

成為初級資料分析師 | Python 程式設計

資料型態

郭耀仁

*There are only two hard things in Computer Science: cache
invalidation and naming things.*

Phil Karlton

大綱



- 物件、命名與註解
- 內建函數
- 整數與浮點數
- 文字
- 布林
- 資料型態轉換

物件、命名與註解

寫程式究竟是怎麼一回事


- 程式設計師腦中想做些什麼事情
- 用程式語言具象化（**realize**）
- 以直譯器（或編譯器）翻譯成電腦語言
- 由電腦具象化

像是記錄電影的資訊

All


Movies, TV & Showtimes Celebs, Events & Photos News & Community Watchlist Sign in

[FULL CAST AND CREW](#) [TRIVIA](#) [USER REVIEWS](#) [IMDbPro](#) [MORE](#) [SHARE](#)



Avengers: Endgame (2019)

PG-13 | 3h 1min | Action, Adventure, Sci-Fi | 26 April 2019 (USA)



1:06 | Trailer 94 VIDEOS | 493 IMAGES


After the devastating events of *Avengers: Infinity War* (2018), the universe is in ruins. With the help of remaining allies, the Avengers assemble once more in order to undo Thanos' actions and restore order to the universe.

Directors: [Anthony Russo](#), [Joe Russo](#)


Writers: [Christopher Markus](#) (screenplay by), [Stephen McFeely](#) (screenplay by) | [3 more credits](#) »


Stars: [Robert Downey Jr.](#), [Chris Evans](#), [Mark Ruffalo](#) | [See full cast & crew](#) »

[+ Add to Watchlist](#)


 78

Metascore
From metacritic.com

 Reviews
6,915 user | 466 critic

 Popularity
1

Why Thomas Middleditch Prefers a Villain



The *Godzilla: King of the Monsters* star picks his favorite sketch shows and the villainous origin story he can't wait to see.

[Watch now »](#)

Related News

[Gwyneth Paltrow's Goop Summit Proves Hollywood Retirement Is Working for Her](#)
21 May 2019 | Variety

['John Wick: Chapter 4' Gets Summer 2021 Release Date](#)
21 May 2019 | The Wrap

['Falcon and the Winter Soldier': MCU Veterans Daniel Bruhl and Emily VanCamp in Talks to Join Disney+ Series](#)
21 May 2019 | The Wrap

以 = 將資訊儲存至一個物件名稱中

```
In [1]: movie_title = "Avengers: Endgame"  
        release_date = "26 April 2019(USA)"  
        print(movie_title)  
        print(release_date)
```

```
Avengers: Endgame  
26 April 2019(USA)
```

以 # 對程式功能做註解

```
In [2]: # 儲存資訊 <--- 註解
movie_title = "Avengers: Endgame"
release_date = "26 April 2019(USA)"
# 印出資訊 <--- 註解
print(movie_title)
print(release_date)
```

```
Avengers: Endgame
26 April 2019(USA)
```


註解不只是寫給別人看，也是寫給未來的自己看

物件的命名風格

- 小寫
- 中間不能有空格
- 單字用 _ 分隔（Google 搜尋：snake case）
- 開頭不能使用數字
- 不能使用保留字
- 不要使用內建函數

```
In [3]: # 開頭不能使用數字：  
# 56_cannot_die = "56不能亡"
```

保留字列表

| | | | | |
|--------|----------|---------|----------|--------|
| False | await | else | import | pass |
| None | break | except | in | raise |
| True | class | finally | is | return |
| and | continue | for | lambda | try |
| as | def | from | nonlocal | while |
| assert | del | global | not | with |
| async | elif | if | or | yield |

Source: https://docs.python.org/3/reference/lexical_analysis.html#keywords
(https://docs.python.org/3/reference/lexical_analysis.html#keywords).

```
In [4]: # 不能使用保留字  
        # False = 0
```

內建函數

函數是一段被命名的程式碼（**Code block**）

- 指定輸入、參數並餵入函數
- 函數依據輸入、參數執行一段程式碼
- 將輸出拋給使用者

寫程式可以想像成是設計一組又一組的函數來完成指定任務

內建函數列表

```
abs()    delattr()    hash()    memoryview()    set()
all()    dict()    help()    min()    setattr()
any()    dir()    hex()    next()    slice()
ascii()  divmod()    id()    object()    sorted()
bin()    enumerate()  input()  oct()    staticmethod()
bool()   eval()    int()    open()   str()
breakpoint()  exec()  isinstance()  ord()    sum()
bytearray()  filter()  issubclass()  pow()    super()
bytes()  float()  iter()  print()  tuple()
callable()  format()  len()    property()  type()
chr()    frozenset()  list()  range()  vars()
classmethod()  getattr()  locals()  repr()  zip()
compile()  globals()  map()    reversed()  __import__()
complex()  hasattr()  max()    round()
```

