid, passwd)
        while not sta.isconnected():
            pass
    print("network config:", sta.ifconfig())
```

9-2 認識HTTP請求

- HTTP請求是使用HTTP通訊協定送出請求，主要有兩種方法，如下所示：
  - GET方法：在瀏覽器輸入URL網址送出的請求就是GET方法的HTTP請求，這是向Web伺服器要求資源的HTTP請求。
  - POST方法：在瀏覽器顯示的HTML表單輸入欄位資料後，按下按鈕送出欄位資料，就是使用POST方法的HTTP請求，可以將欄位輸入資料送至Web伺服器。

9-2 認識HTTP請求
HTTP通訊協定

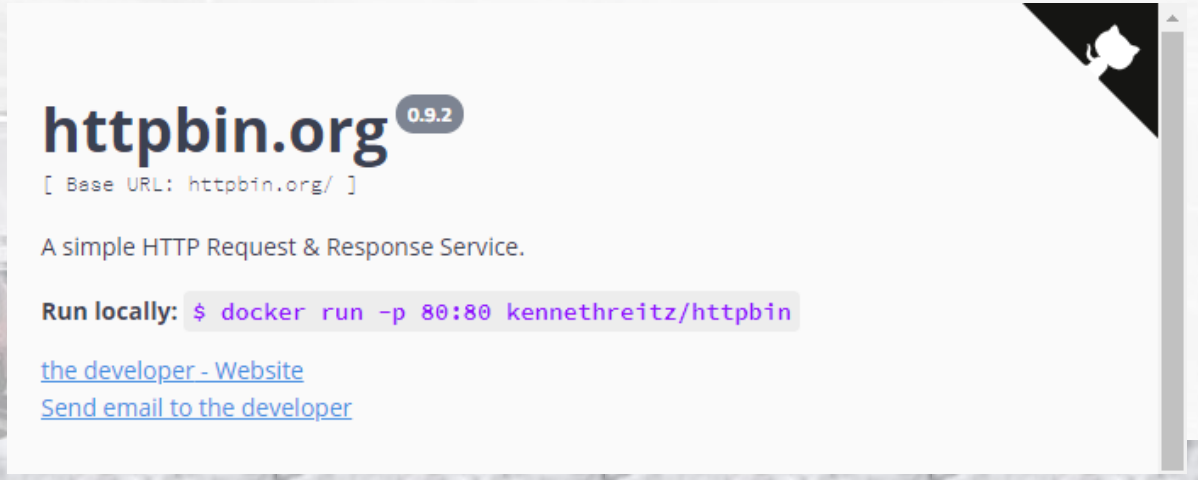
- 瀏覽器是使用「HTTP通訊協定」（Hypertext Transfer Protocol）送出HTTP的GET請求（目標是URL網址的網站），可以向Web伺服器請求所需的HTML網頁資源，如下圖所示：



9-2 認識HTTP請求
使用httpbin.org網站測試HTTP請求

- 為了方便測試HTTP請求和回應，我們可以使用httpbin.org服務來進行測試。在httpbin.org網站提供HTTP請求/回應的測試服務，類似Echo服務，可以將我們送出的HTTP請求，自動使用JSON格式回應送出的HTTP請求資料，支援GET和POST方法等多種方法，其URL網址如下所示：

  http://httpbin.org



9-3 使用urequests送出HTTP請求

- MicroPython實作requests模組的子集，稱為urequests模組，可以讓我們建立MicroPython程式來送出HTTP請求。首先需要匯入模組（需連線WiFi），如下所示：

```
import urequests
```

9-3 使用urequests送出HTTP請求

urequests模組送出GET請求：ch9-3.py

- 在MicroPython送出GET請求是呼叫urequests模組的get()方法，參數是URL網址，如下所示：

```
…
import urequests

r = urequests.get("http://httpbin.org/get")
print(r.status_code)
print(r.text)
```

```
200
{
  "args": {},
  "headers": {
    "Host": "httpbin.org",
    "X-Amzn-Trace-Id": "Root=1-605064df-57737c54309673ef770f24af"
  },
  "origin": "36.226.39.16",
  "url": "http://httpbin.org/get"
}
```

9-3 使用urequests送出HTTP請求

urequests模組送出GET請求取得HTML資料：ch9-3a.py

- 如果是Web網站，MicroPython程式可以使用urequests模組送出GET請求來取得HTML資料，如下所示：

```
…
import urequests

r = urequests.get("https://fchart.github.io/fchart.html")
if r.status_code == 200:
    print(r.encoding)
    print(r.text)
```

