

循環 For-Loop

Python 編程課程 第 2 課

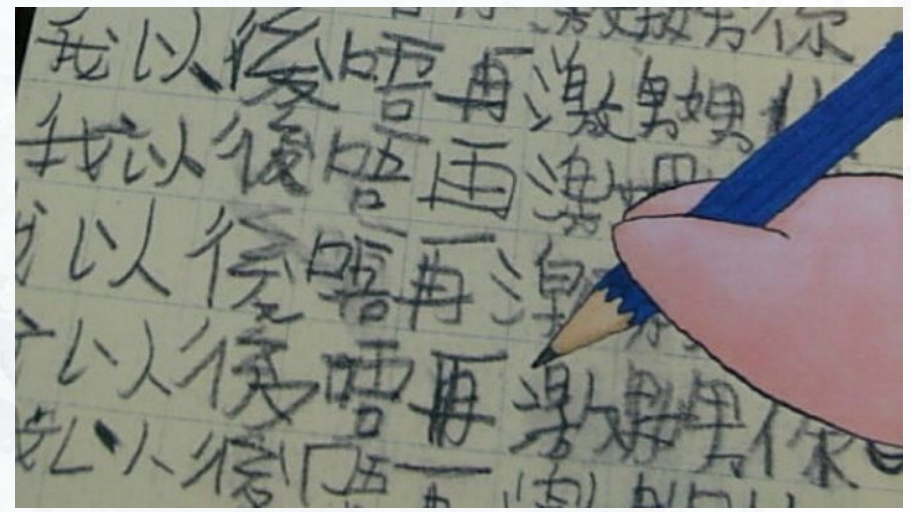
- For 循環
- 循環結構(縮進)
- 嵌套式循環 Nested loop

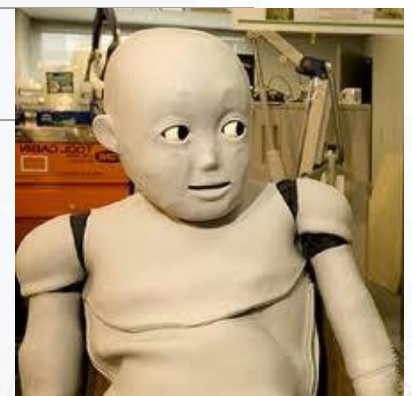
日常例子 - 罰抄

你被老師罰抄，一百句「我以後不敢亂拋垃圾」

但是.....

你的老師不知道你其實是機械人

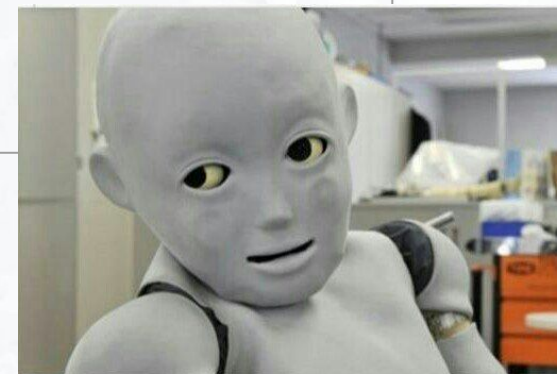


你的程式碼？	輸出
<pre>print("我以後不敢亂拋垃圾") print("我以後不敢亂拋垃圾") print("我以後不敢亂拋垃圾") print("我以後不敢亂拋垃圾") print("我以後不敢亂拋垃圾")</pre>	<p>我以後不敢亂拋垃圾 我以後不敢亂拋垃圾 我以後不敢亂拋垃圾 我以後不敢亂拋垃圾 我以後不敢亂拋垃圾</p> 

日常例子 - 罰抄

你想善用機械人的優勢

你的程式碼？	輸出
唔該電腦 幫我重覆一百次print(“我以後不敢亂拋垃圾”) 感謝你 🙏	我以後不敢亂拋垃圾 我以後不敢亂拋垃圾 我以後不敢亂拋垃圾 我以後不敢亂拋垃圾 我以後不敢亂拋垃圾



For-loop 循環

循環方式(一) - range()

如果你確定循環次數，可以使用函數 range() 來生成數字列表。

數字列表的頭尾值可用作 for 循環的範圍。

```
for number in range(1, 6):  
    print(number)
```

從這個值開始

在這個值之前停止

輸出

```
1  
2  
3  
4  
5
```

循環方式(一) - range()

如果只傳入一個參數，會生成一個**從 0 開始**的數字序列。

```
for number in range(6):  
    print(number)
```