Source: <https://docs.python.org/3/library/functions.html>
(<https://docs.python.org/3/library/functions.html>).

```
In [5]: movie_title = "Avengers: Endgame"
        print(movie_title)
```

Avengers: Endgame

```
In [6]: print = "PRINT"
        print(movie_title)
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-6-7eff6704c2ec> in <module>
      1 print = "PRINT"
----> 2 print(movie_title)

TypeError: 'str' object is not callable
```



```
In [7]: del print  
        print(movie_title)
```

Avengers: Endgame

一開始就要用的內建函數

- `print()` 印出物件所記錄的資料內容
- `help()` 瞭解內建函數的作用
- `type()` 瞭解物件的類別

```
In [8]: help(print)
```

Help on built-in function print in module builtins:

```
print(...)  
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Prints the values to a stream, or to sys.stdout by default.

Optional keyword arguments:

file: a file-like object (stream); defaults to the current sys.stdout.

sep: string inserted between values, default a space.

end: string appended after the last value, default a newline.

flush: whether to forcibly flush the stream.

```
In [9]: help(help)
```

Help on _Helper in module _sitebuiltins object:

```
class _Helper(builtins.object)
|   Define the builtin 'help'.
|
|   This is a wrapper around pydoc.help that provides a helpful message
|   when 'help' is typed at the Python interactive prompt.
|
|   Calling help() at the Python prompt starts an interactive help session.
|   Calling help(thing) prints help for the python object 'thing'.
|
|   Methods defined here:
|
|   __call__(self, *args, **kwargs)
|       Call self as a function.
|
|   __repr__(self)
|       Return repr(self).
|
|   -----
|   Data descriptors defined here:
|
|   __dict__
|       dictionary for instance variables (if defined)
|
|   __weakref__
|       list of weak references to the object (if defined)
```

隨堂練習：將電影 <https://www.imdb.com/title/tt4154796> (<https://www.imdb.com/title/tt4154796>) 頁面中的三個資訊儲存為物件並且印出來

- 片長（以分鐘為單位）
- 評分
- 片名

```
In [11]: print(movie_time)
          print(movie_rating)
          print(movie_title)
```

181

8.5

Avengers: Endgame

隨堂練習：使用 **type()** 函數將這三個物件的類別印出來

```
In [13]: print(type_of_movie_time)
          print(type_of_movie_rating)
          print(type_of_movie_title)
```

```
<class 'int'>
<class 'float'>
<class 'str'>
```

整數與浮點數

數值運算符號（ Arithmetic Operators ）

- + 、 - 、 * 、 / ：加減乘除
- ** ：次方
- % ：餘數
- // ：商數

In [14]:

```
print(5 + 6)
print(5 - 6)
print(5 * 6)
print(5 / 6)
print(5**6)
print(5566 % 56)
print(5566 // 56)
```

11

-1

30

0.8333333333333334

15625

22

99

遵循常見的次方優先、先乘除後加減

使用小括號 () 調整優先順序。

In [15]: `77 - 32 * 5/9`

Out[15]: `59.22222222222222`

In [16]: `(77 - 32) * 5/9`

Out[16]: `25.0`

隨堂練習：將目前的攝氏氣溫轉換為華氏氣溫

$$Fahrenheit = Celsius \times \frac{9}{5} + 32$$

```
In [17]: current_tempc = 25
```

```
In [19]: print(current_tempf)
```

```
77.0
```

隨堂練習：將目前的華氏氣溫轉換為攝氏氣溫

$$Celsius = (Fahrenheit - 32) \times \frac{5}{9}$$

In [21]: `print(current_tempc)`

25.0

隨堂練習：計算俠客歐尼爾巔峰時期的 BMI 身體質量指數

NBA 史上最偉大的中鋒之一「柴油引擎」俠客歐尼爾（Shaquille O'Neal）巔峰時期的身高為 216 公分、體重為 147 公斤。

$$BMI = \frac{weight_{kg}}{height_m^2}$$

