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 (…)

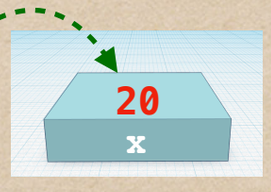
這裡的**受詞**是物件 (object) 的名稱，**動詞**是定義於該物件所屬的類別 (class) 內方法 (method) 的名稱。

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。

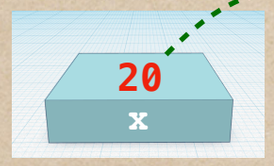
網路上雖然現在中文資料也很豐富的，不少也整理得很好，不過還是建議同學找資料時也查找英文的，同一個問題多比較幾個答案可以讓你對操作的用法有更完整的認識。讀英文資料也順便練習英文。

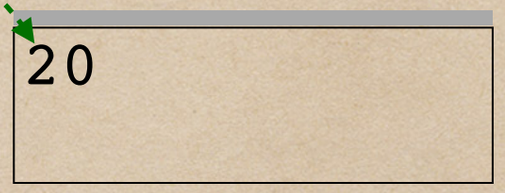
純量變數

```
x = 20
```



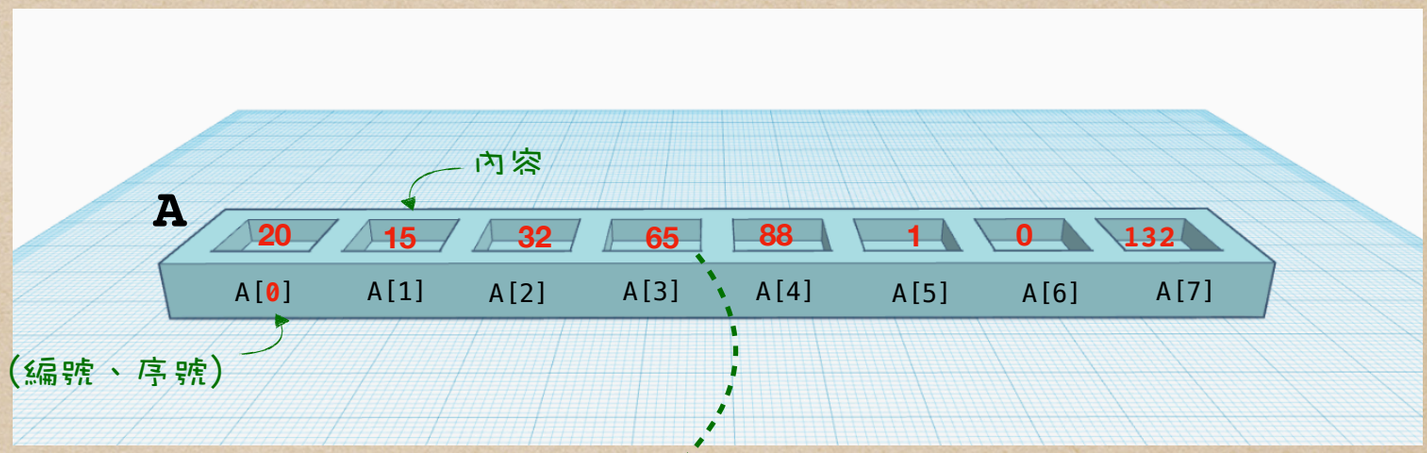

```
print (x)
```





List

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



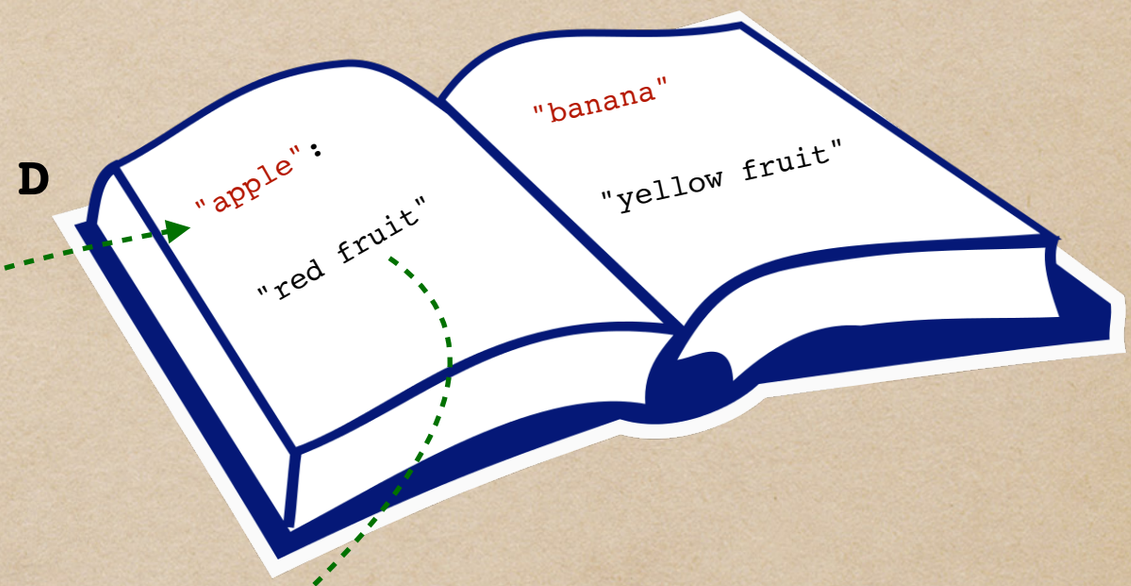

```
print (A[3])
```

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)



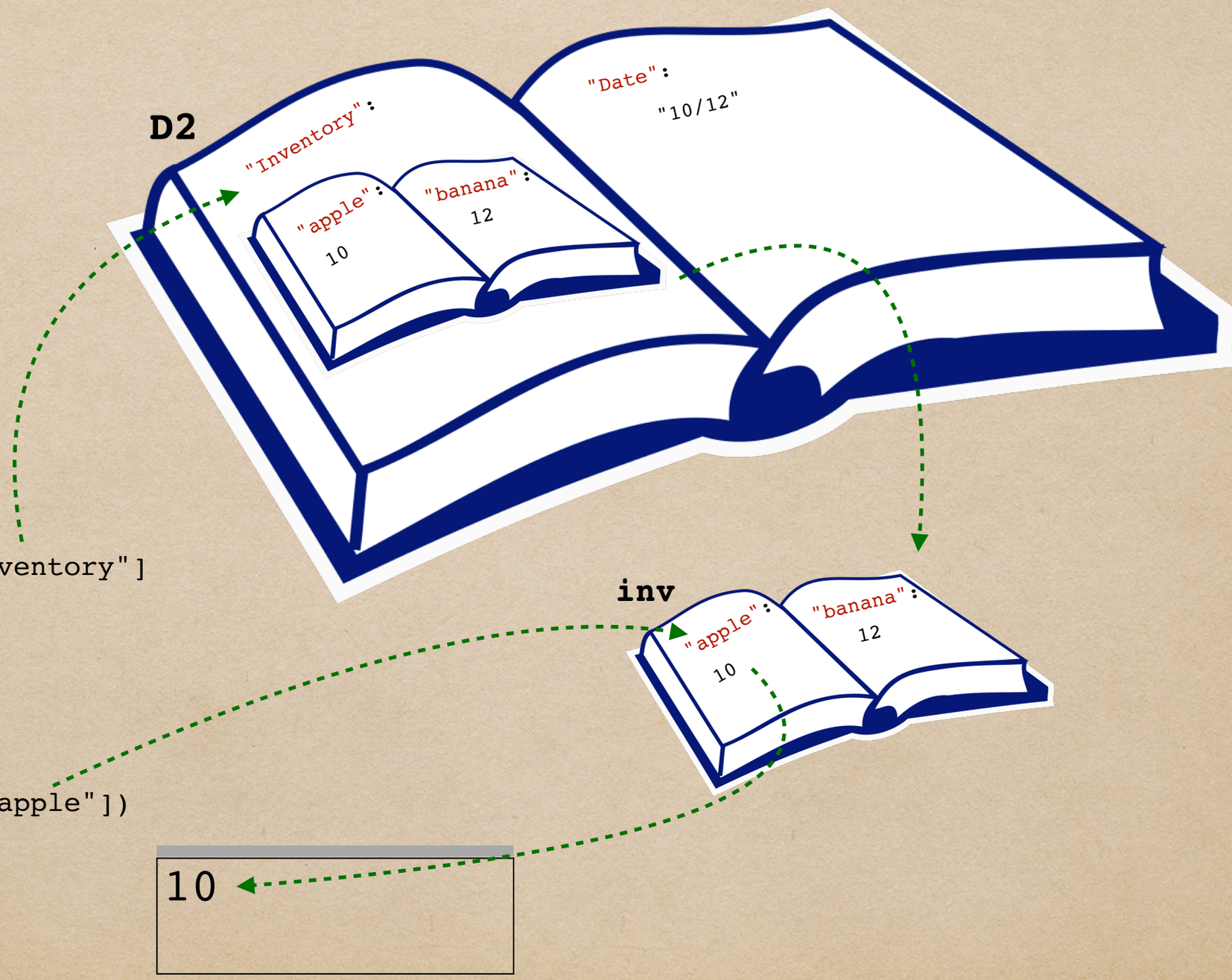
查字典：

```
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"]
```

