

2

快速認識PYTHON程式語言 PART I

資訊社會必修的
12堂Python通識課

本堂課重點

- ▶ Python執行環境的安裝
- ▶ 常數、變數、與資料型態
- ▶ Python的運算式
- ▶ 認識控制流程
- ▶ 輸入與輸出

Python.org : <https://www.python.org/shell/>

JDOOLE : <https://www.jdoodle.com/python-programming-online>


OnlineGDB :
https://www.onlinegdb.com/online_python_compiler

repl.it : <https://repl.it>

Google Colab : <https://colab.research.google.com/>

PYTHON

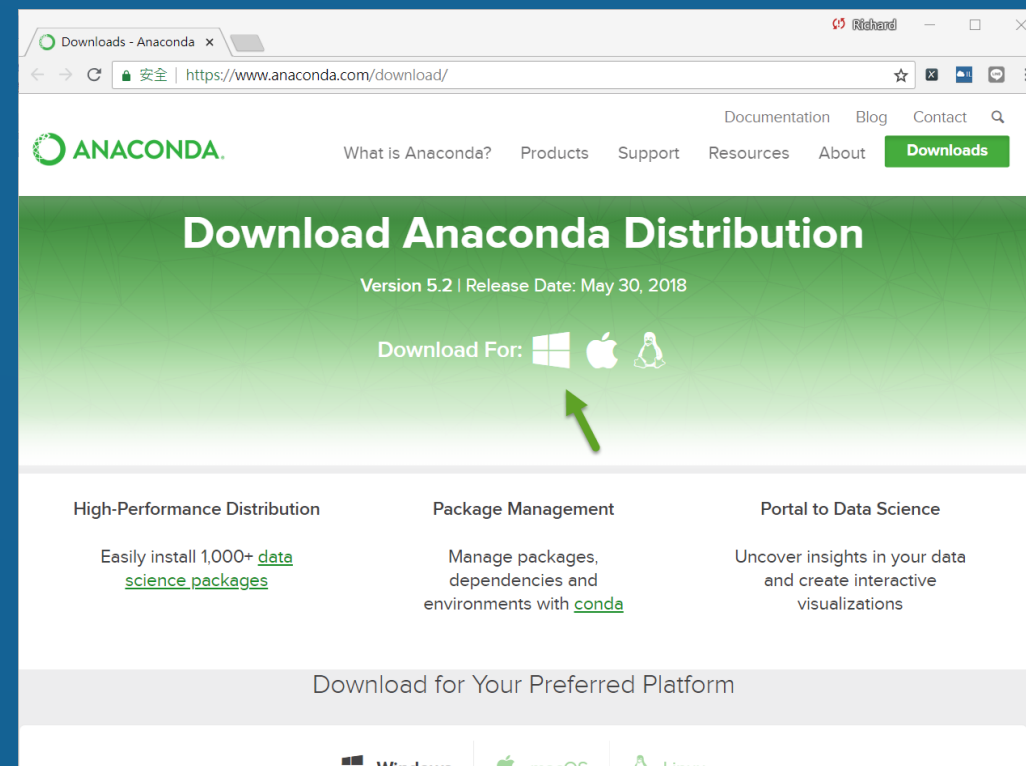
各式各樣的 執行環境

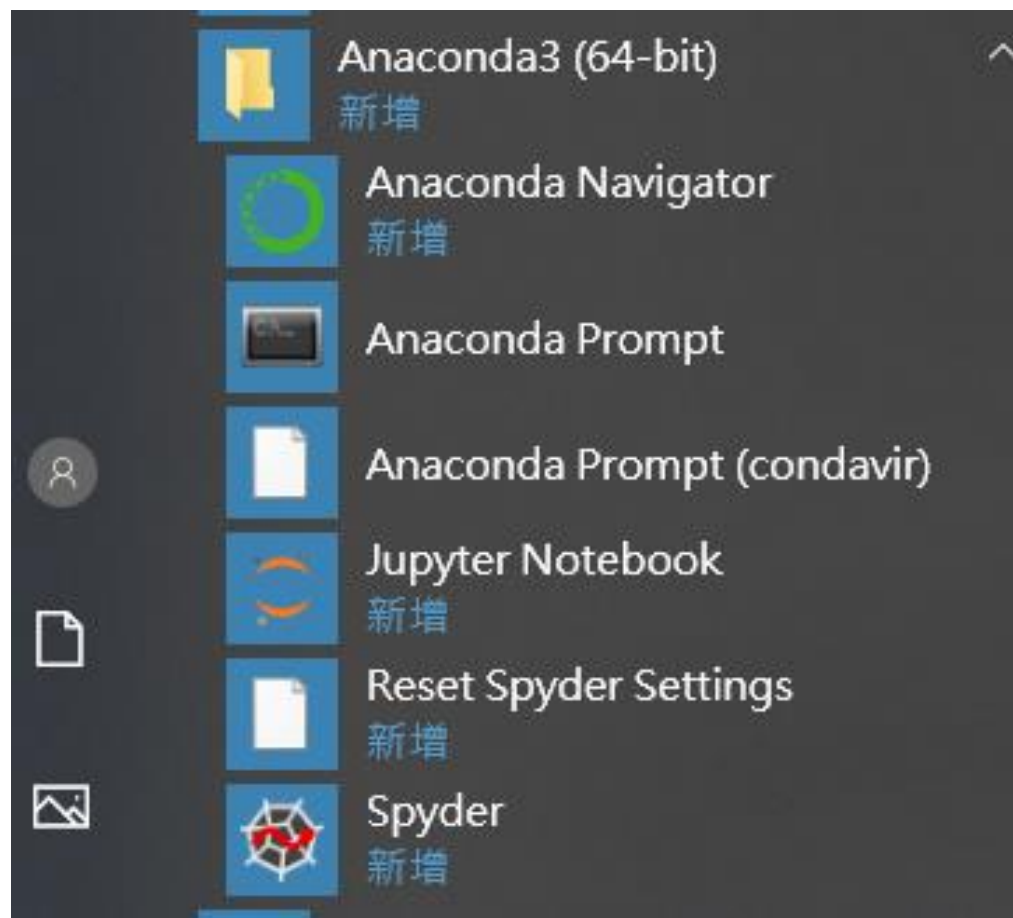
Several white diagonal lines of varying lengths and thicknesses are positioned on the right side of the slide, extending from the middle towards the bottom right corner.

