Python 大數據應用學習參考

黃敦紀
建國中學
2021

目錄

輸出入

```
print (3//2)
```

```
x, y = map (int, input().split())
```

```
print (a, b, sep='') # 和沒有 sep='' 比較
```

選擇結構

```
if ...:
    ...
elif ...:
    ...
else:
    ...
```

```
... if cond else ...
```

```
and, or
```

重複結構

```
while ...:
    ...
```

```
for ... in ...:
    ...
```

```
for i in range (n):
    ...
```

```
for i in range (s, t):
    ...
```

```
for i in range (s, t, inc):
    ...
```

```
for c in s:
    ...
```

list

```
x = [ 1, 2, 3 ] # 中括弧，方括弧
```

```
m = x [1]
```

```
n = len (x)
```

```
x.append (4)
```

```
x.pop (1)
```

```
x.remove (3) # removes the 1st occurrence of 3
```

```
x.clear()
```

```
y = x.copy()
```

```
z = x + [ 4, 5, 6 ]
```

```
c = x.count (2)
```

```
i = x.index (2)
```

```
x.insert (1, 5)
```

```
x.reverse()
```

```
x.sort()
```

```
mx = max (x)
```

```
mn = min (x)
```

```
sm = sum (x) # x can be any iterable
```

```
x, y = map (int, input().split())
```

輸入一行文字

根據空格拆分成數個 tokens (物件方法)，假設輸入測資數量和變數數量一致，否則會發生錯誤

將這些文字 tokens 轉換成整數 (函式呼叫, 參考次頁說明)

比較 C++ iostream 輸出入串流：

```
int x, y;
cin >> x >> y;
```

1. 輸入 2 個整數的測資在同一行以空格分開或是用換行分兩行都可以順利分別讀入 x 和 y，
   空格和換行同為輸出入串流 (steam) 中這些 token 的 seperators
2. 因為 x 和 y 宣告為整數型別，系統會自動將輸入的 token 視為/轉換為整數

JSON 資料格式請參考課程檔案夾中之
<App Inventor 大數據應用專題設計製作>

參考書目請看 <601 科技應用課程計畫.pdf>

模組化設計是程式設計的一個重要的技巧，開發一個功能較為完整的程式專案通常不會像解題刷題那類的程式只有短短幾十行，如果沒有將程式碼模組化，會很難有效地發展與維護。

模組化設計中對程序的操作主要包括兩種結構：
1. 函式 (function) 與
2. 物件導向 (object-oriented programming)。

函式是比較基礎的語法結構，之前的課程應該都有學過，但應用上來講可能練習的機會不多。物件導向則相對上進階一點。

以前頁的輸入為例，`map（int,` *something*）是對 `map` 這個函式的呼叫，這個函式接受兩個參數；而 `input().split()` 則是 `input()` 回傳的 object (物件) 對其 method (方法) `split()` 的呼叫。

你可以這樣理解，這兩種形式都是在表示對一個或數個資料 (變數、物件等) 做一些操作，通常會將這些操作的結果回傳至上一層呼叫的程序。假如操作是「**動詞**」而資料是「**受詞**」，則函式呼叫的形式為：

動詞（受詞, …）

這裡的**動詞**是函式 (function) 的名稱，**受詞**是變數等語法構建的名稱。

而物件導向的形式則為：

受詞.動詞（…）

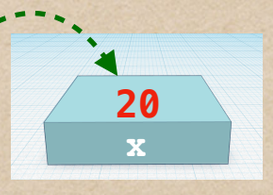
這裡的**受詞**是物件 (object) 的名稱，**動詞**是定義於該物件所屬的類別 (class) 內方法 (method) 的名稱。

我們上課用 Colab 是 Google 的雲服務，Colab 使用 Jupyter Notebook 環境也可以安裝在自己的電腦就可以離線操作 (生科 1 的電腦也有裝)。要在自己的電腦使用 Jupyter Notebook 一般藉由 Anaconda 平台相對上穩定，鼓勵同學自己安裝，不過 Anaconda 平台整組需要不小的磁碟空間，安裝前請先注意一下，有興趣的同學如果有問題可以問我。

