

# D3：實作：Python 下載XML檔案與解析



簡報閱讀



範例與作業



問題討論

**Python 下載XML檔案與解析** >

本日知識點目標 >

XML 檔案格式 >

XML 檔案格式優點與缺點 >

思考流程與使用套件 >

## Python 下載XML檔案與解析



具

以這個 xml 檔案為例 &gt;

一個簡單的範例 -  
xml.dom >一個簡單的範例 -  
xml.etree >

## Day 03 檔案存取與資料解析

## Python 下載XML檔案與解析



出題教練：張維元



x

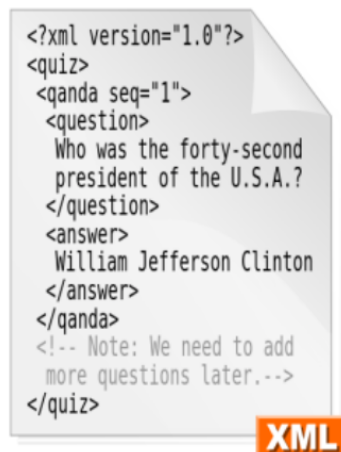
## 本日知識點目標



## 本日知識點目標

- 了解 xml 檔案格式與內容
- 能夠利用套件存取 xml 格式的檔案

XML (eXtensible Markup Language) 可延伸標記式語言，是一種標記式語言，處理包含各種資訊的資料等。



```
<同學資料>
  <同學 no="1" name="王小明" birth="1984/3/11">
    <家人 rel="父" name="王大明"></家人>
    <家人 rel="母" name="李慧慧"></家人>
  </同學>
  <同學 no="2" name="林小美" birth="1984/9/21">
    <家人 rel="父" name="林大雄"></家人>
    <家人 rel="母" name="吳琳琳"></家人>
    <家人 rel="兄" name="林大帥"></家人>
  </同學>
</同學資料>
```

XML 檔案格式會利用 <Label>...</Label> 標籤的方式記錄資料：

<標籤名稱 屬性="值"> 內文 </標籤名稱>

<標籤名稱 屬性="值" />

XML文件的字元分為標記 (Markup) 與內容 (content) 兩類。標記通常以<開頭，以>結尾；每一個標籤代表一個元素，元素當中有屬性與內容兩種設定。

## XML 檔案格式優點與缺點

- 大多瀏覽器可幫忙排版成較易讀格式

缺點：

- 儲存檔案容量較大
- 不一定適合轉換成表格型式

## 思考流程與使用套件

---



下載檔案



開啟檔案



解析檔案

## Python 對 XML 的解析工具

---



1

xml.dom

2

xml.etree

3

xmltodict

## 以這個 xml 檔案為例

```
<?xml version="1.0" encoding="UTF-8"?>
<CUPOY>
<Title>爬蟲馬拉松</Title>
<Author>Wei</Author>
<Chapters>
<Chapter name="01">資料來源與存取</Chapter>
<Chapter name="02">靜態網頁爬蟲</Chapter>
<Chapter name="03">動態網頁爬蟲</Chapter>
</Chapters>
</CUPOY>
```

如果我們想要取出檔案中，**紅色**的部分該怎么做？

```
1 import xml.dom.minidom
2
3 # 存取檔案
doc = xml.dom.minidom.parse("./sample.xml")

# 存取我們的資訊
print(doc.getElementsByTagName("Title")[0].firstChild.nodeValue)

# 用迴圈存取我們的資訊
chapters = doc.getElementsByTagName("Chapter")
for chapter in chapters:
    print (chapter.getAttribute('name'), chapter.firstChild.nodeValue)
```

## 一個簡單的範例 - xml.etree

---



```
3 # 存取檔案
    tree = ET.parse('./sample.xml')
    root = tree.getroot()

    # 存取我們的資訊
    print(root[0].text)

    # 用迴圈存取我們的資訊
    chapters = root[2]
    for chapter in chapters:
        print (chapter.attrib['name'], chapter.text)
```

## 一個簡單的範例 - xmltodict

```
1 import xmltodict
2
3 # 存取檔案

    with open('./sample.xml') as fd:
        doc = dict(xmltodict.parse(fd.read()))

    # 存取我們的資訊
    print(doc['CUPOY']['Title'])

    # 用迴圈存取我們的資訊
    chapters = doc['CUPOY']['Chapters']['Chapter']
    for chapter in chapters:
        print (chapter['@name'], chapter['#text'])
```



xml.dom



xml.etree



xmldict **建議使用**

### 1. xml.dom

將 XML 資料在記憶體中解析成一個樹狀結構，通過對樹的操作來操作。

### 2. xml.etree

輕量級的 DOM，具有方便友好的API。程式碼可用性好，速度快，消耗記憶體少。

### 3. xmldict

將 XML 轉成 Dict，可以利用 Dict 的方式做操作。

## 重要知識點複習

- 了解 xml 檔案格式與內容
- 能夠利用套件存取 xml 格式的檔案



## Difference between XML and HTML

### Difference Between XML and HTML (with Comparison)

The prior difference is that in XML there are provisions for defining new elements while HTML doesn't provide a specification to define new

[techdifferences.com](http://techdifferences.com)

完整比較 XML 跟 HTML 的關係與差異。

## 解題時間

解題時間  
LET'S CRACK IT

Sample Code & 作業

開始解題



[下一步：閱讀範例與完成作業](#)