本機環境大補帖

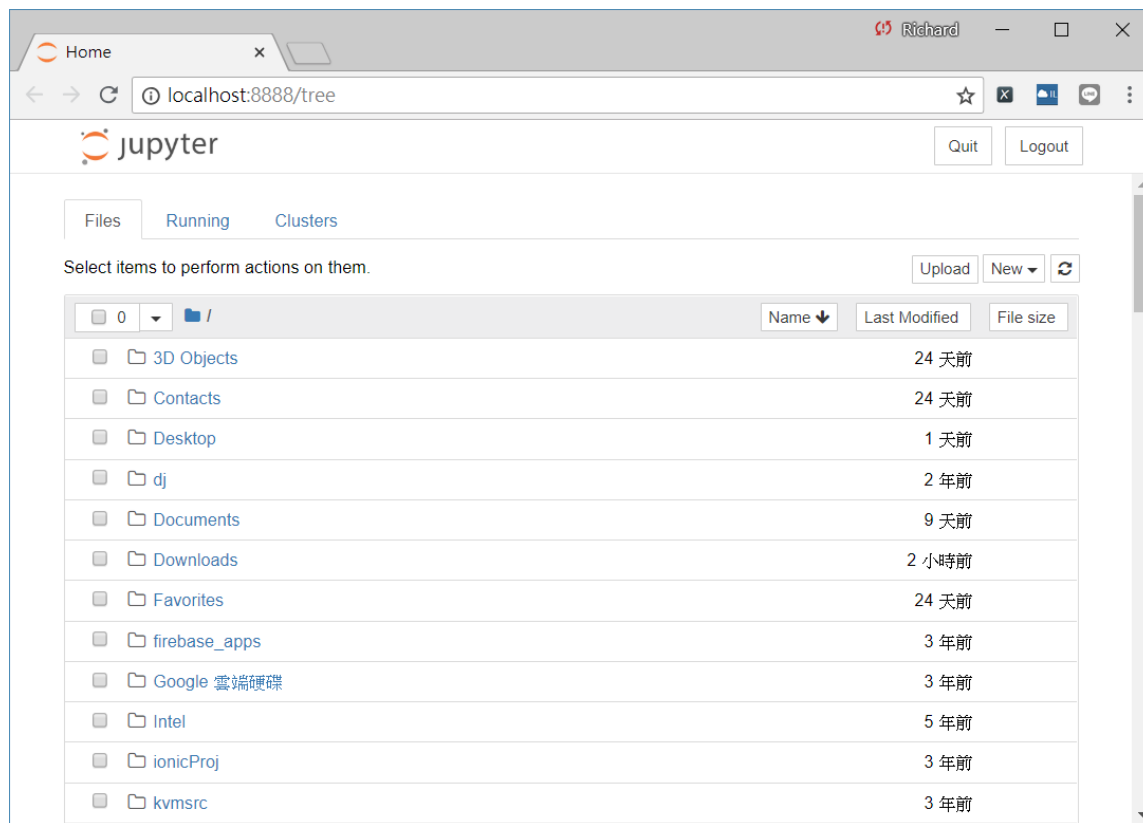
ANACONDA

支援各種作業系統





WINDOWS環境下安裝完成之畫面



JUPYTER NOTEBOOK 執行環境

pyter Notebook

```
09:03:57.218 NotebookApp] The Jupyter Notebook is running at:
[I 09:03:57.218 NotebookApp] http://localhost:8888/?token=951871fcb991f817e77ecee
753dddcee511e662d9b1d2a4
[I 09:03:57.218 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[C 09:03:57.225 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
http://localhost:8888/?token=951871fcb991f817e77ecee753dddcee511e662d9b1
2a4&token=951871fcb991f817e77ecee753dddcee511e662d9b1d2a4
[I 09:03:57.314 NotebookApp] Accepting one-time-token-authenticated connection fr
om ::1
[I 09:04:35.356 NotebookApp] Creating new notebook in
[I 09:04:40.761 NotebookApp] Kernel started: 8aba92ce-4b66-45d3-9065-148c2682c7fc
[W 09:04:50.778 NotebookApp] Timeout waiting for kernel_info reply from 8aba92ce-
4b66-45d3-9065-148c2682c7fc
[I 09:05:10.497 NotebookApp] Adapting to protocol v5.1 for kernel 8aba92ce-4b66-4
5d3-9065-148c2682c7fc
[I 09:06:40.276 NotebookApp] Saving file at /Untitled32.ipynb
[I 09:06:56.525 NotebookApp] Starting buffering for 8aba92ce-4b66-45d3-9065-148c2
682c7fc:56f596fd13344f881a707f78ab44c3f
```

