成為初級資料分析師 | Python 與資料科學應用

Python 程式設計常用技巧

郭耀仁

Readability counts. The Zen of Python, Tim Peters

大綱

- 彈性參數
- 匿名函數
- 迭代函數（Iterators）
- List Comprehensions
- Generators
- 物件導向
- 常用文字方法

彈性參數

有時我們的函數不確定使用者會想輸入幾個參數

- `*args` : for list-like arguments
- `**kwargs`: for dict-like arguments

In [1]:
```
def get_fahrenheit(c):
    return c*9/5 + 32

get_fahrenheit(18)
```

Out[1]:
```
64.4
```

In [2]:

```
def get_fahrenheits(*args):
    fahrenheits = []
    for c in args:
        fahrenheits.append(c*9/5 + 32)
    return fahrenheits

print(get_fahrenheits(18))
print(get_fahrenheits(18, 20))
print(get_fahrenheits(18, 20, 22))
```

```
[64.4]
[64.4, 68.0]
[64.4, 68.0, 71.6]
```

In [3]:

```
def get_city_fahrenheit(city, c):
    city_f = {
        city: c*9/5 + 32
    }
    return city_f

get_city_fahrenheit("Taipei", 18)
```

Out[3]:

```
{'Taipei': 64.4}
```

In [4]:

```
def get_city_fahrenheits(**kwargs):
    city_f = {}
    for k, v in kwargs.items():
        v = v*9/5 + 32
        city_f[k] = v
    return city_f

print(get_city_fahrenheits(Taipei=18))
print(get_city_fahrenheits(Taipei=18, London=20))
print(get_city_fahrenheits(Taipei=18, London=20, Japan=22))
```

```
{'Taipei': 64.4}
{'Taipei': 64.4, 'London': 68.0}
{'Taipei': 64.4, 'London': 68.0, 'Japan': 71.6}
```

隨堂練習：寫一個函數 `get_std(*args)` 回傳 `*args` 所組成之數列的樣本標準差

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

https://en.wikipedia.org/wiki/Standard_deviation (https://en.wikipedia.org/wiki/Standard_deviation)

In [6]:
```
print(get_std(1, 3, 5, 7, 9))
print(get_std(3, 4, 5, 6, 7))
print(get_std(3))
```

```
3.1622776601683795
1.5811388300841898
Please input at least 2 numbers.
```

匿名函數

有些時候我們需要比 def 更簡潔的語法來定義函數

In [7]:
```
def squared(x):
    return x**2

squared(2)
```

Out[7]:
```
4
```

匿名函數又稱為 lambda 函數

```
FUNCTION_NAME = lambda arg0, arg1, ...: USING arg0, arg1
```

In [8]:
```
squared = lambda x: x**2

squared(2)
```

Out[8]:
```
4
```

In [9]:
```
my_abs = lambda x: -x if x < 0 else x

print(my_abs(-2))
print(my_abs(2))
```

```
2
2
```

使用迭代函數（Iterators）時候會產生匿名函數需求

迭代函數（Iterators）

常與匿名函數一起出現的迭代函數

- `map()`
- `filter()`

In [10]:
```
def get_fahrenheit(c):
    return c*9/5 + 32

temp_c = [18, 20, 22]
temp_f = map(get_fahrenheit, temp_c)
list(temp_f)
```

Out[10]:
```
[64.4, 68.0, 71.6]
```

In [11]:
```
# map()
temp_c = [18, 20, 22]
temp_f = map(lambda x: x*9/5 + 32, temp_c)
list(temp_f)
```

Out[11]:
```
[64.4, 68.0, 71.6]
```

In [12]:
```
# filter()
temp_c = [-10, 18, 20, -5, -3]
below_zero = filter(lambda x: x < 0, temp_c)
list(below_zero)
```

Out[12]:
```
[-10, -5, -3]
```

其他常用迭代函數

- `enumerate()`：同時取用一個 iterable 中的 index 與 value
- `zip()`：同時取用多個 iterables 中的 values