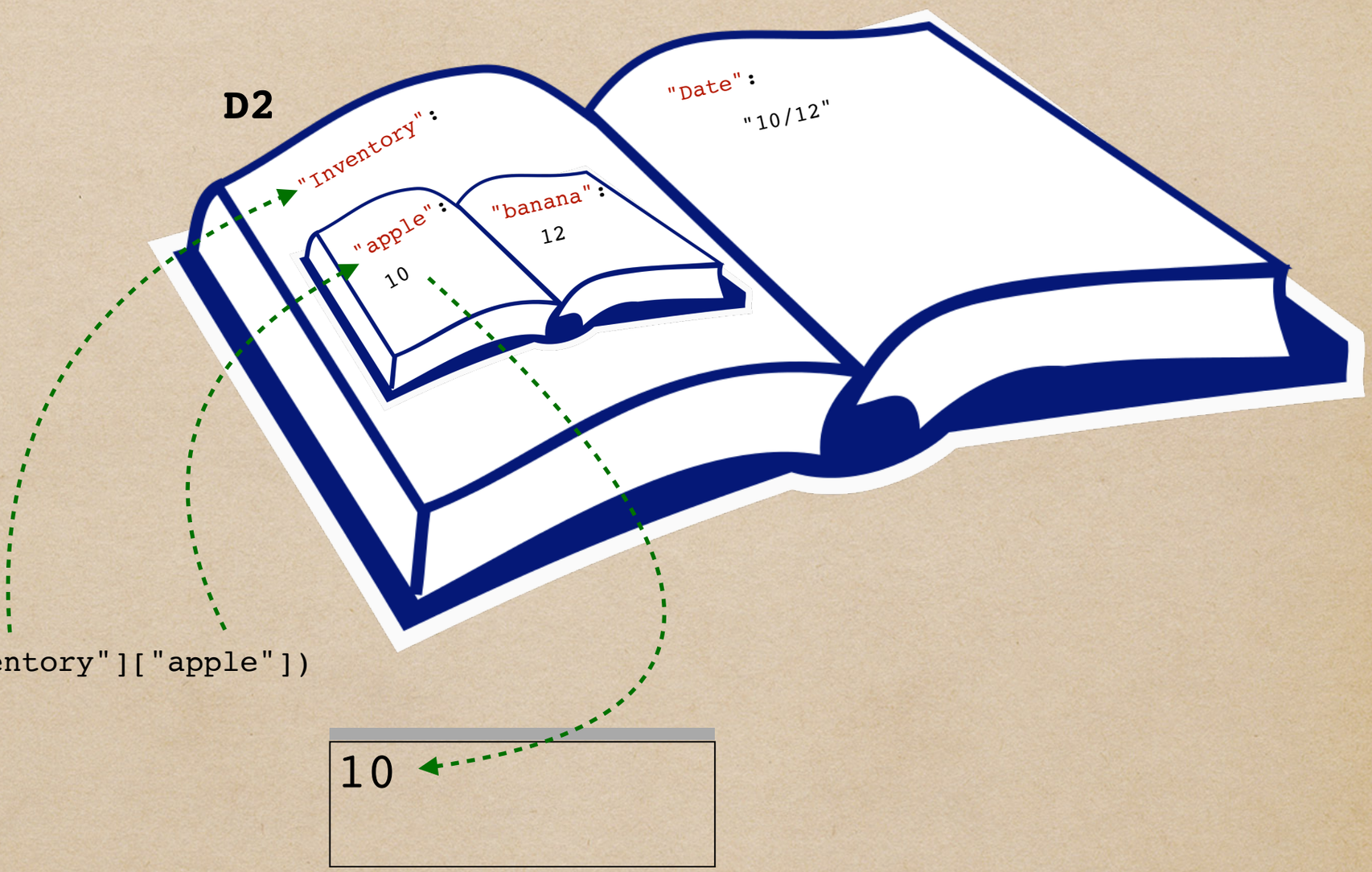
```
print (inv ["apple"])
```

```
10
```