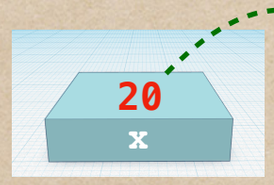
網路上雖然現在中文資料也很豐富的，不少也整理得很好，不過還是建議同學找資料時也查找英文的，同一個問題多比較幾個答案可以讓你對操作的用法有更完整的認識。讀英文資料也順便練習英文。

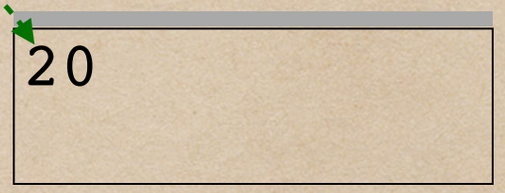
純量變數

```
x = 20
```



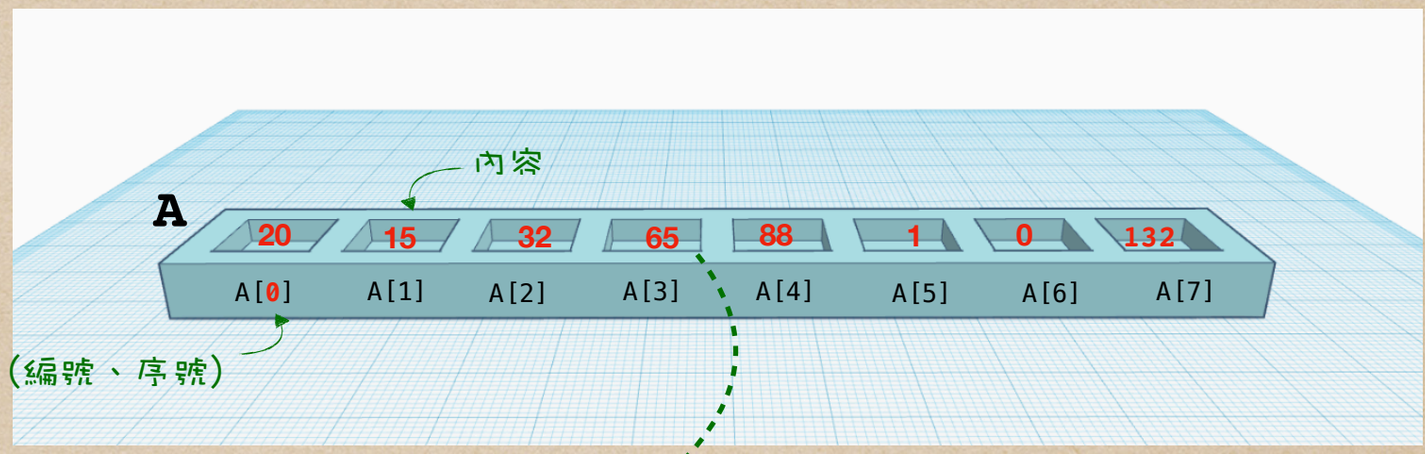

```
print (x)
```





List

```
A = [ 20, 15, 32, 65, 88, 1, 0, 132 ]
```




