

## D2：使用 Python 解析存取 CSV 與 XML 等檔案內容

📖

🖨

💬

📝

課程閱讀

本日作業

問題討論

學習心得(完成)

### 使用 Python 解析存取 CSV 與 XML 等檔案內容

#### 使用 Python 解析存取 CSV 與 XML 等檔案內容

00

Day 02 資料來源與檔案存取

使用 Python 解析存取 CSV 與 XML 等檔案內容







出題教練：張梅元

#### 本日知識點目標

CUPOY

本日知識點目標

- 了解 csv 和 xml 檔案格式與內容
- 能夠利用套件存取 csv 和 xml 格式的檔案

#### CSV 檔案格式

CSV (Comma-Separated Values，逗點分隔值) 檔案以純文字形式儲存文字資料，每條記錄由欄位組成，欄位間利用逗號作為分隔符號。

可以使用一般的文字編輯器以原始格式開啟，也可以使用 excel 或 number 等試算表軟體以表格方式開啟

一般格式如下，第一列會記錄格式，第二列開始記錄資料。

1	id,name,relation	C1	1	C2	C3
2	1,Harry,father	1	id	name	relation
3	2,Chloe,mother	2	1	Harry	father
4	3,Dylan,brother	3	2	Chloe	mother
5	4,Emily,sister	4	3	Dylan	brother
		5	4	Emily	sister

#### CSV 檔案格式優點與缺點

- 優點：
- 結構單純
  - 人機皆可讀
  - 檔案小
- 缺點：
- 未限定編碼(big5, utf-8 ...)
  - 值內有逗點「,」可能造成欄位判斷錯誤
  - 第一行不一定是欄位名稱
  - 換行問題

#### 思考流程與使用套件



#### 一個簡單的範例

```
1 import csv
2
3
4 spamReader = csv.reader(open('eggs.csv'), delimiter=',', quotechar='"')
5 for row in spamReader:
6     print(','.join(row))
7
8 # Spam, Spam, Spam, Spam, Spam, Baked Beans
9 # Spam, Lovely Spam, Wonderful Spam
10
```

#### 更多的參數使用

- csv.reader(csvfile, dialect='excel', \*\*fmtparams)
- 返回值：csv object
  - 參數表：
    - csvfile: 檔案位置
    - dialect: 用什麼方式定義 CSV 格式，預設採用逗號 (excel 風格)，常見的設定有：
      - delimiter：分隔符號，預設是逗號
      - lineterminator：換行符號，預設是 \n
      - 更多可以參考：



fmtparams: 有更多參數可以複寫，一般不會動到

#### XML 檔案格式

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。



XML

<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，>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。

