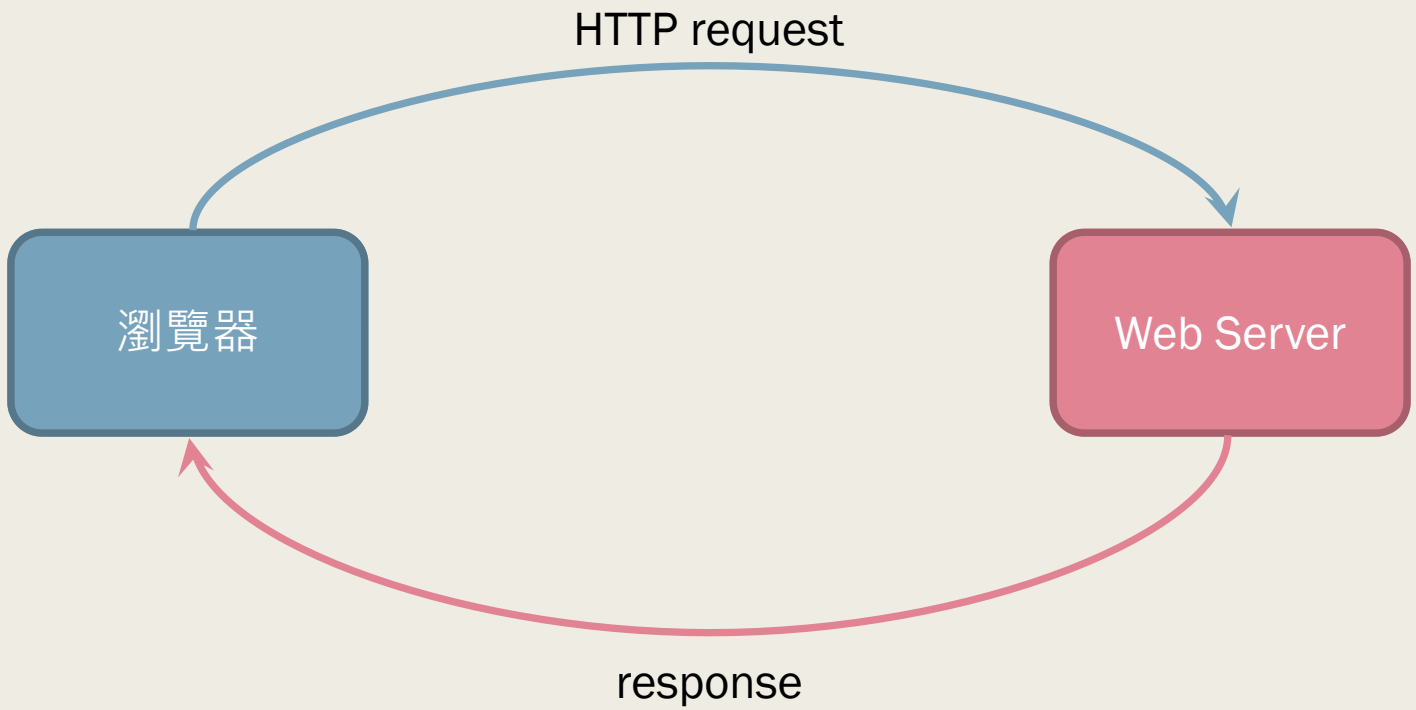
HTTP 協定

朱克剛

WEB運作原理



網頁組成

- HTML標籤
- JavaScript程式
- CSS樣式

```
<html>
<head>
    <title>我的網站</title>
</head>
<script language="javascript">
function blue_color(){
   document.getElementById("main").style ="color: blue";
}
</script>
<body>
    <span id="main" style="color: red">Hello World</span>
    <p/>
    <input type="button" value="藍色" onclick="blue_color()"/>
</body>
</html>
```

網頁中包含圖片

- 1. 瀏覽器發出 HTTP request
- 2. Web Server 傳回 HTML 網頁
- 3. 瀏覽器發出 a.jpg 的 HTTP request
- 4. Web Server 傳回 a.jpg
- 5. 瀏覽器發出 b.jpg 的 HTTP request
- 6. Web Server 傳回 b.jpg

```
<html>
<body>
    <img src="a.jpg"/>
    <img src="b.jpg"/>
</body>
</html>
```

GET

- 將資料放在網址列傳給 web server
- 例如：
  - *Google 地圖*
    - https://www.google.com.tw/maps/place/台北101大樓
  - *譯點通*
    - https://yun.dreye.com/dict_new/dict.php?w=python
- 特性
  - *網址列資料無法加密，有資安問題*
  - *網址會被 bookmark、存入歷史清單或被cache*
  - *有長度限制*

POST

- 將資料放在body中傳給web server

```
POST /data/index.php HTTP/1.1
Host: foo.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

say=Hi&to=Mom
```

