

Week04 - 2023-1015

- ☒ `for`
- ☐ `while`

一個半小時講解，一個半小時題目講解 [c315](#), [e948](#), [d485](#), [d490](#), [c418](#), [c419](#), [c420](#)

上週回顧 `list` 的使用

新增元素到 `list`

```
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print(fruits)          # ['apple', 'banana', 'cherry', 'orange']
```

`for` 迴圈使用

依序處理一份有順序可迭代(iterable) 的資料，直到所有資料處理完畢為止。

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

由上述例子我們可以回顧先前學習到 `range` 的用法，`range` 會產生一個有順序可迭代的資料，而 `for` 迴圈會依序處理這個資料，直到所有資料處理完畢為止。

`for` 迴圈的語法

```
for 變數 in 可迭代物件:
    # 做些什麼
```

用例子思考 `for` 的使用

我們從以下例子可以看出，在Python中，`for` 迴圈的使用非常靈活，它可以用於遍歷各種數據結構，包括列表，以及在不同情境下使用不同的迭代方式。以下是對每個情況的簡要說明：

- 第一個例子使用 `for` 迴圈遍歷列表 `prices` 中的元素，並將每個價格打印出來。
- 第二個例子使用 `for` 迴圈遍歷一個範圍，從0到5，並打印出每個數字。
- 第三個例子使用 `for` 迴圈和 `range` 函數，遍歷範圍內的數字，然後使用這些數字作為索引來訪問 `prices` 列表中的元素，並將它們打印出來。
- 最後一個例子是使用 `for` 迴圈和 `range` 函數，但使用步進值為2，這導致只遍歷 `prices` 列表中的偶數索引，並打印出相應的價格。

```
prices = [100, 200, 300, 400, 500]
for i in prices:
    print(i)

for i in range(0, 5, 1):
    print(i)
# 1
# 2
# 3
# 4
# 5

for i in range(0, 5, 1):
    print(prices[i])
# 100
# 200
# 300
# 400
# 500

for i in range(0, 5, 2):
    print(prices[i])
# 100
# 300
# 500
```

這些例子突出了 for 迴圈在Python中的多功能性和適應性，可以根據需要進行多種不同的迭代操作。

while 迴圈使用

當條件成立時，重複執行某些程式碼，直到條件不成立為止。

```
i = 0
while i < 5:
    print(i)
    i += 1
```

while 迴圈的語法

```
while 條件:
    # 做些什麼
```

用例子思考 **for** 和 **while** 的差別

我們可以想像 **for** 是已經知道次數，並且要每一個都走過；然而 **while** 是你只知道這麼做是對的，但你不知道終點在哪裡，所以你只能一直做下去，直到你發現終點。

- 命題：當今天肚子餓就要吃飯，並且要吃到飽
- **for**：我知道每碗飯有多少份量，我這餐吃五碗就會飽了，所以我一吃完五碗就不吃了。
- **while**：我不知道每碗飯有多少份量，但我知道我要吃到飽，所以我一直吃直到吃不下為止。

```
# for
for i in 五碗飯:
    吃一碗飯

# while
while 肚子餓:
    吃一碗飯，直到吃飽為止
```

break 與 **continue** 的使用

break 的使用

```
for i in range(5):
    if i == 3:
        break
    print(i)

# 0
# 1
# 2
# 4
```

continue 的使用

```
for i in range(5):
    if i == 3:
        continue
    print(i)

# 0
# 1
# 2
```

c315 I, ROBOT 前傳

小明有一台機器人，藉由對機器人下達指令，小明可以控制機器人的移動。機器人一開始在原點 (x=0, y=0) 的位置。下達了很多個指令後，小明卻找不到他的機器人最後移動到哪一格，他把所有下過的指令都給你，請你幫他計算機器人最後移動到了哪一格？

機器人的指令由兩個數字 a b 組成，a 代表移動的方向，b 代表移動的格子數。

- a = 0 時表示往 y 正方向移動
- a = 1 時表示往 x 正方向移動
- a = 2 時表示往 y 負方向移動
- a = 3 時表示往 x 負方向移動

舉例如下：a=2 b=3 時代表往 y 的負方向移動 3 格。

Sample	Input	Output
說明	第 1 行有一個數字 N，表示小明下過的指令數。接下來 N 行，依序為機器人收到的指令，每行有兩個數字 a b。 (0 <= a <= 3) (b >= 0)	請輸出兩個整數，x y，中間用一個空格分隔。表示機器人最後停的位置。
# 1	4 0 10 1 4 2 3 3 6	-2 7

