循環 While-Loop

Python 編程課程 第 4 課

- While 循環
- 中斷、跳過 break & continue
- [重溫]嵌套式循環 Nested loop

熱身：重覆買飲品

情境：
你幫你的朋友到自動販賣機買飲品，在自動販賣機的視角下：

當有商品庫存，你又有付錢[條件]
它就會分發商品。[循環]

（換句話說：一旦缺貨／你不再投幣，就會停止重覆出貨）

熱身:重覆買飲品

自動販賣機程式碼	**輸出**
choice = input("你要什麼飲品: ") while (choice 還有庫存) and (你有付錢): 　出貨 　choice庫存 - 1	你要什麼飲品: 維X奶 <你得到一瓶維X奶>

<u>簡易記憶點:</u>
while-loop = 有條件地循環

While 循環例子(一)- 當.....就重覆

While 循環適用場合：
知道條件 卻**不肯定執行次數**

[條件] 只要stop不等於”n”
[循環] 列印i、i增加1、再次詢問stop

（**終止條件**：stop == “y”）
（**執行次數**：未知）

程式碼

```
i = 1
stop = "n"                                  while 迴圈
while(stop != "y"):
    print(i)
    i = i + 1
    stop = input("Stop? (y/n): ")
```

輸出

```
1
Stop? (y/n): n
2
Stop? (y/n): n
3
Stop? (y/n): n
4
Stop? (y/n): y
```

While 循環例子(二)- 實現 For 循環

程式碼

```
1  i = 1
2  while( i < 4 ):                    while 循環
3      print(i)
4      i = i + 1


print("Program ends, i = 4")
```

程式碼

```
for 1 in range(1, 4):                 for 循環
    print(i)


print("Program ends, i = 4")
```

四個重要元素:

1. 初始化:給變數一個初始值。
2. While 循環條件:循環開始時評估的布林表達式。
3. 循環本體:當條件為真時要重覆執行的程式碼。
4. (可選)變數更改:每次迭代更改變數值, 將反映到循環條件的判斷。

輸出

```
1
2
3
Program ends, i = 4
```

堂課 - 請打開Thonny一起做

練習：重覆新增水果

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

```
fruits = []  # 空列表 fruits
count = 0          # 計數變量 count
new_fruit = "" # 初始化被新增水果

# 進入循環，只要水果不輸入 "n"就：
while(_____________):
    # 使用 input() 寫入 new_fruit
    new_fruit = _____________________________
    # 如果新水果不是 "n"就加入 fruits 和增加 count
    if _____________:
    _____________

# 列印
```

輸出

```
Add fruit ("n" to quit):apple
Add fruit ("n" to quit):banana
Add fruit ("n" to quit):cherry
Add fruit ("n" to quit):n

There are 3 fruits.
They are: ['apple', 'banana', 'cherry']
```

練習：重覆新增水果

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

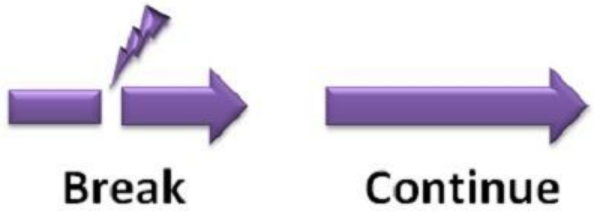
程式碼

```
fruits = []
count = 0
new_fruit = ""

while(new_fruit != "n"):

    new_fruit = input("Add fruit (\"n\" to quit):")

    if new_fruit != "n":
        fruits.append(new_fruit)
        count = count + 1

print("There are", count, "fruits.")
print("They are:", fruits)
```

輸出

```
Add fruit ("n" to quit):apple
Add fruit ("n" to quit):banana
Add fruit ("n" to quit):cherry
Add fruit ("n" to quit):n

There are 3 fruits.
They are: ['apple', 'banana', 'cherry']
```

中斷、跳過 break & continue



break - 跳出迴圈	**continue - 直接進入下個循環**
for i in range(0, 1000): … *< 循環塊中 >* if … : break … … *< 循環塊外面 >* …	for i in range(0, 1000): … *< 循環塊中 >* if … : continue … *< 循環塊外面 >* …

中斷 Break 例子

使用者輸入正數，並使用 while 循環繼續提示，**直到輸入負數**

循環後列印正數總和

程式碼

```
total = 0

while True:

    num = int(input("Enter a positive number: "))
    if num < 0:
        break
    total = total + num

print("Sum of positive numbers:", total)
```

