

1.11 流程控制 for

你將學習到

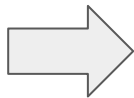
1. for 是什麼
2. for 用作數據處理
3. for 用作重複動作
4. continue, break 語法
5. range() 函數
6. enumerate() 函數

for 迴圈做重複性動作

for 迴圈用作對 list, tuple, dict, string (統稱 Iterables) 的元素做重複性的動作, 用 for .. in .. 定義。例如以下程式做的是把 List 中每個數值乘上 100。

手動把每個元素乘100, 好累贅

```
a = [0.1, 0.5, 0.25, 0.8, 0.9]
print( a[0] * 100 )
print( a[1] * 100 )
print( a[2] * 100 )
print( a[3] * 100 )
print( a[4] * 100 )
```



用 for 迴圈實現, 大幅簡化程式 !

```
a = [0.1, 0.5, 0.25, 0.8, 0.9]
for i in a:
    print( i * 100 )
```

i 是一個自定義的變數, 變數名可任改,
用來儲存 List 中每個元素的數值
0.1, 0.5, 0.25 ...

for 迴圈運作流程

```
for i in [0.1, 0.5, 0.25, 0.8, 0.9]:  
    print( i*100 )
```

i 是一個自定義的變數，變數名可任
改，用來儲存 List 中每個元素的數
值 0.1, 0.5, 0.25 ...

等同於

```
i = 0.1  
print( i*100 )  
  
i = 0.5  
print( i*100 )  
  
i = 0.25  
print( i*100 )  
  
i = 0.8  
print( i*100 )  
  
i = 0.9  
print( i*100 )
```

for 迴圈運作流程

我們今次實作股票採集程式時，會希望程式一次過採集數個股票現價 (eg. 00001 - 00003)，這可以用以下 for-loop 實現。

```
for share in ["00001", "00002", "00003"]:  
    print( share )
```

等同於

```
share = "00001"  
print( share )  
  
share = "00002"  
print( share )  
  
share = "00003"  
print( share )
```

數據處理常用的迴圈方法 (list)

```
a = [0.1, 0.5, 0.25, 0.8, 0.9]    # 要轉換成百分比的數值

percent = []                       # 定義空 list, 儲存計算結果

for i in a:

    percent.append( i * 100 )      # 把 i 乘上100, 然後儲存到 percent

# percent = [10.0, 50.0, 25.0, 80.0, 90.0]
```

重複動作迴圈

有時候我們無 List, 只想指定電腦執行動作 n 次。
這時你可以用 range() 函數產生一個包含 n 個元素的
List (0, 1, 2, ... n)。

```
for i in range(5):
```

```
    action()
```

```
# 以上動作會執行 5 次
```

重複動作迴圈

`range()` 函數會產生一個包含 n 個元素的 List $(0, 1, 2, \dots, n)$ 。

```
for i in range(5):  
    action()
```

等同於

```
for i in [0, 1, 2, 3, 4]:  
    action()
```


range() 函數

你也可以指定產生出來的序列，用法是：

range(stop) 或 range(start, stop, step)

```
range(5)                # 輸出 range(0, 5), 無法看到具體數值!
list( range(5) )         # 轉換成 List 以觀看數值, 輸出 [0, 1, 2, 3, 4]
list( range(1, 6) )      # 轉換成 List 以觀看數值, 輸出 [1, 2, 3, 4, 5]

# 注意 range(5) 等同於 range(0, 5)
```

跳出迴圈 break

程式執行到 break, 會立刻跳出迴圈。

```
for i in [0, 1, 2, 3, 4]:  
    if i == 2:  
        break  
    print( i )
```

輸出:

0

1

返回迴圈頂部 `continue`

程式執行到 `continue`, 會立刻返回迴圈頂部, 執行下一次。

```
for i in [0, 1, 2, 3, 4]:  
    if i == 2:  
        continue  
    print( i )
```

輸出:

0
1
3
4

enumerate() 函數

enumerate() 可以讓我們取得 List 元素的 index。

```
a = ["a","b","c"]  
for i, j in enumerate(a):  
    print( i )  
    print( j )  
# i 會儲存元素 index, j 會儲存元素。
```

輸出：

0

a

1

b

2

c

enumerate() 函數

enumerate() 寫法

```
a = ["a","b","c"]  
for i, j in enumerate(a):  
    print( str(i) + " : " + j )
```

輸出:

```
0 : a  
1 : b  
2 : c
```

不用 enumerate()

```
a = ["a","b","c"]  
for i in range(0, len(a)):  
    print( str(i) + " : " + a[i] )
```