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

或者直接參照

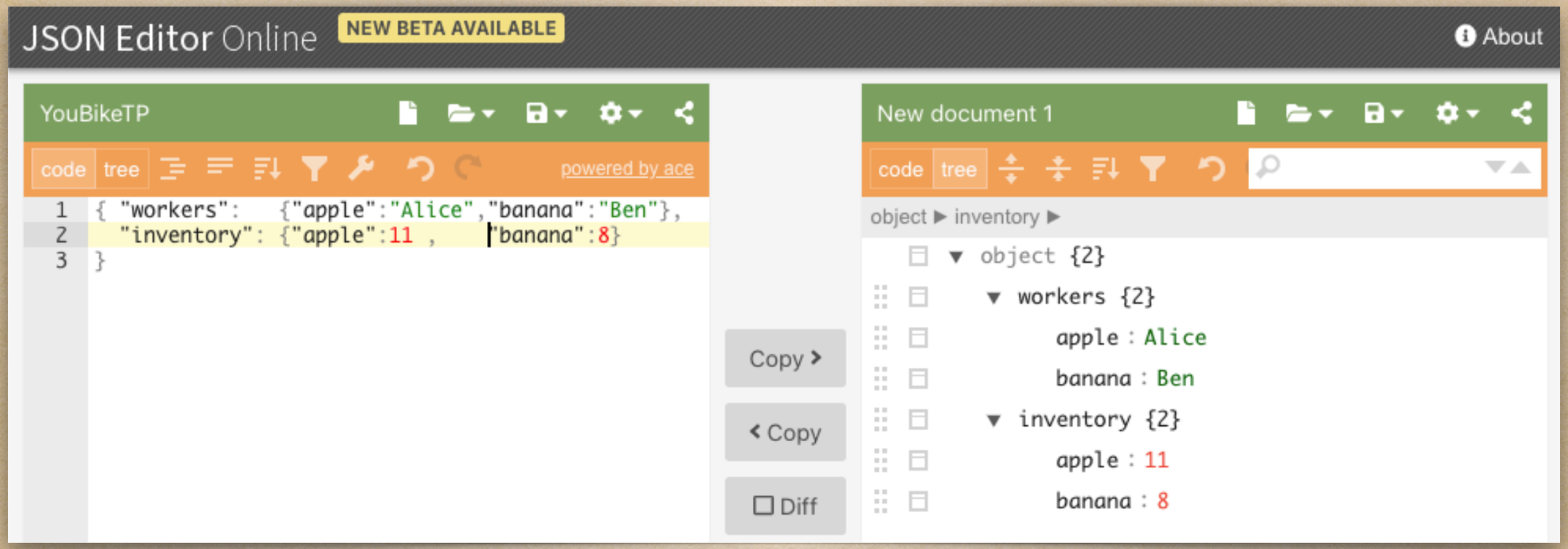
```
D2 = { "Inventory": { "apple":10, "banana":12 } : , "Date": "10/12" }
```




```
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。

如何取得台北市與新北市的 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。

同樣的做法可以取得新北市的 Youbike 即時資訊

```
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