背景執行伺服器畫面，不能關閉

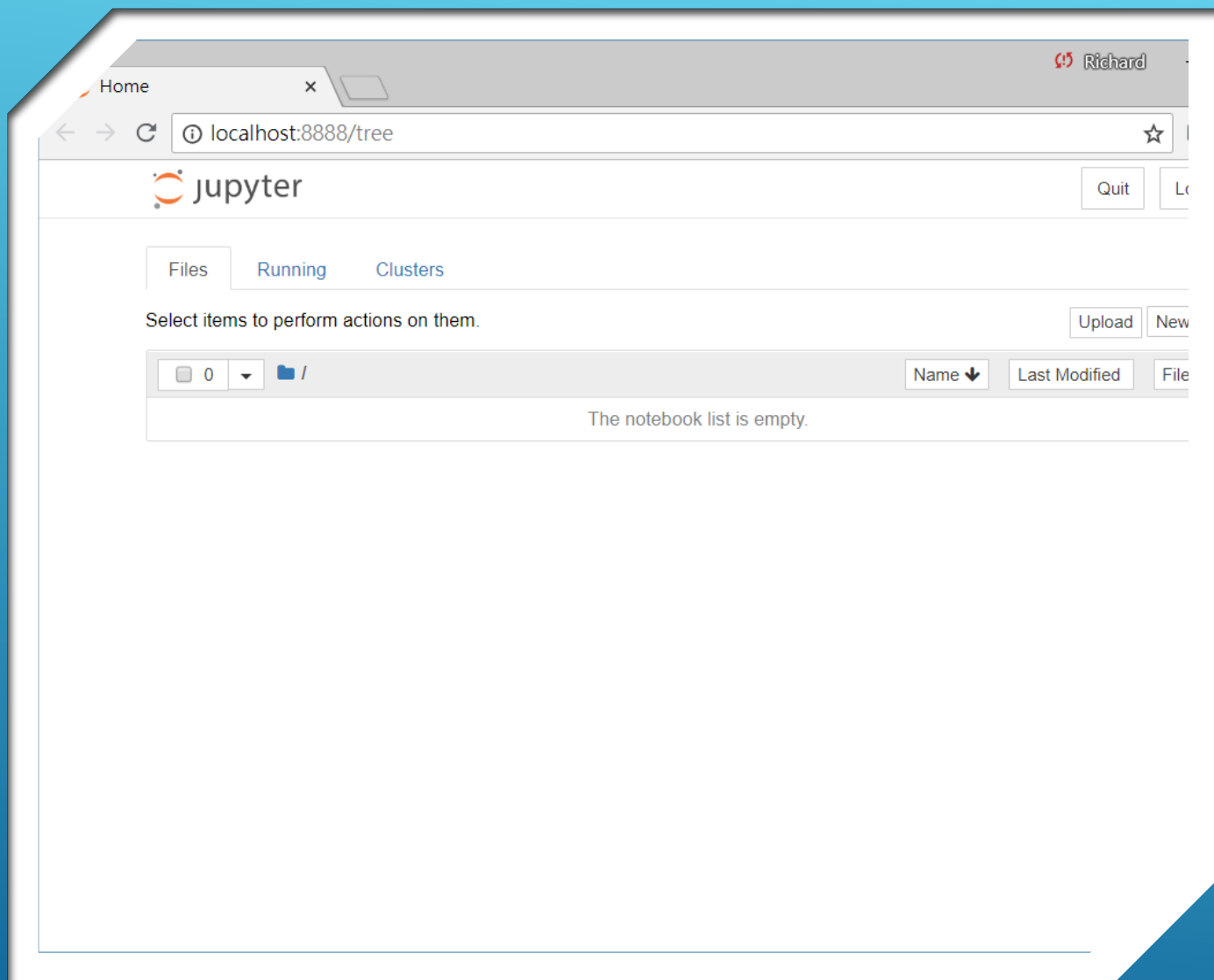
```
Anaconda Prompt
(base) C:\Users\USER>cd \myPython
(base) C:\myPython>jupyter notebook_
```

```

JupyterLab beta preview extension loaded from D:\Anaconda3_5.2\lib\site-packages\jupy
JupyterLab application directory is D:\Anaconda3_5.2\share\jupyter\lab
Serving notebooks from local directory: C:\myPython
0 active kernels
The Jupyter Notebook is running at:
[I 09:22:31.934 NotebookApp] http://localhost:8888/?token=eef8db1936f7daa6f6ffb59e722e8202fc66fd9958954a2f
[I 09:22:31.934 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmati
on).
[C 09:22:31.939 NotebookApp]