```
utf-8
<!doctype html>
<html>
<head>
    <title>fChart程式設計教學工具簡介</title>
    <meta charset="utf-8" />
    <meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
    <style type="text/css">
    body {
        background-color: #f0f0f2;
    }
    div {
```

9-3 使用urequests送出HTTP請求
在GET請求使用URL參數：ch9-3b.py

- 在get()方法參數的URL網址最後可以加上URL參數，這是位在「?」問號之後的成對URL參數值，如下所示：

  http://httpbin.org/get?a=15

- 上述超連結有1個名為a的URL參數，其值為15。如果參數不只一個，請使用「&」符號分隔，如下所示：

  http://httpbin.org/get?a=15&b=22

- 上述URL網址傳遞參數a和b，其值分別是「=」等號後的15和22。

9-3 使用urequests送出HTTP請求
在GET請求使用URL參數：ch9-3b.py

- MicroPython程式和ch9-3a.py相同，只有URL網址字串不同，如下所示：

```
…
import urequests

r = urequests.get("http://httpbin.org/get?a=15&b=22")
if r.status_code == 200:
    print(r.encoding)
    print(r.text)
```

```
utf-8
{
  "args": {
    "a": "15",
    "b": "22"
  },
  "headers": {
    "Host": "httpbin.org",
    "X-Amzn-Trace-Id": "Root=1-605065d5-03b9f87b0c98fc6a3afbcbc6"
  },
  "origin": "36.226.39.16",
  "url": "http://httpbin.org/get?a=15&b=22"
}
```

9-3 使用urequests送出HTTP請求

urequests模組送出POST請求：ch9-3c.py

- 當使用者在HTML表單欄位輸入值且按鈕送出後，在瀏覽器是建立POST請求送至Web伺服器，其傳遞資料包含使用者輸入的表單欄位值。MicroPython是使用urequests模組的post()方法來送出HTTP POST請求，如下所示：

```
…
import urequests

data = '{ "a":15, "b":22 }'  # JSON資料的字串
r = urequests.post("http://httpbin.org/post", data=data)
if r.status_code == 200:
    print(r.encoding)
    print(r.text)
```

9-3 使用urequests送出HTTP請求

urequests模組送出POST請求：ch9-3c.py

```
utf-8
{
  "args": {},
  "data": "{ \"a\":15, \"b\":22  }",
  "files": {},
  "form": {},
  "headers": {
    "Content-Length": "19",
    "Host": "httpbin.org",
    "X-Amzn-Trace-Id": "Root=1-6050663d-7e426fb47e22bcf454fc3efb"
  },
  "json": {
    "a": 15,
    "b": 22
  },
  "origin": "36.226.39.16",
  "url": "http://httpbin.org/post"
}
```

9-3 使用urequests送出HTTP請求
自訂HTTP請求的標頭資訊：ch9-3d.py

- 我們可以使用urequests模組的get()方法建立自訂HTTP標頭的HTTP請求，如下所示：

```
...
import urequests
headers={'content-type': 'application/json'}
r = urequests.get("http://httpbin.org/get",
headers=headers)
if r.status_code == 200:
    j = r.json()
    print(j)
```

```
{'url': 'http://httpbin.org/get', 'headers': {'X-Amzn-Trace-Id': 'Root=1-60506ee5
-5657184347dfcbf132c50a4b', 'Host': 'httpbin.org', 'Content-Type': 'application/j
son'}, 'args': {}, 'origin': '36.226.39.16'}
```

9-4 取得和剖析JSON資料

- 9-4-1 認識JSON資料
- 9-4-2 JSON資料處理和剖析
- 9-4-3 Google Chrome的RestMan擴充功能
- 9-4-4 剖析網路取得的JSON資料

9-4-1 認識JSON資料

- JSON（JavaScript Object Notation）是由Douglas Crockford創造的一種輕量化資料交換格式，使用大括號定義成對的鍵和值（Key-value Pairs），相當於物件的屬性和屬性值（Python剖析JSON物件是轉換成字典），如下所示：

```
{
   "key1": "value1",
   "key2": "value2",
   "key3": "value3",
   ...
}
```

9-4-1 認識JSON資料

- JSON物件陣列是使用方括號來定義（Python剖析JSON陣列是轉換成串列），如下所示：

```
[
    {
    "title": "ASP.NET網頁設計",
    "author": "陳會安",
    "category": "Web",
    "pubdate": "06/2015",
    "id": "W101"
    },
```

9-4-1 認識JSON資料