- 特性：
  - *資料可以加密傳輸*
  - *無法bookmark、無法進入歷史清單、永遠不會cache*
  - *無長度限制*

以GET方式送出 – Python2

```
import urllib2 as urllib

response = urllib.urlopen('http://1.2.3.4/add.cgi?a=5&b=3')
print (response.read())
```

以GET方式送出 – Python3

```
import urllib.request as urllib

response = urllib.urlopen('http://1.2.3.4/add.cgi?a=5&b=3')
print (response.read().decode("utf-8"))
```

以POST方式送出 – Python2

```
import urllib2 as urllib

data = "a=5&b=3"
request = urllib.Request('http://1.2.3.4/add.cgi', data)
response = urllib.urlopen(request)
print (response.read())
```

以POST方式送出 – Python3

```
import urllib.request as urllib

data = "a=5&b=3"
request = urllib.Request('http://1.2.3.4/add.cgi', data)
response = urllib.urlopen(request)
print (response.read().decode("utf-8"))
```

取得照片並存檔 – Python 2

```
import urllib2 as urllib

response = urllib.urlopen("http://.../a.jpg")

with open("file.jpg", "wb") as f:
    f.write(response.read())
```

取得照片並存檔 – Python 3

```
import urllib.request as urllib

response = urllib.urlopen("http://.../a.jpg")

with open("file.jpg", "wb") as f:
    f.write(response.read())
```

JSON解析

JSON – Array 格式

```
[
    {
        "name": "王大明",
        "age":40
    },
    {
        "name": "李大媽",
        "age":36
    }
]
```

JSON – Dictionary 格式

```
{
  "id": "A01",
  "name": "王大明",
  "age":40,
  "height":175,
  "weight":50,
  "birthday":317555382
}
```

第1步：匯入library

```
import json
```

第 2 步：Array格式解析

```
import json
text = open("t.json").read()

list = json.loads(text)
for p in list:
   print p["name"], p["age"]
```

```
>> 王大明 40
>> 李大媽 36
```

```
[
    {
        "name": "王大明",
        "age":40
    },
    {
        "name": "李大媽",
        "age":36
    }
]
```

第 2 步：Dictionary格式解析

```
{
  "id": "A01",
  "name": "王大明",
  "age":40,
  "height":175,
  "weight":50,
  "birthday":317555382
}
```

```
import json, time
text = open("t.json").read()

dict = json.loads(text)
print dict["name"]

epoch = dict["birthday"]
print time.strftime("%d %b %Y", time.localtime(epoch))
```

```
>> 王大明
>> 24 Jan 1980
```

XML解析

XML解析

```
<?xml version="1.0"?>
<data>
    <country name="Liechtenstein">
        <rank>1</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

第1步：匯入library

```
import xml.etree.ElementTree as ET
```

第2步：從檔案來解析

```
import xml.etree.ElementTree as ET

tree = ET.parse('demo.xml')
root = tree.getroot()

print root.tag
print root.attrib


>> data
>> {}
```

第2步：從字串來解析

```
xml_string ="....."
root = ET.fromstring(xml_string)
```

第1招：取得子標籤

```
for child in root:
    print child.tag, child.attrib
```

```
>> country {'name': 'Liechtenstein'}
>> country {'name': 'Singapore'}
>> country {'name': 'Panama'}
```

```
<?xml version="1.0"?>
<data>
    <country name="Liechtenstein">
        <rank>1</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

第2招：運用陣列

```
print root[1][1].text
```

```
>> 2011
```

```
<?xml version="1.0"?>
<data>
    <country name="Liechtenstein">
        <rank>1</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

第3招：取得特定標籤

```
for neighbor in root.iter('neighbor'):
    print neighbor.attrib
```

```
>> {'direction': 'E', 'name': 'Austria'}
>> {'direction': 'W', 'name': 'Switzerland'}
>> {'direction': 'N', 'name': 'Malaysia'}
>> {'direction': 'W', 'name': 'Costa Rica'}
>> {'direction': 'E', 'name': 'Colombia'}
```

```
<?xml version="1.0"?>
<data>
    <country name="Liechtenstein">
        <rank>1</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

第4招：取得第一層內容

```xml
<?xml version="1.0"?>
<data>
    <country name="Liechtenstein">
        <rank>1</rank>
        <year>2008</year>
        <gdppc>141100</gdppc>
        <neighbor name="Austria" direction="E"/>
        <neighbor name="Switzerland" direction="W"/>
    </country>
    <country name="Singapore">
        <rank>4</rank>
        <year>2011</year>
        <gdppc>59900</gdppc>
        <neighbor name="Malaysia" direction="N"/>
    </country>
    <country name="Panama">
        <rank>68</rank>
        <year>2011</year>
        <gdppc>13600</gdppc>
        <neighbor name="Costa Rica" direction="W"/>
        <neighbor name="Colombia" direction="E"/>
    </country>
</data>
```

```python
for country in root.findall('country'):
    rank = country.find('rank').text
    name = country.get('name')
    print name, rank
```

```
>> Liechtenstein 1
>> Singapore 4
>> Panama 68
```