Python
Map 映射、Filter篩選

臺北科技大學資訊工程系

map

- map(function, iterable1 [, iterable2])
  - 輸入 function 和不定個數的 iterable (list, tuple, …)
  - 將iterable當參數傳入 function，一一回傳function執行結果
    - 對傳入的iterable每一個元素進行對映，回傳新對映後iterable
  - 使用 map() 可縮短程式碼和加速執行，map後得到map object
  - map無法處理
    - iterable 長度不一致
    - 對應位置運算元型別不一致

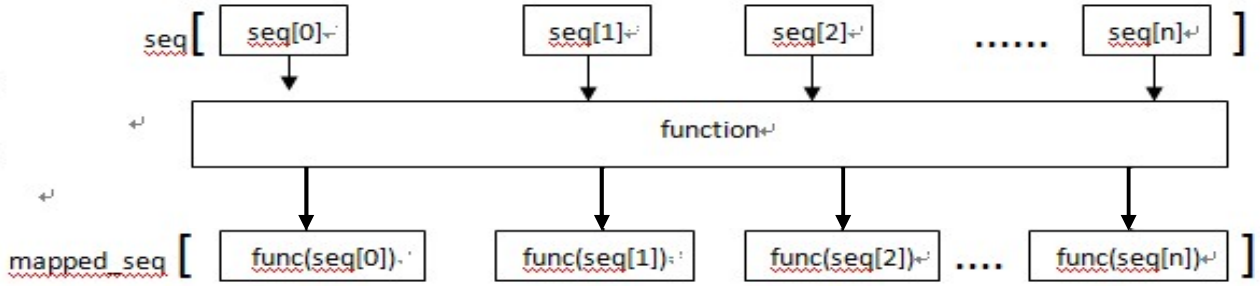
```
a=list(map(lambda x , y : x ** y, [2,4,6],[3,2,1]))
print(a)
```

```
scores = [50, 52, 54, 56, 58, 60]
new_scores = map(lambda x: 60 if 55 <= x < 60 else x, scores)
print(list(new_scores))
# 顯示處理後成績：[50, 52, 54, 60, 60, 60]
```

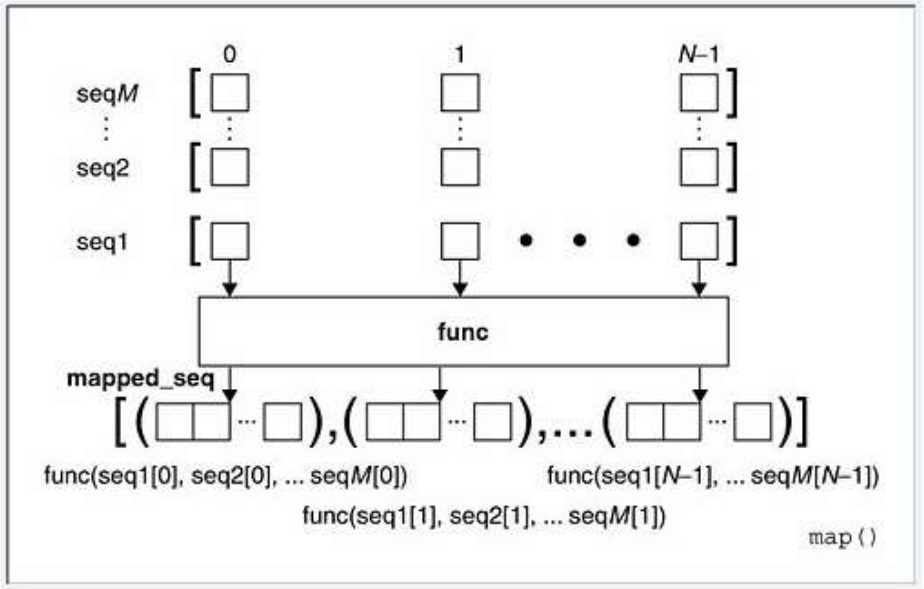
```
def multiply2(x, y):
    return x * y

a=list(map(multiply2, [2, 4, 6], [3, 2, 1]))
print(a)        # Output [6, 8, 6]
```

map

- 當seq/iterable只有一個時，map(func, seq)以函式func處理seq每個元素，得到一個新的seq



- 當seq多於一個時，map可平行處理每個seq



Exercise

- 利用map()函式，輸出？

```
def adder(x,y,z):
    return x+y+z

def multiple2(x):
    return x*x

list1 = [1,3,5,7,9]
for x in map(multiple2,list1):
    print(x, end=' ')
print([x for x in map(multiple2,list1)])
print([multiple2(x) for x in list1])
list1 = [1,3,5,7,9]
list2 = [2,4,6,8,10]
list3 = [100,100,100,100,100]
print([x for x in map(adder,list1,list2,list3)])
my_list = [1, 2, 3]
a=list(map( lambda i: i * i, my_list ))
print(a)
```