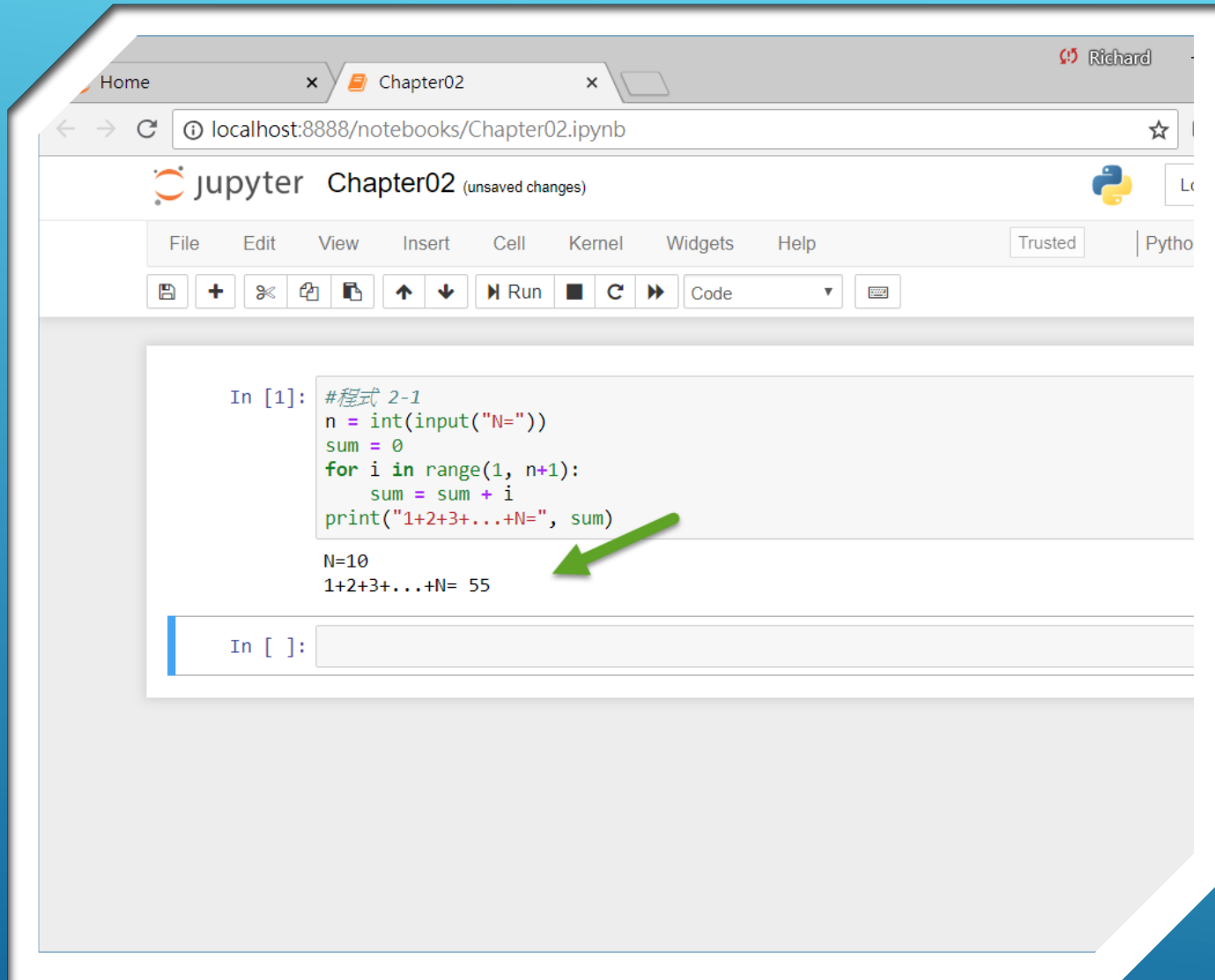
Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://localhost:8888/?token=eef8db1936f7daa6f6ffb59e722e8202fc66fd9958954a2f&token=eef8db1936f7daa6f6ffb5
9e722e8202fc66fd9958954a2f
[I 09:22:32.034 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```

建立專屬目錄，再來執行JUPYTER
NOTEBOOK



會開出一個全新的家目錄

在JUPYTER NOTEBOOK 中輸入程式及 觀看執行結果



變數與常數

- ▶ 常數 constant
- ▶ 字面值 literal
- ▶ 變數 variable



什麼是左值



什麼是右值



指定符號 (=)



「左值 = 右值」，是什麼意思



`count = count + 1` 合理嗎？

變數內容
的設定

$a = 38$

$b = 49$

$c = 13$

$d = a + b + c$

$a, b, c = 38, 49, 13$

$a = a + b + c$

變數及運算式練習