```
In [22]: shaq_height = 216  
         shaq_weight = 147
```

```
In [24]: print(shaq_bmi)
```

```
31.507201646090532
```

運算符號的小捷徑

將 3h 1min 換算成 181 min 。

```
In [25]: hours = 3  
mins = 1  
mins = mins + hours * 60  
print(mins)
```

181

```
In [26]: hours = 3  
mins = 1  
mins += hours * 60 # 注意這行  
print(mins)
```

181

數值運算符號常走捷徑

- `a += b` as `a = a + b`
- `a -= b` as `a = a - b`
- `a *= b` as `a = a * b`
- `a /= b` as `a = a / b`
- `a **= b` as `a = a ** b`
- `a %= b` as `a = a % b`
- `a //= b` as `a = a // b`

其中我們在後續章節最常使用的是

`a += b` as in `a = a + b`

文字

使用成雙的單引號或成對的雙引號來建立文字，多數的時候使用單引號或者雙引號不會有分別

```
In [27]: movie_title = "Avengers: Endgame"  
print(movie_title)  
print(type(movie_title))
```

```
Avengers: Endgame  
<class 'str'>
```

```
In [28]: movie_title = 'Avengers: Endgame'  
print(movie_title)  
print(type(movie_title))
```

```
Avengers: Endgame  
<class 'str'>
```

亦可以使用六個雙引號來建立長文字，像是一個有換行的文章段落

```
In [29]: story_line = """
After the devastating events of Avengers: Infinity War (2018), the universe is in ruins.

With the help of remaining allies, the Avengers assemble once more in order to undo Than
os' actions and restore order to the universe.
"""
print(story_line)
print(type(story_line))
```

After the devastating events of Avengers: Infinity War (2018), the universe is in ruins.

With the help of remaining allies, the Avengers assemble once more in order to undo Thanos' actions and restore order to the universe.

<class 'str'>

少數的時候使用雙引號或單引號會有差異

```
In [30]: shaq = "Shaquille O'Neal"
```

```
In [31]: shaq = 'Shaquille O'Neal'
```

```
File "<ipython-input-31-6589a1dcd440>", line 1
```

```
    shaq = 'Shaquille O'Neal'
```

```
                        ^
```

```
SyntaxError: invalid syntax
```

Single quotes 與 Apostrophe 是不同的

必須使用 \（跳脫字元）來標註 Apostrophe。

```
In [32]: shaq = 'Shaquille O\'Neal'
```


隨堂練習：I'm lovin' it!

印出 I'm lovin' it!

```
In [34]: print(mcd)
```

```
I'm lovin' it!
```

隨堂練習：What did Ross say?

Let's put aside the fact that you "accidentally" pick up my grandmother's ring.

```
In [36]: print(ross_said)
```

Let's put aside the fact that you "accidentally" pick up my grandmother's ring.

文字嵌入 String interpolation

印出不同物件的值 "{}".format(data)

```
In [37]: print("復仇者聯盟：終局之戰的評分為：8.8")  
         print("復仇者聯盟：無限之戰的評分為：8.5")
```

復仇者聯盟：終局之戰的評分為：8.8
復仇者聯盟：無限之戰的評分為：8.5

```
In [38]: print("{}的評分為：{}".format("復仇者聯盟：終局之戰", 8.8))  
         print("{}的評分為：{}".format("復仇者聯盟：無限之戰", 8.5))
```

復仇者聯盟：終局之戰的評分為：8.8
復仇者聯盟：無限之戰的評分為：8.5

調整數值呈現的格式

- `{:.0f}` 整數
- `{:.2f}` 小數位數兩位的浮點數
- `{:,}` 千分位逗點

In [39]:

```
pi = 3.14159
```

```
print("{:.0f}".format(pi))
```

```
print("{:.2f}".format(pi))
```

3

3.14

```
In [40]: n_feedbacks = 448360  
print("{:,}".format(n_feedbacks))
```

448,360

搭配 `input()` 函數來索取使用者輸入

In [41]: `help(input)`

Help on method raw_input in module ipykernel.kernelbase:

`raw_input(prompt='') method of ipykernel.ipkernel.IPythonKernel instance`
Forward `raw_input` to frontends

Raises

`StdinNotImplementedError` if active frontend doesn't support `stdin`.