```
print (A[3])
```

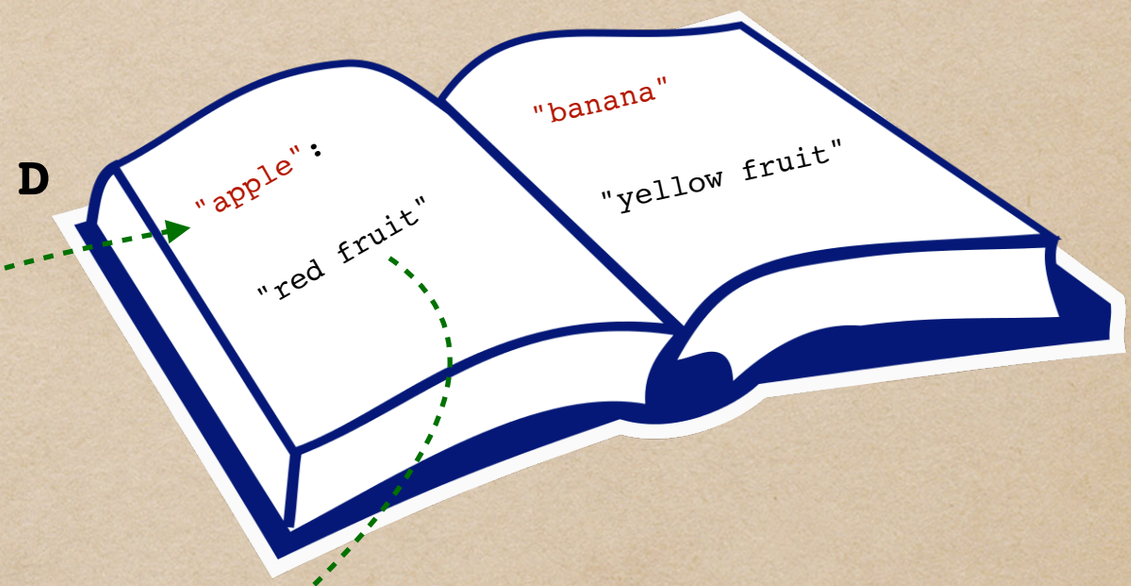
```
65
```

Dictionary

```
D = { "apple" : "red fruit", "banana" : "yellow fruit" }
```

key 鍵 / value 值 / 項目 item, entry

每個字典裡的項目是一個 key-value pair (鍵值對)



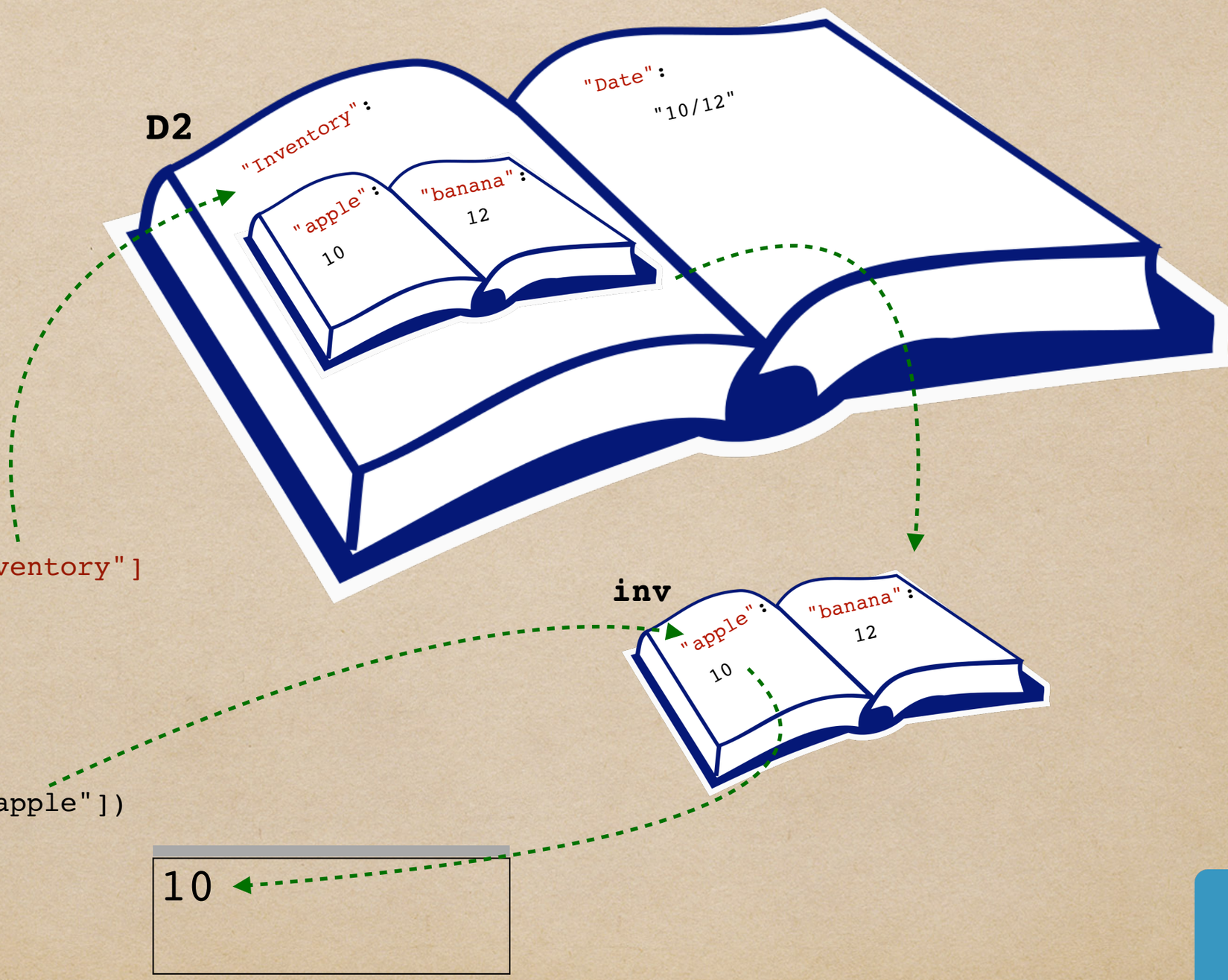
查字典：

```
print ( D["apple"] )
```

```
red fruit
```