```
{
"title": "PHP網頁設計",
"author": "陳會安",
"category": "Web",
"pubdate": "07/2015",
"id": "W102"
},
...
]
```

9-4-2 JSON資料處理和剖析

- MicroPython實作json模組的子集，稱為ujson模組，在MicroPython程式處理JSON資料需要匯入ujson模組，如下所示：

```
import ujson
```

9-4-2 JSON資料處理和剖析
從JSON字串轉換成Python字典：ch9-4-2.py

- 在ujson模組是使用loads()方法從JSON字串轉換成Python字典，如下所示：

```
import ujson

json_str = """{"name":"John"}"""
parsed = ujson.loads(json_str)
print(parsed)
print(type(parsed))
```

```
{'name': 'John'}
<class 'dict'>
```

9-4-2 JSON資料處理和剖析

將Python字典轉換成JSON字串：ch9-4-2a.py

- 在ujson模組是使用dumps()方法將Python字典轉換成JSON字串，如下所示：

```
import ujson

dic = {"device":"temperature","id":543,"values":[1,2,3]}
json_str = ujson.dumps(dic)
print(json_str)
print(type(json_str))
```

```
{"values": [1, 2, 3], "device": "temperature", "id": 543}
<class 'str'>
```

9-4-2 JSON資料處理和剖析

取出JSON剖析結果的資料：ch9-4-2b.py

- 在使用loads()方法從JSON字串轉換成Python字典後，就可以使用字典的鍵來取出剖析結果的資料，如下所示：

```
import ujson

json_str =
    """{"device":"temperature","id":543,"values":[1,2,3]}"""
parsed = ujson.loads(json_str)
print(parsed["device"])
print(parsed["id"])
print(parsed["values"])
print(type(parsed["values"]))
```

```
temperature
543
[1, 2, 3]
<class 'list'>
```

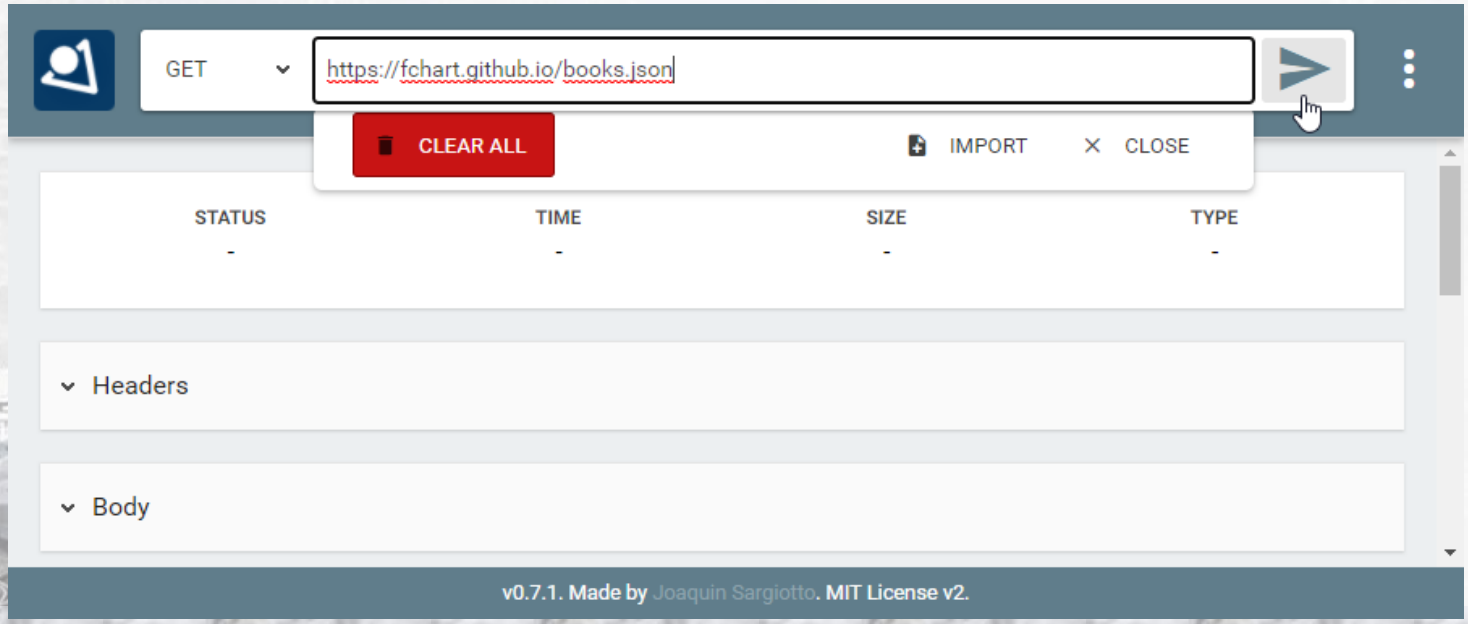
9-4-3 Google Chrome的RestMan擴充功能
安裝RestMan擴充功能

- Google Chrome的RestMan擴充功能提供圖形化介面來送出HTTP請求，可以檢視回應資料和格式化顯示回應的JSON資料。

9-4-3 Google Chrome的RestMan擴充功能
使用RestMan擴充功能

- Step 1：請在Chrome瀏覽器右上方工具列點選RestMan擴充功能圖示，在請求方法欄選GET後，在後方欄位填入Web API的URL存取網址後，按之後箭頭鈕送出HTTP請求，如下所示：

  https://fchart.github.io/books.json



9-4-3 Google Chrome的RestMan擴充功能
使用RestMan擴充功能

- Step 2：在送出HTTP請求取得回應後，請捲動視窗，可以在下方檢視回應的JSON資料，【JSON】標籤是格式化顯示的JSON資料；【HTML PREVIEW】標籤是網頁預覽。

9-4-4 剖析網路取得的JSON資料

使用ujson剖析網路取得的JSON資料：ch9-4-4.py

- 使用urequests模組的get()方法從網路取得JSON資料後，使用ujson模組的loads()方法剖析JSON資料，如下所示：