```
In [42]: movie_title = input("請輸入電影名稱：")  
movie_rating = input("請輸入電影評分：")  
print("{}的評分為：{}".format(movie_title, movie_rating))
```

請輸入電影名稱：Avengers: Endgame

請輸入電影評分：8.8

Avengers: Endgame的評分為：8.8

隨堂練習：讓使用者輸入城市名稱與天氣，並印出
「我在000，天氣000」

文字支援 + 和 * 這兩個運算符號

```
In [44]: print("55" + "66")  
         print("5"*6)
```

```
5566  
555555
```

布林

布林只有 True 與 False 這兩個值

In [45]:

```
print(type(True))  
print(type(False))
```

```
<class 'bool'>  
<class 'bool'>
```

比較符號（ Comparison Operators ）

- == 、 != ：等於以及不等於
- > 、 >= 、 < 、 <= ：大於、大於等於、小於以及小於等於

電影評分是否高於 8.0

```
In [46]: movie_rating = 8.8  
print(movie_rating > 8.0)  
print(type(movie_rating > 8.0))
```

```
True  
<class 'bool'>
```

布林運算符號（ Boolean Operators ）

- not ：非
- is 、 is not ：是否為相同的值與類型
- and 、 or ：交集與聯集

```
In [48]: print(not True)  
          print(not False)
```

```
False  
True
```



```
In [49]: print(5566 is 5566.0)
print(5566 is not "5566")
```

False

True

In [50]:

```
print(True and True)
print(True and False)
print(False and False)
print(True or True)
print(True or False)
print(False or False)
```

```
True
False
False
True
True
False
```

資料型態轉換

回憶一下這個隨堂練習

```
In [51]: city = input("請輸入您所在的城市：")  
         weather = input("請輸入現在的天氣：")  
         print("我在{}, 天氣{}".format(city, weather))
```

請輸入您所在的城市：台北
請輸入現在的天氣：晴
我在台北，天氣晴

加入氣溫的資訊

```
In [52]: city = input("請輸入您所在的城市：")
weather = input("請輸入現在的天氣：")
tempc = input("請輸入現在的攝氏氣溫：")
tempf = tempc*9/5 + 32

print("我在{}, 天氣{}, 攝氏 {} 度, 華氏 {} 度".format(city, weather, tempc, tempf))
```

請輸入您所在的城市：台北
請輸入現在的天氣：晴
請輸入現在的攝氏氣溫：25

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-52-aed3dd2b726d> in <module>
      2 weather = input("請輸入現在的天氣：")
      3 tempc = input("請輸入現在的攝氏氣溫：")
----> 4 tempf = tempc*9/5 + 32
      5
      6 print("我在{}, 天氣{}, 攝氏 {} 度, 華氏 {} 度".format(city, weather, tempc, tem
pf))
```

TypeError: unsupported operand type(s) for /: 'str' and 'int'

檢查 **tempc** 的類別

```
In [53]: print(type(tempc))
```

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

透過 **input()** 函數獲得的資料類別是文字

多數的數值運算符號（Arithmetic Operators）不支援文字類別

使用與資料類別相同名稱的函數轉換

- `int()`：轉換為整數類別
- `float()`：轉換為浮點數類別
- `str()`：轉換為文字類別
- `bool()`：轉換為布林類別

In [54]:

```
city = input("請輸入您所在的城市：")
weather = input("請輸入現在的天氣：")
tempc = input("請輸入現在的攝氏氣溫：")
tempc = float(tempc) # 轉換為浮點數類別
tempf = tempc*9/5 + 32

print("我在{}, 天氣{}, 攝氏 {} 度, 華氏 {} 度".format(city, weather, tempc, tempf))
```

請輸入您所在的城市：台北

請輸入現在的天氣：晴

請輸入現在的攝氏氣溫：25

我在台北，天氣晴，攝氏 25.0 度，華氏 77.0 度

作業

球員 BMI 與過重判斷

- 以 `input()` 函數請使用者輸入球員姓名、身高與體重
- 計算球員的 BMI 並且印出
- 超過 30 代表過重，請印出判斷結果

執行範例

```
## 請輸入球員姓名：俠客歐尼爾  
## 請輸入球員身高（cm）：216  
## 請輸入球員體重（kg）：147
```

```
In [57]: print("{}的身體質量指數為：{:.2f}".format(player_name, player_bmi))  
print("{}是否過重：{}".format(player_name, player_bmi > 30))
```

俠客歐尼爾的身體質量指數為：31.51
俠客歐尼爾是否過重：True