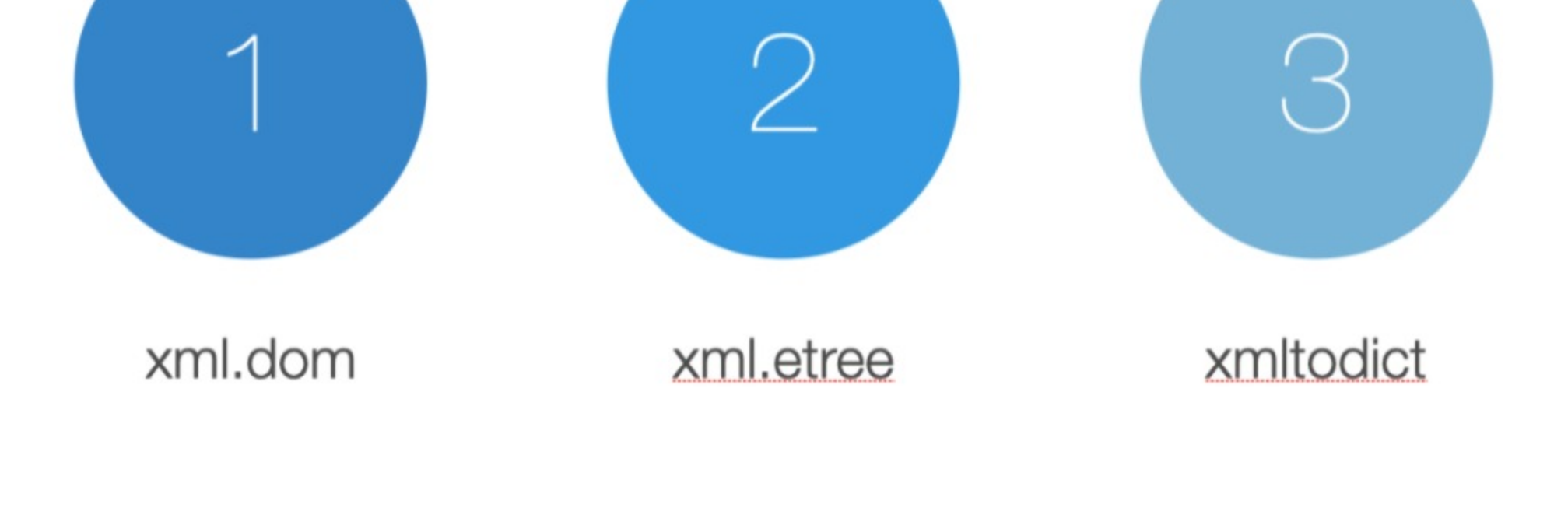
#### XML 檔案格式優點與缺點

- 優點：
- 可以存放結構較複雜的資料
  - 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



#### Python 對 XML 的解析工具



#### 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，紅色的部分該怎麼做？

#### 一個簡單的範例 - xml.dom

```
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

```
1 import xml.etree.ElementTree as ET
2
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)
```

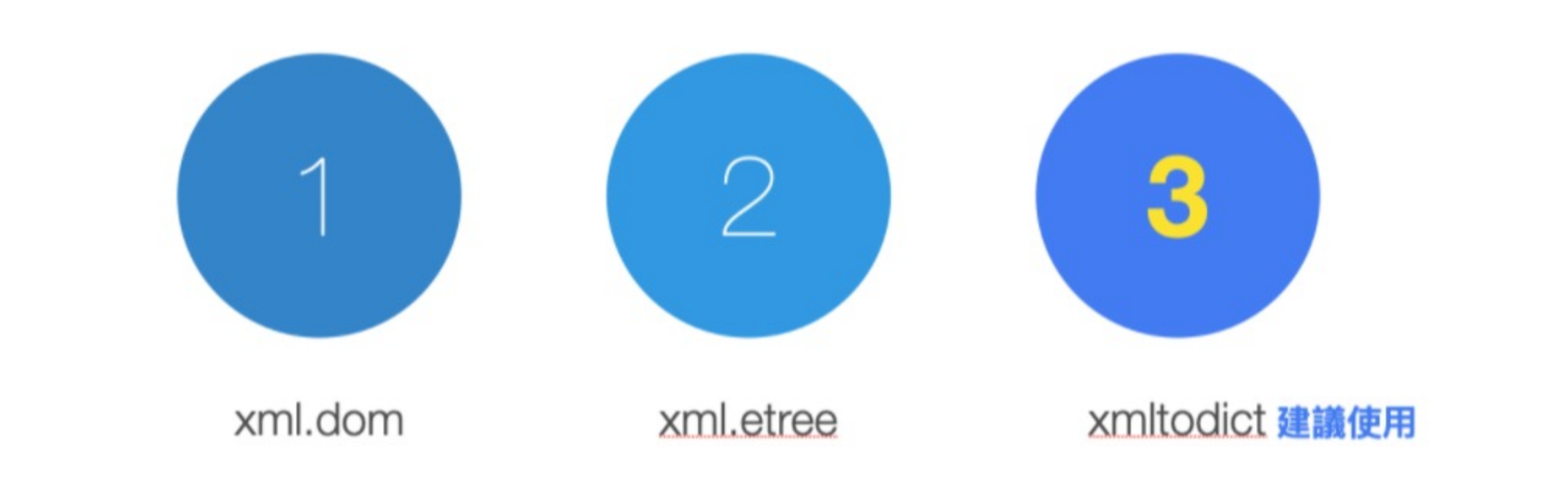
#### 一個簡單的範例 - xmldict

```
1 import xmldict
2
3 # 存取檔案
with open("./sample.xml") as f:
    doc = dict(xmldict.parse(f.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"])
```

#### Python 對 XML 的解析工具



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 csv 和 xml 檔案格式與內容
- 能夠利用套件存取 csv 和 xml 格式的檔案

#### 參考資料

CSV File Reading and Writing




csv — CSV File Reading and Writing — Python 3.8.4rc1 documentation

undefined

docs.python.org

CSV 函式庫的官方文件，上面有完整的參數列表可以參考。

Reading and Writing CSV Files in Python




Reading and Writing CSV Files in Python – Real Python

Learn how to read, process, and parse CSV from text files using Python. You'll see how CSV files work, learn the all-important "csv" library built into Python, and see how CSV parsing works using

realpython.com

除了基本的使用外，該文章提供其他種存取 CSV 格式的做法。

Difference between XML and HTML



Difference Between XML and HTML (with Comparison Chart) - Tech

The prior difference is that in XML there are provisions for defining new elements while HTML doesn't provide a specification to define new element and it uses predefined tags.

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

START



[下一步：完成作業](#)