Nested (巢狀、Hierarchical 層級的) Dictionary

```
D2 = { "Inventory": { "apple":10, "banana":12 }, "Date": "10/12" }
```




```
inv = D2 ["Inventory"]
```

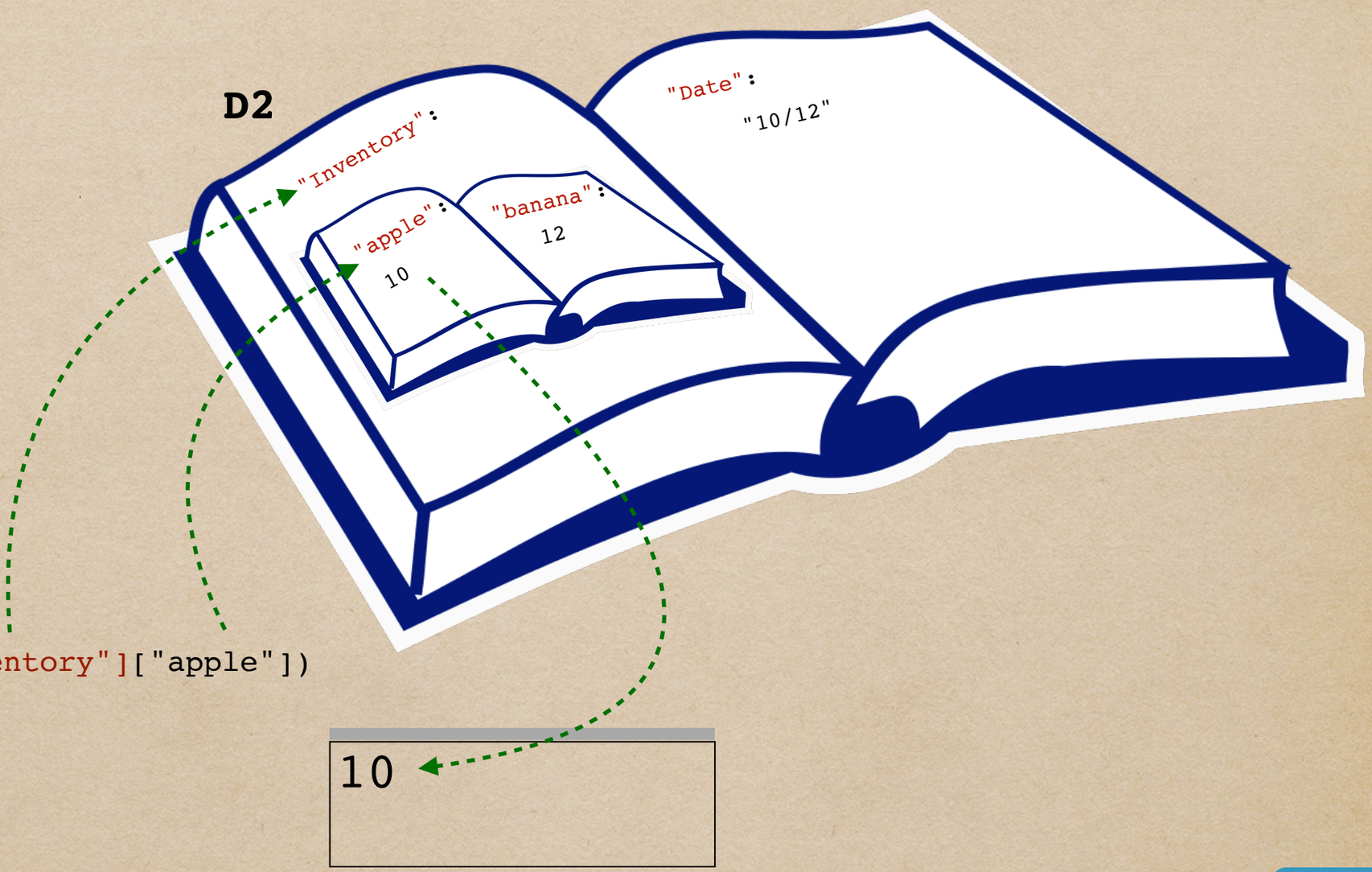
```
print (inv ["apple"])
```

```
10