從 0 開始

在這個值之前
停止

輸出

0
1
2
3
4
5



請記住，從 0 開始是 Python 中的慣例。

循環方式(二) - 列出項目

我們可以使用for循環來列出項目。

for循環的組成部分：

- for 關鍵字
- 臨時變量名稱
- in 關鍵字
- 列表
- : 冒號
- 縮進的for循環塊

程式碼

```
cooks = ['peter', 'paul', 'mary']  
for cook in cooks:  
    print(cook)
```

for循環

輸出

```
peter  
paul  
mary
```



臨時變量名稱：你可以取任何名字。通常是列表名稱的單數形式、“item”、字母 i。
方便易認即可。

循環方式(二) - 列出項目

程式碼

```
cooks = ['peter', 'paul', 'mary']  
for cook in cooks:  
    print(cook)
```

for循環



輸出

peter

◀ 第一次循環

paul

◀ 第二次循環

mary

◀ 第三次循環

在for循環中缺少冒號：

忘記在寫 for 循環時放置一個冒號：

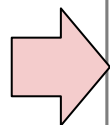
這會導致語法錯誤

```
cooks = ['peter', 'paul', 'mary']  
for cook in cooks  
    print(f"{cook.title()} is a good cook!")
```

輸出

```
File "/Users/LP/___JetDrive/temp/testing.py", line 2  
    for cook in cooks  
        ^  
SyntaxError: invalid syntax
```

縮進／縮排



```
cooks = ['peter', 'paul', 'mary']  
for cook in cooks: for循環  
    print(f"{cook.title()} is a good cook!")  
    print(f"And {cook.title()} works in HK!\n")  
print(f"We got {len(cooks)} nice cooks in HK!")
```

i 縮進：Python中用以識別**同一段碼塊**的特徵

做法：制表符（Tab）、四／兩個空格

	空白塊	佔用空格
制表符（Tab）		4
四個空格 （spacebar）	<div><div></div><div></div><div></div><div></div></div>	4

提示：在Python中將縮進設置為4個空格是一種易辨識的良好習慣

縮進錯誤

正如縮進是 Python 中解我們的程序結構的關鍵，有不正確的縮進時，Python 會出現 **Indentation Error**。

當 Indentation Error 發生時，它會告訴你問題出在哪裡。

```
cooks = ['peter', 'paul', 'mary']  
for cook in cooks:  
    print(f"{cook.title()} is a good cook!")
```

輸出

```
File "/Users/LP/___JetDrive/temp/testing.py", line 3  
    print(f"{cook.title()} is a good cook!")  
    ^  
IndentationError: expected an indented block
```

for 循環之後的代碼

我們嘗試在 for 循環之後編寫新print代碼。

新代碼行不會在 for 循環中運行，只會在循環後**運行一次**。

```
cooks = ['peter', 'paul', 'mary']
```

```
for cook in cooks:
```

for循環

```
    print(f"{cook.title()} is a good cook!")
```

```
    print(f"And {cook.title()} works in HK!\n")
```

```
print(f"We got {len(cooks)} nice cooks in HK!")
```

輸出

Peter is a good cook!

And Peter works in HK!

Paul is a good cook!

And Paul works in HK!

Mary is a good cook!

And Mary works in HK!

We got 3 nice cooks in HK!

沒有放進 for 循環塊

有時可能會忘記縮進，使代碼沒有放進 for 循環中，然後看起來像這樣.....😅

應該重覆列印的句子被跳過了

有意將代碼放入 for 循環中，但卻忘記了 ...

```
cooks = ['peter', 'paul', 'mary']
for cook in cooks:
    print(f"{cook.title()} is a good cook!")
    print(f"And {cook.title()} works in HK!\n")
print(f"We got {len(cooks)} nice cooks in HK!")
```

for 循環

輸出

```
Peter is a good cook!
And Peter works in HK!
Paul is a good cook!
And Paul works in HK!
Mary is a good cook!
And Mary works in HK!
We got 3 nice cooks in HK!
```

堂課 - 請打開Thonny一起做

range()

粉紅色斜體: 題目

黑色正常字體: 提供的程式碼

紅色底線: 請填入程式碼

程式碼

使用 range() 列印 2到6

```
for number in range(____):  
    print(_____)
```

使用 range() 列印 0到4

```
for _____  
    print(_____)
```

輸出

2

3

4

5

6

0

1

2

3

4

列出項目

粉紅色斜體: 題目

黑色正常字體: 提供的程式碼

紅色底線: 請填入程式碼

程式碼

建立列表變量 foods

foods = ['apple', 'bread', 'chicken']