In [13]:
```
# enumerate()：同時取用一個 iterable 中的 index 與 value
the_avenger_movies = ["The Avengers", "Avengers: Age of Ultron", "Avengers: Infini
ty War", "Avengers: Endgame"]
for i, val in enumerate(the_avenger_movies):
    print("復仇者聯盟第{}集：{}".format(i+1, val))
```

```
復仇者聯盟第1集：The Avengers
復仇者聯盟第2集：Avengers: Age of Ultron
復仇者聯盟第3集：Avengers: Infinity War
復仇者聯盟第4集：Avengers: Endgame
```

In [14]:

```
# zip(): 同時取用多個 iterables 中的 values
release_years = [2012, 2015, 2018, 2019]
the_avenger_movies = ["The Avengers", "Avengers: Age of Ultron", "Avengers: Infini
ty War", "Avengers: Endgame"]
for y, movie in zip(release_years, the_avenger_movies):
    print("{} 上映年份 {}".format(movie, y))
```

```
The Avengers 上映年份 2012
Avengers: Age of Ultron 上映年份 2015
Avengers: Infinity War 上映年份 2018
Avengers: Endgame 上映年份 2019
```

List Comprehensions

將使用 loop 構建 list 壓縮為簡潔單行的方法

In [15]:
```
# loop construction
squared_list = []
for i in range(10):
    squared_list.append(i**2)
print(squared_list)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [16]:
```
# list comprehension
squared_list = [i**2 for i in range(10)]
print(squared_list)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [17]:

```
# list comprehension with if
even_numbers = [i for i in range(10) if i % 2 == 0]
print(even_numbers)
```

```
[0, 2, 4, 6, 8]
```

In [18]:
```
# list comprehension with if-else
is_even_numbers = [True if i % 2 == 0 else False for i in range(10)]
print(is_even_numbers)
```

```
[True, False, True, False, True, False, True, False, True, False]
```

隨堂練習：將公里的距離都轉換成英里

1 kilometer = 0.62137 mile

In [19]:
```
kilometers = [1.6, 3, 5, 10, 21.095, 42.195]
```

In [21]:
```
print(miles)
```

```
[0.994192, 1.86411, 3.1068499999999997, 6.213699999999999, 13.10780015, 26.218
70715]
```

Generators

Generators 是用來產生資料的物件

常見的 generators

- `map()`
- `filter()`
- `enumerate()`
- `zip()`

In [22]:
```
# map()
temp_c = [18, 20, 22]
temp_f = map(lambda x: x*9/5 + 32, temp_c)
print(type(temp_f))
print(temp_f)
```

```
<class 'map'>
<map object at 0x1048b2240>
```

In [23]:
```
list(temp_f)
```

Out[23]:
```
[64.4, 68.0, 71.6]
```

In [24]:
```
list(temp_f)
```

Out[24]:
```
[]
```

In [25]:
```
# filter()
temp_c = [-10, 18, 20, -5, -3]
below_zero = filter(lambda x: x < 0, temp_c)
print(type(below_zero))
print(below_zero)
```

```
<class 'filter'>
<filter object at 0x1048b24e0>
```

In [26]:
```
list(below_zero)
```

Out[26]:
```
[-10, -5, -3]
```

In [27]:
```
list(below_zero)
```

Out[27]:
```
[]
```

隨堂練習：使用 map() 將公里的距離都轉換成英里

1 kilometer = 0.62137 mile

In [28]:
```
kilometers = [1.6, 3, 5, 10, 21.095, 42.195]
```

In [30]:
```
print(miles)
print(list(miles))
```

```
<generator object <genexpr> at 0x1048a7a20>
[0.994192, 1.86411, 3.1068499999999997, 6.213699999999999, 13.10780015, 26.218
70715]
```

In [31]:
```
# enumerate()
the_avenger_movies = ["The Avengers", "Avengers: Age of Ultron", "Avengers: Infini
ty War", "Avengers: Endgame"]
enumerate_generator = enumerate(the_avenger_movies)
print(type(enumerate_generator))
print(enumerate_generator)
```

```
<class 'enumerate'>
<enumerate object at 0x10488b828>
```