```
1: #程式2-2
2: height = 1.74
3: weight = 63
4: BMI = weight / (height * height)
5: print("Your BMI is", BMI)
```

程式2-2：身體質量指數計算



第一個字元要是英文字母



在第二個字元以後可以使用數字



不要使用除了底線以外的符號



名稱要明確有意義，每一個字母的大小寫會被視為是不同的符號



可以使用底線分隔每一個英文字



變數的名稱長一點沒有關係

變數命名原則

```
$ python
```

```
Python 3.5.1 | Anaconda 2.4.1 (x86_64) | (default, Dec 7 2015, 11:24:55)
```

```
[GCC 4.2.1 (Apple Inc. build 5577)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> a = 38
```

```
>>> type(a)
```

```
<class 'int'>
```

```
>>> a = 38.0
```

```
>>> type(a)
```

```
<class 'float'>
```

```
>>> a = '38'
```

```
>>> type(a)
```

```
<class 'str'>
```

使用PYTHON SHELL查詢資料型態

基本資料型態

- ▶ 數值：整數int，浮點數float
- ▶ 字串：str
- ▶ 布林：Boolean
- ▶ 串列：list
- ▶ 元組：tuple
- ▶ 字典：dict
- ▶ 集合：set

整數型態變數

- ▶ 沒有小數點的數值
- ▶ 可以使用int()進行型態轉換
- ▶ 變數使用之前，直接設定整數值即算完成宣告
 - ▶ temperature = 38

浮點數型態變數

- ▶ 有小數點的數值
- ▶ 可以使用float()進行型態轉換
- ▶ 變數使用之前，直接設定具小數的數值即算完成宣告
 - ▶ temperature = 38.0

字串型態變數

- ▶ 使用雙引號或單引號所包含的文數字
- ▶ 可以使用str()進行型態轉換
- ▶ 變數使用之前，直接設定以單引號或雙引號含括的文數字
 - ▶ name = "Richard"
 - ▶ name = 'Richard'
 - ▶ message = "That's good!"

串列

- ▶ 把許多資料項目串成一串的複合式資料型態
- ▶ 同一變數內的資料項目以數字為索引
- ▶ 同一串列中的資料項目，任何資料型態都能夠接受
- ▶ 宣告方式
 - ▶ `my_list = list()`
 - ▶ `my_list = []`
 - ▶ `temperatures = [35, 36.7, 38, 34.5]`
 - ▶ `scores = [[45, 65, 66],[65, 55, 98]]`

取得串列中資料 項目的方法

- ▶ `print(chi_scores[3])`
- ▶ `print(chi_scores[3:])`
- ▶ `print(chi_scores[2:5])`
- ▶ `print(chi_scores[-1])`

新增串列 資料項目的方法

```
scores = [89, 74]  
scores.append(45)  
scores.append([34, 56])  
scores += [55, 66]  
print(scores)
```

► [89, 74, 45, [34, 56], 55, 66]

函數名稱	用途	使用範例
append	在串列的後面附加上更多的元素	lst.append(x)
clear	清除串列中的所有內容	lst.clear()
copy	把串列的內容以複製的方式複製一份到另外一個串列	lst2 = lst1.copy()
count	計算指定元素在串列中出現的次數	lst.count(65)
extend	把兩個串列串接成一個串列	lst1.extend(lst2) lst.extend([55, 66, 77, 88])
index	找出指定元素在串列中首次出現的索引值	lst.index(88)
insert	把某一元素插入到指定的位置	lst.insert(4, 100)
pop	傳回串列中的最後一個元素，並把它移除	lst.pop()
remove	移除串列中指定元素的第一個	lst.remove(99)
reverse	把串列中的元素位置倒序排序	lst.reverse()
sort	把串列中的元素由大至小排列順序	lst.sort() lst.sort(reverse=True)

- ▶ `lst1 = [65, 45, 98, 48, 87]`

- ▶ `lst2 = lst1`

- ▶ `lst3 = lst1.copy()`

- ▶ `lst2[3] = 100`

- ▶ `lst3[2] = 100`

- ▶ `print("lst1:", lst1)`

- ▶ `print("lst2:", lst2)`

- ▶ `print("lst3:", lst3)`

- ▶ `lst1: [65, 45, 98, 100, 87]`

- ▶ `lst2: [65, 45, 98, 100, 87]`

- ▶ `lst3: [65, 45, 100, 48, 87]`

為什麼串列變數需要COPY這個函數

函數名稱	用途	使用範例
len	傳回此串列的元素個數	len(lst)
max	傳回此串列中的最大值	max(lst)
min	傳回此串列中的最小值	min(lst)
list	建立一個串列變數，如果沒有給參數，就會建立一個空的串列，如果給的是一個字串，則會把字串的個別字元拆開變成串列中的每一個元素	lst = list() lst = list("I love Python")
sorted	傳回排序過的串列	sorted(lst)
reversed	傳回和原串列反向序列的串列	reversed(lst)

可以使用在串列上的函數

```
student1 = ['林小明', True, 89, 45, 67, 'A23001', '林先生']  
student2 = ['王小華', False, 99, 85, 72, 'A23002', '王太太']  
student3 = ['劉明明', False, 67, 45, 92, 'A23003', '劉先生']  
student4 = ['曾小花', False, 99, 99, 100, 'A23004', '曾先生']  
students = [student1, student2, student3, student4]  
print(students[3])  
print(students[3][0])
```

二維串列的宣告與使用

- ▶ 可以把它想成是，一旦設定之後就不能修改內容的串列
- ▶ 除了速度上的考量，在參數傳遞上也更加地便利
- ▶ 宣告時只用的是小括號
 - ▶ `my_tuple = (32, 34, 33, 34, 33)`
 - ▶ `my_tuple = tuple(my_list)`
 - ▶ `rgb = (120, 110, 255)`

元組型態TUPLE

- ▶ 以「鍵」來進行索引操作的資料型態
- ▶ 標準宣告方法：
- ▶ `dict_var = {`
- ▶ `key1 : value1,`
- ▶ `key2 : value2,`
- ▶ `key3 : value3`
- ▶ `}`
- ▶ `dict_var = dict()`
- ▶ `dict_var['key1'] = value1`
- ▶ `dict_var['key2'] = value2`
- ▶ `...`

字典型態DICT {KEY:VALUE}的對應型態