輸出:

```
0 : a  
1 : b  
2 : c
```

額外資料 (選讀)

以下為額外資料供同學學習更多知識，課堂上無需用到。

for 對 string 的應用

```
a = "abc"
```

```
for i in a:
```

```
    print( i )    # for 用於 string 時, i 會等於每個字母
```

```
# 輸出:
```

```
a
```

```
b
```

```
c
```

for 對 dict 的應用

```
a = { "id": "1", "name": "bensong" }
```

```
for i in a:
```

```
    print( i )
```

```
    print( a[i] )
```

for 用於 dict 時, i 會儲存鍵 (key), 數值則需用 a[i] 取得

輸出:

id

1

name

bensong

for 對 list of tuples 的應用

```
a = [ (0, "a"), (1, "b"), (2, "c") ]
```

```
for i in a:
```

```
    print( i )
```

```
    # for 用於 list of tuples 時, i 會等於每個 Tuple
```

輸出:

(0, "a")

(1, "b")

(2, "c")

for 對 list of tuples 的應用

```
a = [ (0, "a"), (1, "b"), (2, "c") ]  
for i in a:  
    print( str(i[0]) + ", " + i[1] )  
    # 我們需用 i[0], i[1]  
    # 取得 tuple 內的元素。
```

輸出:

```
0, "a"  
1, "b"  
2, "c"
```

小提示:

i[0] 會等於 0, 1, 2, 型別是整數, i[1] 會等於 "a", "b", "c", 型別是字串。

但由於整數無法和字串直接相加, 因此我們要用 str() 函數把 i[0] 轉換為字串, 才能夠把 i[0]、", "、i[1] 三個資料相加打印出屏幕。

for 對 list of tuples 的應用

```
a = [ (0, "a"), (1, "b"), (2, "c") ]
```

```
for i, j in a:    # 原來我們可以用兩個變數儲存 Tuple 內每個元素！
```

```
    print( str(i) + ", " + j )
```

```
    # for 用於 list of tuples 時, i, j 會順序等於 Tuple 內每個元素
```

```
    # 注意這個例子中, i 會等於 0, 1, 2, 型別是整數, 因此我們要用
```

```
    # str() 函數把它轉換為字串。
```

輸出:

0, a

1, b

2, c

for 對 list of tuples 的應用

```
a = [ (0, "a", "a1"), (1, "b", "b1"), (2, "c", "c1") ]
```

```
for i, j, k in a:      # 用三個變數儲存 Tuple 內每個元素
```

```
    print( str(i) + ", " + j + ", " + k )
```

```
# for 用於 list of tuples 時, i, j, k 會順序等於 Tuple 內每個元素
```

```
# 注意這個例子中, i 會等於 0, 1, 2, 型別是整數, 因此我們要用
```

```
# str() 函數把它轉換為字串。
```

輸出:

0, a, a1

1, b, b1

2, c, c1

enumerate() 函數

有時候你會想在 for-loop 內取得 list 的 index, 這時你可以用 enumerate() 產生一個附有 index 的 List of tuples。

```
list( enumerate(["a","b","c"]) )
```

```
# 輸出 [ (0, "a"), (1, "b"), (2, "c") ]
```

enumerate() 函數

enumerate() 寫法

```
a = ["a","b","c"]  
for i, j in enumerate(a):  
    print( str(i) + " : " + j )
```

輸出:

```
0 : a  
1 : b  
2 : c
```

不用 enumerate()

```
a = ["a","b","c"]  
for i in range(0, len(a)):  
    print( str(i) + " : " + a[i] )
```

輸出:

```
0 : a  
1 : b  
2 : c
```

練習

1. 定義變數 `a` 賦值為 `[0.1, 0.2, 0.3, 0.4, 0.5]`
2. 定義變數 `b` 賦值為 `[]`
3. 利用 `for ... in ...`, 把變數 `a` 每個元素乘100, 並加入變數 `b`。
4. 重新把變數 `b` 賦值為 `[]`,
然後利用 `for ... in enumerate(...)` 和 `if`, 把變數 `a` 每個元素乘100, 並利用 `if` 捨棄 `a[1]`, 加到變數 `b`
程式運行後, `b = [10.0, 30.0, 40.0, 50.0]`

答案

1. `a = [0.1, 0.2, 0.3, 0.4, 0.5]`

2. `b = []`

3. `for i in a:`

`b.append(i*100)`

4. `b = []`

`for i, j in enumerate(a):`

`if i != 1:`

`b.append(j * 100)`

學習回顧

1. for 是什麼
2. for 用作數據處理
3. for 用作重複動作
4. continue, break 語法
5. range() 函數
6. enumerate() 函數