輸出

```
Enter a positive number: 10
Enter a positive number: 5
Enter a positive number: 8
Enter a positive number: -2
Sum of positive numbers: 23
```

跳過 Continue 例子

挑選球員，如果名字長度4個字母以上就**跳過不選**。

程式碼

```
player_list = ["Alan", "Alex", "Chris", "Ben",
"Daniel", "Ken", "Jackson", "Leo"]
pick_list = []

for name in player_list:
    if len(name) > 4:
        continue
    pick_list.append(name)

print(pick_list)
```

輸出

```
['Alan', 'Alex', 'Ben', 'Ken', 'Leo']
```

堂課 - 請打開Thonny一起做

練習：加人工談判

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

```
amount = 0
# 使用 while(True): 直接進入循環
while(True):
    print("Amount now:", amount)
    # 使用 input() 輸入人工加幅
    increase = ____________________________
    # 如果你加太多(>100)，就跳過這個迴圈
    if ____________________:
        print("Too large.")

        ____________
    # 如果你輸入負數(<0)，就中斷這個迴圈
    if ____________________:
        print("Oh no!")

        ______
    # 累積人工加幅
    amount = ____________________________

# 列印最終加幅
```

輸出

```
Amount now: 0
How much you would like to increase: 20
Amount now: 20
How much you would like to increase: 100
Amount now: 120
How much you would like to increase: 101
Too large.
Amount now: 120
How much you would like to increase: 99
Amount now: 219
How much you would like to increase: -5
Oh no!
Total amount: 219
```

練習：加人工談判

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

```
amount = 0

while(True):
    print("Amount now:", amount)
    # 使用 input() 輸入人工加幅
    increase = int(input("How much you would like to increase: "))
    # 如果你加太多(>100)，就跳過這個迴圈
    if increase > 100:
        print("Too large.")
        continue
    # 如果你輸入負數(<0)，就中斷這個迴圈
    if increase < 0:
        print("Oh no!")
        break
    # 累積人工加幅
    amount = amount + increase

print("Total amount:", amount)
```

輸出

```
Amount now: 0
How much you would like to increase: 20
Amount now: 20
How much you would like to increase: 100
Amount now: 120
How much you would like to increase: 101
Too large.
Amount now: 120
How much you would like to increase: 99
Amount now: 219
How much you would like to increase: -5
Oh no!
Total amount: 219
```

嵌套式循環 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）。

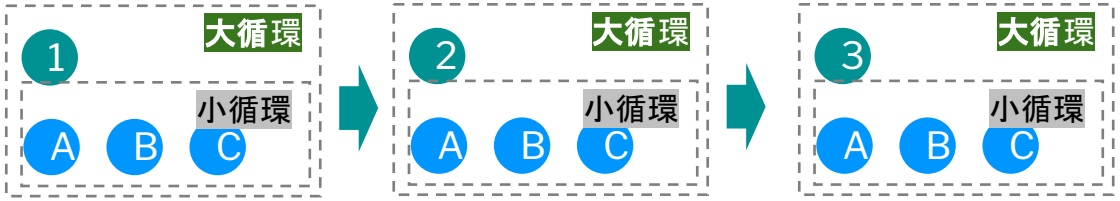
也就是會有一個**外部循環 outer loop**，和一個**內部循環 inner loop**。

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:
    for fruit in fruits:
        print(adjective + " " + fruit)
```

外部循環

內部循環

堂課 - 請打開Thonny一起做

While 版嵌套式循環

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
i = _____
while i_____ :
    # 小循環 - 列出項目
    index = 0
    while index_____ :
        # 列印 大循環數字 + 項目
        print(f"{_____} {_______}.")
        # 循環變量 i 和 index +1
        index = _________
    i = _________
```

輸出

```
1 apple
1 bread
1 chicken

2 apple
2 bread
2 chicken

3 apple
3 bread
3 chicken
```

總結

```
# while循環（當……就重覆）
while user_input != "No":
        salary = salary + 100

# while循環（實現 For 循環）
i = 0
while i < 9:
        print(i)
        i = i + 1

# 中斷、跳過 break & continue
if user_input > 100:
        continue
elif user_input < 0:
        break
```

```
# While 版嵌套式循環
i = 0
while i <= 3:
        j = 0
        while j <= 5:
                print(j)
                j = j + 1
        i = 1 + 1
```