```
week = {  
    'Sunday': "星期日",  
    'Monday': "星期一",  
    'Tuesday': "星期二",  
    'Wednesday': "星期三",  
    'Thursday': "星期四",  
    'Friday': "星期五",  
    'Saturday': "星期六",  
}  
print(week)  
print(week['Sunday'])  
print(week.keys())  
print(week.values())  
print(week.items())
```

```
{'Sunday': '星期日', 'Monday': '星期一', 'Tuesday': '星期二', 'Wednesday':  
'星期三', 'Thursday': '星期四', 'Friday': '星期五', 'Saturday': '星期六'}  
星期日  
dict_keys(['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday',  
'Friday', 'Saturday'])  
dict_values(['星期日', '星期一', '星期二', '星期三', '星期四', '星期五', '星期六'])  
dict_items([('Sunday', '星期日'), ('Monday', '星期一'), ('Tuesday', '星期二'),  
( 'Wednesday', '星期三'), ('Thursday', '星期四'), ('Friday', '星期五'),  
( 'Saturday', '星期六')])
```

- ▶ 數學上集合的特性
- ▶ 可以進行集合的數學操作
- ▶ 以大括號含括集合項目
- ▶ 同一集合變數中的元素內容不會重複

集合型態SET

```
chi_set = {'皮卡丘', '可達鴨', '鯉魚王', '胖丁'}
chi_set.add('綠毛蟲')
chi_set.add('皮卡丘')
eng_set = {'妙蛙種子', '可達鴨', '比比鳥', '皮卡丘'}
print("chi_set:", chi_set)
print("eng_set:", eng_set)
print("chi_set和eng_set的交集:", chi_set.intersection(eng_set))
print("chi_set和eng_set的聯集:", chi_set.union(eng_set))
print("chi_set和eng_set的差集:", chi_set.difference(eng_set))
```

```
chi_set: {'胖丁', '皮卡丘', '鯉魚王', '可達鴨', '綠毛蟲'}
eng_set: {'比比鳥', '皮卡丘', '妙蛙種子', '可達鴨'}
chi_set和eng_set的交集: {'皮卡丘', '可達鴨'}
chi_set和eng_set的聯集: {'胖丁', '皮卡丘', '鯉魚王', '比比鳥', '綠毛蟲', '妙蛙種子', '可達鴨'}
chi_set和eng_set的差集: {'胖丁', '綠毛蟲', '鯉魚王'}
```


算術運算式

關係運算式

邏輯運算式

運算式

Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract design element.

運算符號	功用	運算符號	功用
=	指定值到變數	+, -, *, /	加, 減, 乘, 除
//	整除	**	次方
%	取餘數	+, -	正數, 負數

主要的數學運算符號

```
In [3]: a, b, c = 1, 2, 3
```

```
In [4]: print(a, b, c)
```

```
1 2 3
```

```
In [5]: a, b = b, a
```

```
In [6]: print(a, b, c)
```

```
2 1 3
```



左值和右值的個數一定要一樣

同時設定多個變數的值

運算符號	功用	運算符號	功用
<	小於	<=	小於或等於
==	等於	!=	不等於
>	大於	>=	大於或等於

關係運算式的運算符號

運算符號	功用	運算符號	功用
and	且	or	或
not	否		

邏輯運算式的運算符號

邏輯運算練習（程式2-3）

```
1: # 程式2-3
2: # -*- coding: utf-8 -*-
3:
4: age = int(input("請輸入你的年紀："))
5: with_parent = input("和父母一起來嗎？(Y/N)")
6:
7: if age >= 18:
8:     print("可以看限制級電影")
9: elif age >= 12:
10:    print("可以看輔導級電影")
11: elif (age >= 6 and age < 12) and (with_parent=='Y' or with_parent=='y'):
12:    print("可以看保護級電影")
13: else:
14:    print("只能看普遍級電影")
```