```

Nested (巢狀、Hierarchical 層級的) Dictionary (續)

或者直接參照

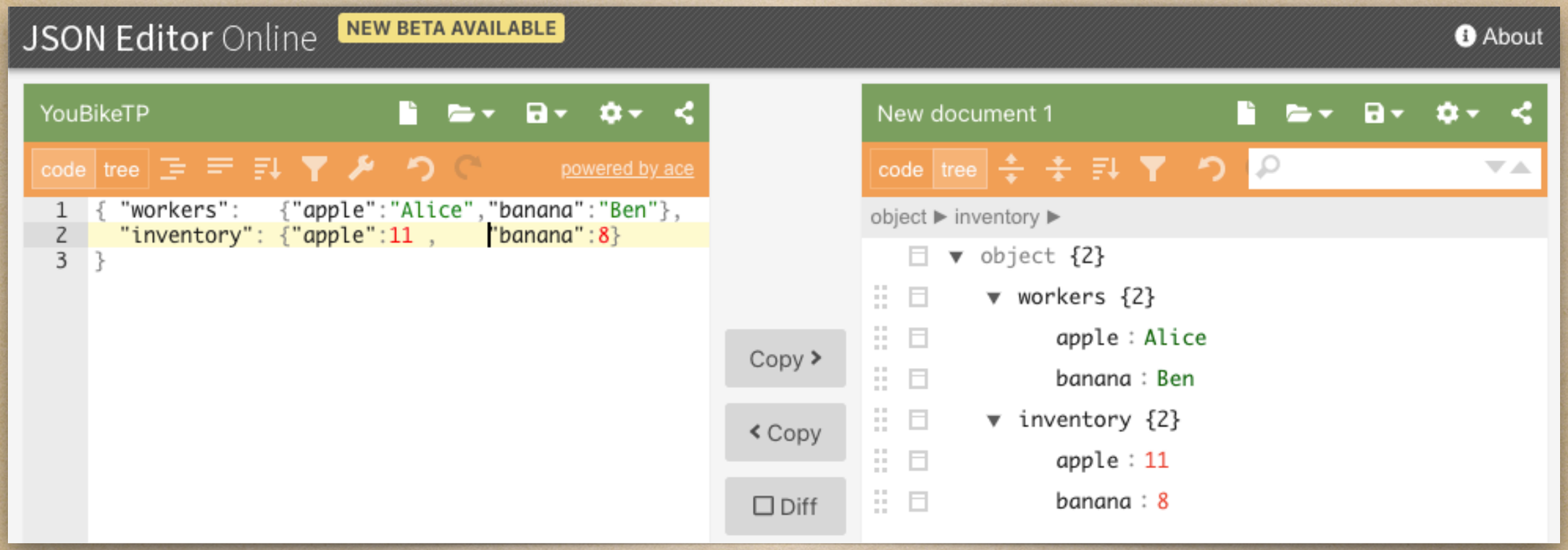
```
D2 = { "Inventory": { "apple":10, "banana":12 } : , "Date": "10/12" }
```




```
print (D2 ["Inventory"]["apple"])
```

```
10
```