In [32]:
```
list(enumerate_generator)
```

Out[32]:
```
[(0, 'The Avengers'),
 (1, 'Avengers: Age of Ultron'),
 (2, 'Avengers: Infinity War'),
 (3, 'Avengers: Endgame')]
```

In [33]:
```
list(enumerate_generator)
```

Out[33]:
```
[]
```

In [34]:
```
# zip()
release_years = [2012, 2015, 2018, 2019]
the_avenger_movies = ["The Avengers", "Avengers: Age of Ultron", "Avengers: Infini
ty War", "Avengers: Endgame"]
zip_generator = zip(the_avenger_movies)
print(type(zip_generator))
print(zip_generator)
```

```
<class 'zip'>
<zip object at 0x1048b8148>
```

In [35]:
```
list(zip_generator)
```

Out[35]:
```
[('The Avengers',),
 ('Avengers: Age of Ultron',),
 ('Avengers: Infinity War',),
 ('Avengers: Endgame',)]
```

In [36]:
```
list(zip_generator)
```

Out[36]:
```
[]
```

物件導向

注意物件導向的三個應用面

- 初始化
- 靜態的屬性
- 動態的方法

In [37]:

```
class Movie:
    def __init__(self, rating, movie_time):
        self._rating = rating
        self._movie_time = movie_time
        self._genre = []

    def get_rating(self):
        return self._rating

    def get_movie_time(self):
        return self._movie_time

    def get_genre(self):
        return self._genre

    def add_genre(self, genre):
        self._genre.append(genre)
        return True
```

In [38]:
```
avengers_endgame = Movie(8.8, '3h 1min') # 初始化
# 靜態的屬性
print(avengers_endgame._rating)
print(avengers_endgame._movie_time)
print(avengers_endgame._genre)
```

```
8.8
3h 1min
[]
```

In [39]:
```
# 動態的方法
print(avengers_endgame.get_rating())
print(avengers_endgame.get_movie_time())
print(avengers_endgame.get_genre())
avengers_endgame.add_genre("Action")
avengers_endgame.add_genre("Adventure")
avengers_endgame.add_genre("Sci-Fi")
print(avengers_endgame.get_genre())
```

```
8.8
3h 1min
[]
['Action', 'Adventure', 'Sci-Fi']
```

常用文字方法

格式化文字

.format()

In [40]:
```
pi = 3.14159
print("圓周率的值為: {}".format(pi))
```

```
圓周率的值為: 3.14159
```

In [41]:

```
pi_str = "圓周率"
pi = 3.14159

print("{}取兩位小數為: {:.2f}".format(pi_str, pi))
print("{}整數部分是 {:.0f}".format(pi_str, pi))
```

```
圓周率取兩位小數為: 3.14
圓周率整數部分是 3
```

更改文字大小寫的方法

- `.title()`
- `.upper()`
- `.lower()`

In [42]:
```
use_the_force = "Luke, use the Force!"

print(use_the_force.title())
print(use_the_force.upper())
print(use_the_force.lower())
```

```
Luke, Use The Force!
LUKE, USE THE FORCE!
luke, use the force!
```

去除多餘空白、換行符號的方法

- `.rstrip()`
- `.lstrip()`
- `.strip()`

In [43]:
```
use_the_force = """

Luke, use the Force!

"""

use_the_force
```

Out[43]:
```
'\n     \nLuke, use the Force!\n     \n'
```

In [44]:
```
print(use_the_force.rstrip())
print(use_the_force.lstrip())
print(use_the_force.strip())
```

```
Luke, use the Force!
Luke, use the Force!


Luke, use the Force!
```

取代文字的方法

`.replace()`

In [45]:
```
skywalker = "Anakin Skywalker"
print(skywalker)
print(skywalker.replace("Anakin", "Luke"))
```

```
Anakin Skywalker
Luke Skywalker
```

切割文字的方法

.split()

In [46]:
```
use_the_force = "Luke, use the Force!"
print(use_the_force.split())
print(use_the_force.split(","))
```

```
['Luke,', 'use', 'the', 'Force!']
['Luke', ' use the Force!']
```