使用 for 列出 foods 中的項目

for food in _____
 print(_____)

輸出

apple

bread

chicken

進階／延伸應用(一)

嵌套式循環／巢狀迴圈 Nested loop

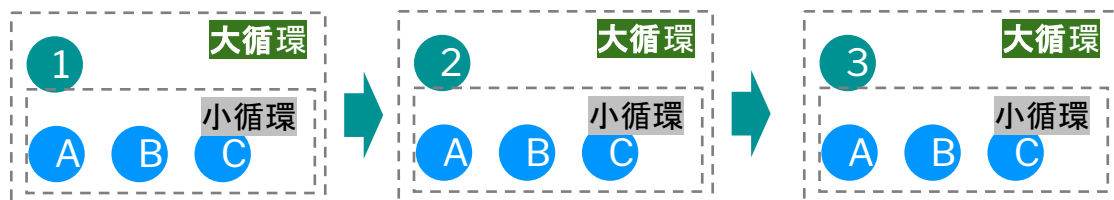
編程時，你可能會遇到需要在一個**大循環**中進行**小循環**的情況。

這就稱為**嵌套式循環**（巢狀迴圈）。

也就是會有一個**外部循環** **outer loop**，和一個**內部循環** **inner loop**。

外部循環：
["green", "big", "juicy"]
[1 2 3]

內部循環：
["apple", "banana", "cherry"]
[A B C]



```
adjectives = ["green", "big", "juicy"]  
fruits = ["apple", "banana", "cherry"]
```

```
for adjective in adjectives:
```

外部循環

```
    for fruit in fruits:
```

內部循環

```
        print(adjective + " " + fruit)
```

取用列表項目

粉紅色斜體: 題目

黑色正常字體: 提供的程式碼

紅色底線: 請填入程式碼

程式碼

```
# 建立列表變量 foods
foods = ['apple', 'bread', 'chicken']

# 大循環 - 1、2、3
for i in _____

    # 小循環 - 列出項目
    for food in _____

        # 列印 大循環數字 + 項目
        print(f"{{}} {{}}".format(i, food))
```

輸出

```
1 apple
1 bread
1 chicken

2 apple
2 bread
2 chicken

3 apple
3 bread
3 chicken
```

進階／延伸應用(二)

range() 步進值參數

當你在 `range()` 函數中傳入 3 個參數時，

你可以在第三個參數中設定 **步進值** (間隔大小)

```
for number in range(1, 10, 2):  
    print(number)
```

從這個值
開始

在這個值之
前停止

「步進值」

輸出

1
3
5
7
9

使用 range() 函數生成數字列表

我們可以使用 `range()` 和 `list()` 來生成一個數字列表

- 先用 `range()` 生成一系列數字
- 再用 `list()` 轉換成一個列表 `list`。

```
number_list = range(6)
```

```
print(number_list)
```

◀ 這不是一個列表

```
number_list = list(range(6))
```

```
print(number_list)
```

◀ 在這裡，我們將它轉換為一個列表

輸出

```
range(0, 6)
```

◀ 這不是一個列表

```
[0, 1, 2, 3, 4, 5]
```

◀ 這是一個列表

列表推導式 List Comprehensions

基本的方法

➡ 一種 Pythonic 高級編寫代碼的方式

4行

```
squares = []  
for number in range(1, 6):  
    square = number ** 2  
    squares.append(square)  
print(squares)
```

輸出

[1, 4, 9, 16, 25]

1行

```
squares = [ number ** 2 for number in range(1,6) ]  
print(squares)
```

輸出

[1, 4, 9, 16, 25]

數字列表的常用分析函數

我們可以使用以下函數對數字列表進行一些統計分析：

- `sum()`: 總和
- `max()`: 最大數值
- `min()`: 最小數值
- `len()`: 列表長度

```
>>> number_list = list(range(1, 11))
>>> number_list
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> sum(number_list)
55
>>> max(number_list)
10
>>> min(number_list)
1
>>> len(number_list)
10
```


總結

```
# for循環(range)
for i in range(0, 5):
    print(i)
```

```
# for循環(列出項目)
for item in list:
    print(item)
```

```
# 縮進
for j in range(5):
    print(j)
```

```
# 嵌套式循環
for i in range(0, 3):
    for item in list:
        print(i)
        print(item)
```