JSON Editor



1.JSON Editor 左右編輯器各為獨立的操作視窗，
2.各自的左上角 code/tree 切換顯示的格式，
3.中間有操作兩個編輯器複製和比較的功能按鈕。
4.右邊的編輯器右上角有搜尋格。

JSON 的格式直接對應到 Python 的資料結構。JSON 是「檔案格式」，list 和 dictionary 是 Python 的「資料結構型別」，這兩者的差別如果各位以前的課程沒有學過的話，我們有機會再來細說，現階段先專注在如何操作。

如何取得台北市的 Youbike 即時資訊

```
import requests
```

匯入 requests 這個模組。這類功能就是我們使用 Python 的原因，開源分享社群已經寫好很多常用功能的模組可以使用。

```
r = requests.get ("https://tcgbusfs.blob.core.windows.net/blobyoubike/YouBikeTP.json")
```

呼叫 `requests` 裡面這個模組的 `get` 函式，取得參數裡寫的網址所提供的資料。

```
ubikes = r.json()
```

將 `r` 轉換成對應的 Python 資料結構，將結果指定給 `ubikes` 這個變數。上述的網址列是台北市 Youbike 即時資訊，轉換後的 `ubikes` 可從 JSON Editor 看出來第一層是 dictionary。

作業要求你去找特定站的資訊，這個時候你可以用站名到 JSON Editor 去找到站號，而要知道站名可以到 Youbike 網站打開站點地圖來找。

這樣可以完成作業的要求。可是這樣的流程很麻煩，每次要知道一個站的資訊就要去查找地圖，這不是一個完整的應用，這就是我們要談到設計思考的地方，可以怎麼設計你的應用，讓這個流程自動化，例如老師所提供的範例：給定一個地址，就可以提供附近站點的資訊。這樣的過程，同學可以用來規劃自己專題。

如何取得新北市的 Youbike 即時資訊

同樣的做法可以取得新北市的 Youbike 即時資訊 (不完整)

```
import requests

r = requests.get ("https://data.ntpc.gov.tw/api/datasets/71CD1490-A2DF-4198-BEF1-318479775E8A/json?page=0&size=100")

ntpubikes = r.json()
```

請注意：台北市的 `requests.get().json()` 呼叫回傳的資料是 dictionary，新北市的是 list

```
type (ntpubikes)
```