[Python 解]

```
x = 0
y = 0

times = int(input())

for i in range(times):
    a, b = map(int, input().split())

    if a == 0:
        y = y + b
    elif a == 1:
        x = x + b
    elif a == 2:
        y = y - b
    elif a == 3:
        x = x - b

print(x, y)
```

d485 我愛偶數

文文很喜歡偶數，他甚至有收集偶數的習慣。你給他一個範圍的連續整數，他就會把其中的偶數留下來收藏。如今他又拿到了一個範圍的整數，請問他這次收藏了幾個偶數？對文文來說，0 也算是一個偶數哦！

Sample	Input	Output
說明	輸入只有一行，其中含有兩個由空白隔開的整數 a, b ($0 \leq a \leq b \leq 2147483647$)。	輸出一個整數，代表 a 與 b 之間 (含 a 與 b) 一共有多少個偶數。
# 1	1 4	2

[Python 解]

```
start, end = map(int, input().split())

if start % 2 == 1:
    start += 1
if end % 2 == 1:
    end -= 1

times = ((end - start) / 2) + 1

print(int(times))
```

d490 我也愛偶數

文文愛偶數，無獨有「偶」地，珊珊也愛偶數。珊珊除了收藏偶數以外，每次她收到一些數字時，她還會把其中的偶數挑出來把玩並予以加總。今天珊珊又收到了一個範圍的連續整數，請問這次她從這段數字中所收集到的偶數的總和是多少？

Sample	Input	Output
說明	輸入只有一行，其中含有兩個由空白隔開的整數 a, b ($0 \leq a \leq b \leq 2147483647$)。	請輸出一個整數，代表 a 與 b 之間 (含 a 與 b) 所有偶數的和，(答案會 ≤ 2147483647)。
# 1	2 5	6

[Python 解]

```
a, b = map(int, input().split())
ans = 0

for i in range(a, b + 1):
    if i % 2 == 0:
        ans += i

print(ans)
```

--- [本週上課內容分隔線] ---

c418 Bert的三角形 (1)

Bert 想要一個 n 層的三角形，第 i 層就要有 i 個 " * "

請你寫個程式幫幫可憐的 Bert ~~

Sample	Input	Output
說明	單筆輸入~~ 輸入只有一個整數 n ($1 \leq n \leq 100$)	輸出整個三角形~~
# 1	3	* ** ***

[Python 解]

```
layer = int(input())

for i in range(1, layer + 1):
    print('*' * i)
```

c419 Bert的三角形 (2)

Bert 又想要另外一種 n 層的三角形，定義如下:

第 i 層一樣要有 i 個 " * "，但要向右對齊

請你寫個程式幫幫 Bert ~~

Sample	Input	Output
說明	單筆輸入~~ 輸入只有一個整數 n ($1 \leq n \leq 100$)	輸出整個三角形~~ 因為空格不好辨識，請以 "_" 代替 ~~
# 1	3	_* _** _***

[Python 解]

```
layer = int(input())

for i in range(1, layer + 1):
    print('_' * (layer - i) + '*' * i)
```

c420 Bert的三角形 (3)

Bert 有天騎著海豚到了埃及，看到了金字塔不經意的發出『哇~~』現在 Bert 想請你用程式記下金字塔的樣子~~現在有一種 n 層的三金字塔，定義如下:

第 i 層要有相對數量的 " * "，請注意要像金字塔一樣向中間對齊

請你寫個程式幫幫 Bert ~~

Sample	Input	Output
說明	單筆輸入~~	輸出整個三角形~~
	輸入只有一個整數 n (1 <= n <= 100) n 保證為奇數	因為空格不好辨識，請以 "_" 代替 ~~
# 1	3	<pre> _ _ * _ _ _ * * * _ _ _ _ * * * * *</pre>

[Python 解]

```
layer = int(input())

for i in range(1, layer + 1):
    print('_' * (layer - i), end='')
    print('*' * ((2 * i) - 1), end='')
    print('_' * (layer - i))
```

License

All of these teaching materials are owned by [Hugo ChunHo Lin](#).
These materials are intended for tutoring purposes. They are open-source to foster a more vibrant Python learning community. We warmly welcome fellow enthusiasts interested in Python to use them. If you use a substantial portion of the source code, please include a link back to this repository.