```
…
import urequests, ujson

r = urequests.get("https://fchart.github.io/books.json")
if r.status_code == 200:
    j = ujson.loads(r.text)
    for book in j:
        print(book["title"])
        print(book["author"])
        print(book["id"])
```

```
ASP.NET網頁程式設計
陳會安
W101
PHP網頁程式設計
陳會安
W102
Java程式設計
陳會安
P102
Android程式設計
陳會安
M102
```

9-4-4 剖析網路取得的JSON資料

直接剖析網路取得的JSON資料：ch9-4-4a.py

- 我們可以不用ujson模組，直接使用r.json()方法來剖析JSON資料，如下所示：

```
…
import urequests


r = urequests.get("https://fchart.github.io/books.json")
if r.status_code == 200:
    j = r.json()
    for book in j:
        print(book["title"])
        print(book["author"])
        print(book["id"])
```

9-5 整合應用：Google圖書查詢的Web API

- Web API是一個網路上的URL存取網址，其使用方式如同在瀏覽器輸入URL網址來瀏覽網頁，目前很多公開API都可以直接在瀏覽器執行請求來取得回應資料，其回應資料大多採用JSON格式。目前的Web API主要可以分成兩種，如下所示：
  - 公開API（Public/Open API）：任何人不需註冊帳號就可以使用的Web API，例如：Google圖書查詢服務。
  - 認證API（Authenticated API）：需要先註冊帳號後才能使用的Web API，例如：OpenWeatherMap天氣資料。

9-5 整合應用：Google圖書查詢的Web API

使用Google Web API查詢圖書資料：ch9-5.py

- MicroPython程式的執行結果可以看到找到的3本圖書資料，如下所示：

```
下載:  7819 字元
---------------------------
圖書名:  Introducing Python
出版商:  O'Reilly Media
出版日:  2019-11-06
---------------------------
圖書名:  Serious Python
出版商:  No Starch Press
出版日:  2018-12-27
---------------------------
圖書名:  Python Crash Course
出版日:  2019-05
---------------------------
```

9-6 整合應用：OpenWeatherMap天氣資訊指示燈

- OpenWeatherMap是一個提供天氣資料的線上服務，可以提供目前的天氣資料、天氣預測和天氣的歷史資料（從1979年至今）。我們準備整合三色LED，當氣溫小於等於18度時，點亮藍色LED；19~25度點亮綠色LED，大於25點亮紅色LED。
  - 註冊OpenWeatherMap帳號
  - 使用OpenWeatherMap的Web API
  - 取得OpenWeatherMap天氣資料：ch9-6.py

9-6 整合應用：OpenWeatherMap天氣資訊指示燈-取得OpenWeatherMap天氣資料：ch9-6.py

- MicroPython程式的執行結果可以顯示台北目前的天氣資料，指示燈顯示GREEN綠色，如下所示：

```
----------------------------
天氣描述:  晴, 少雲
----------------------------
目前溫度:  24.24
大氣壓力:  1014
目前溼度:  73
最低溫度:  23.89
最高溫度:  25
----------------------------
GREEN
```