會顯示 `ntpubikes 是一個 list`

接續的課程方向

- 多元資料：
  同學先選，再統整成幾個建議一起研究講解
- 多元 Python 工具模組：(見專題指引)
- 多元專題方案：例如結合 Arduino
- 多元工具：例如 MIT App Inventor (設計思考)
- 程式設計基礎

專題指引

請各位各自去找一個或數個國內外網路上的開放資料 (可參考下一頁)，融合運算處理這些資料，做一個能提供使用者衍生資訊的應用，最好是互動的。

請注意找的資料是不收費的，例如 Google 的資料會收費，請不要使用。一個原則是不要找需要登入帳號、沒有要求綁定信用卡的。有問題請詢問老師

1. 資料要包含頻繁更新的，最少一個月更新一次。JSON 或 csv 都可以。
2. 可以做網頁爬蟲但不建議。(因為網頁碼常常更替，且大多數網頁不歡迎爬蟲。但開放平台公開提供的大數據其資料目的就是提供給大家以程式對接使用)
3. 你可以以至少 3 個方向來開發這個專題：
    - 3.1. Python 應用、
    - 3.2. App Inventor 或用其他工具開發手機大數據應用手機 App、以及
    - 3.3. 用 firmata 結合 Arduino 開發專題 / 結合網路和 Arduino 做無線 Arduino IoT 應用。
4. 建議學習使用常用 Python 模組，例如
    - 資料處理：Pandas, Numpy, SQLite;
    - 使用者介面：Matplot, pyWidget, tkinter, PyQt, Folium, etc.;
    - 網頁爬蟲：Selenium, BeautifileSoup

這些都是很好用、以後用得到的工具，也是使用 Python 的一大價值。

國內外開放數據平台例

✋ 請注意你的應用程式不要頻繁去網頁抓取資料，無必要地增加伺服器負擔。

- Kaggle
- 政府資料開放平台
- 臺北市政府交通即時開放資料專區
- 新北市政府資料開放平臺 API
- 全國公共運輸旅運資料服務API
- 臺北市資料大平台
- JSON Editor

各位在規劃自己的專題歡迎來找老師討論。有一些東西可以先幫你考量。例如新北市的 Youbike 資訊找不到即時的資料，那我就會建議你可以用下載的資料作為範例來做就好，這樣對專題在時間內完成會是比較務實的安排，雖然成品功能沒能那麼完整，但是就學期成就表現，我會看你其他數據處理應用的發揮，這樣在時間和學習歷程的呈現之間取的一個合理的平衡。

如果有可能的話，我會提供你需要的程式碼資源。例如我所示範的 Youbike 範例，在從地址取得座標這塊，技術上比較有難度，我也是從網路上拿別人開源分享的模組，像這樣的模塊，除非你特別有興趣，我都不會要求你一定要都鑽研清楚。只要版權沒有疑慮，這樣的做法也是現在須要很快開發一個應用原型常見的模式。

重點在於你能在應用的設計中某個環節發揮出獨特的價值，每個模組都是自己做的在時間的限制下並不實際，在學習目標的達成上也沒有必要。

不過你不能在網路上看到一段現成的程式，能夠完成某種功能，在老師不知道情況下單純只交那段程式作為專題成品，完全沒有自己貢獻的模組。這個份際如果有疑問請找老師討論。

有同學拿他做的手遊給我看，相當好，因為這類作品符合這個課程的目標，我鼓勵他學習歷程一併放入。我知道可能有這類作品的同學不少，有的話請來找我討論。

當你把課程內做的大數據應用和其他你課餘做的等多個作品放到學習歷程，呈現的組織就很重要，要想辦法讓有興趣的審查教授能夠看到你想給他看的作品。

例如我整理的講義，一開始是你必須往下捲才可以看到下面的內容，後來內容多了就加了目錄連結讓同學可以點擊直接跳到不同時間需要看的部分。這類技巧也算是設計思考的結果。

如果時間夠，安排帶各位用 github.io 展現介紹自己的網頁，或許可以應用到學校申請。如果同學有興趣也可以自己看。