Output

```
1 9 25 49 81
[
[
[
[
```

Exercise

- 利用map()，將 'asDfA13' 字母從大寫轉小寫，小寫轉大寫

```
def u2l_and_l2u (s):
  return s.upper() if [          ]() else s.[          ]

a = list(map(u2l_and_l2u,'asDfA13'))
ms=''
b=ms.join(a)
print(b)
c = ms.join([str(elem) for elem in a])
print(c)
for c in a:
    ms=ms+c
print(ms)
```

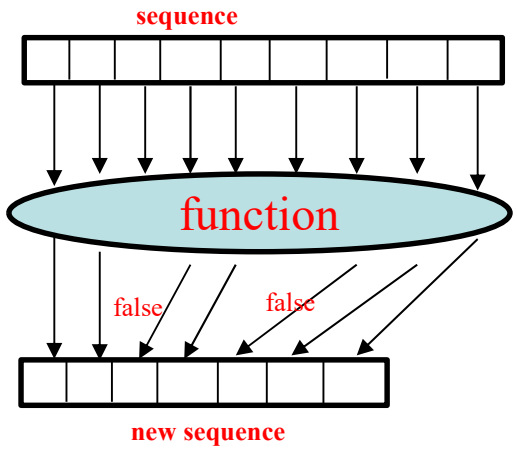
```
ASdFa13
ASdFa13
ASdFa13
```

filter

- filter(function, sequence)
  - sequence的每個元素當參數傳給function進行True/False判斷
  - 過濾sequence中的元素，回傳符合條件之元素組成的新 sequence (將回傳值為 True 的項目組成一個 iterator)

```
# a = list(filter(lambda x: x % 2 == 1, range(10)))
def is_odd(n):
    return n % 2 == 1
a = list(filter(is_odd, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]))
print(a)
```

```
[1, 3, 5, 7, 9]
```




```
import math
def is_sqr(x):
    return math.sqrt(x) % 1 == 0

a = list(filter(is_sqr, range(1, 101)))
print(a)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

```
scores = [90, 50, 80, 40, 100]
fail_scores = filter(lambda x: True if x < 60 else False, scores)
# 回傳 <filter object at 0x7f17c658a630>
print(list(fail_scores))  #[50, 40]
```

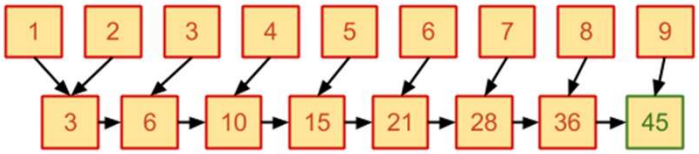
Exercise

- scores=[['John', 90, 80, 90],['Bob', 50, 70, 40], ['Mary', 100, 90, 85],['Tom', 80, 90, 70]]
  - 使用 filter, lambda, 輸出平均及格的 名字、平均

```
scores=[['John', 90, 80, 90],['Bob', 50, 70, 40], ['Mary', 100, 90, 85],['Tom', 80, 90, 70]]

data = filter(lambda                                  ) else False, scores)
k = list(data)
for x in k:
    print(x[0], (x[1]+x[2]+x[3])//3)
print(k)

data_s = sorted(k, key = lambda                  )
for x in data_s:
    print(x[0], (x[1]+x[2]+x[3])//3)
```

reduce()

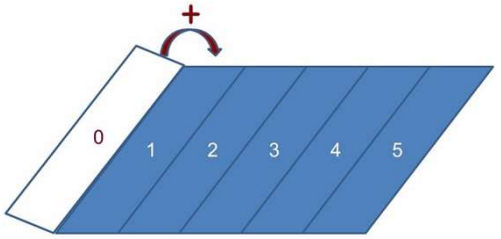
- reduce (f, seq[, init()])
  - 每次迭代都將上次迭代結果與下一個seq元素一同傳入二元(binary)函式(具有兩個參數的函式)中執行
  - init是optional，若指定，則當第一次迭代第一個參數，若無則取seq第一個元素
  - 對list的每個元素反覆呼叫函式f，回傳最終結果值




```
import functools as ft
a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
def fn(x, y):
    return x+y
print(ft.reduce(fn, a))     #45
```

```
import functools as ft
a=ft.reduce(lambda sum, elem: sum + elem, [1, 2, 3, 4, 5], 0)
print(a)
print (ft.reduce(lambda x, y: x * y, range(1, 6),2))
```

```
15
240
```



END