



AI.FINTECH
A I 金 融 科 技 中 心

快速了解Python

製作人：黃宥輔



黃宥輔



現任



國立高雄科技大學 AI金融科技中心 講師

國立高雄科技大學 金融資訊系 講師

AIGO、產業人才投資計畫、新尖兵課程 講師

學歷



國立高雄科技大學 金融資訊系 碩士

國立高雄科技大學 會計資訊系 學士

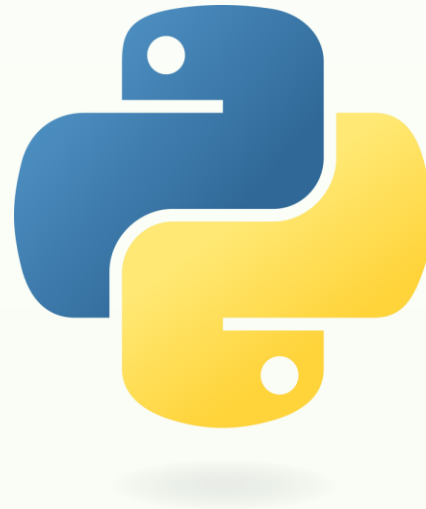
專長



大數據分析、AI機器學習預測模型建構、Android開發



- **高階程式語言**：Python 是一種直譯式、動態類型的高階語言
- **開源與免費**：Python 是開源且免費的，擁有龐大的社群支持
- **跨平台支援**：可在 Windows、Mac、Linux 等多種平台運行



- **簡潔易讀**：強調可讀性，程式碼結構明確
- **縮排語法**：使用縮排取代括號，強制良好程式碼風格
- **動態類型**：變數不需要聲明類型，靈活性高

Python

```
1 # 印出 1 到 5 的數字
2 for i in range(1, 6): # 使用 range 函數產生數字範圍
3     print(f"Number: {i}") # f-string 格式化輸出
```

JAVA

```
1 // 印出 1 到 5 的數字
2 public class Main {
3     public static void main(String[] args) {
4         for (int i = 1; i <= 5; i++) { // 明確定義變數類型
5             System.out.println("Number: " + i); // 使用字符串拼接輸出
6         }
7     }
8 }
```

- 多範式支援：支援面向物件、函數式及程序式編程
- 豐富第三方庫：如 NumPy、Pandas、TensorFlow，適合數據分析與機器學習
- 優秀的文字處理能力：適合處理自然語言、爬蟲與自動化腳本。



- 官方文件：Python 官方提供詳細的使用手冊與 API 文件
- 開放學習資源：有許多免費教學網站，如 Real Python、W3Schools
- 支援社群：Stack Overflow、Reddit 等專注於解決開發者的問題
- 套件管理工具：使用 pip 或 conda 管理第三方庫



國立高雄科技大學AI金融科技中心Youtube頻道
www.youtube.com/@aifintech_nkust

【教學】Anaconda安裝【高科大AI金融科技中心】
<https://www.youtube.com/watch?v=sL5blh88aho>

- 開頭：英文或底線，不能使用數字，不建議使用中文，如Var_1
- 大小寫有區別：Var_1與var_1是兩個不同的變數
- 變數之間不能有空格Var 1不符合變數規則
- 盡量避免使用保留字，如def、for、import等，簡便的判別辦法為編譯器會將報留字用不同顏色標註

如下圖，黑色的為一般變數，紅色的為數字，藍色的為保留字，紫色的為內建函數

```
1 # 印出 1 到 5 的數字
2 for i in range(1, 6): # 使用 range 函數產生數字範圍
3     print(f"Number: {i}") # f-string 格式化輸出
```


型別	變數範例
整數(int)	A = 123
浮點數(float)	A = 123.45
布林值	A = True
字串(String)	A = "123"
串列(list)	A = [1, 2, "字串"]
字典(dict)	A = {1 : [1, 2, 3], 2 : "字串" }

在Python中，布林值的True可視為整數1、False可視為整數0，所以也可以加入運算，例如True * 3，結果為3

```
1 String_1 = "高科大"
2 List_1 = [1, 2, "高科大"]
3
4 print(String_1[0])
5 print(List_1[1])
```

索引(index)	0	1	2
變數名稱			
String_1	高	科	大
List_1	1	2	高科大

Var[start : stop : step]

start: 起始索引 (包含) , 默認為 0 。

stop: 終止索引 (不包含) , 默認為結束位置 。

step: 步長 , 默認為 1 。

```
1 List_2 = list(range(10, 20))
2 print(List_2[1::2])
```

index	value
0	10
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19

$a, b = 7, 3$

運算子	說明	範例	輸出結果
+	加法	$a + b$	10
-	減法	$a - b$	4
*	乘法	$a * b$	21
/	除法	a / b	2.3333333333333335
//	整數除法(取商數)	$a // b$	2
%	取餘數	$a \% b$	1
**	次方	$a ** b$	343

```
1 list_1 = [1, 2, 3]
2 list_2 = [4, 5, 6]
3
4 Str_1 = "高科大"
5 Str_2 = "AI金融科技中心"
6
7 print(list_1 + list_2)
8 print(list_1 * 2)
9
10 print(Str_1 + Str_2)
11 print(Str_1 * 3)
```

輸出結果

```
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 1, 2, 3]
```

```
高科大AI金融科技中心
高科大高科大高科大
```

$a, b = 7, 3$

運算子	說明	範例	輸出結果
$>$	大於	$a > b$	True
$<$	小於	$a < b$	False
$>=$	大於等於	$a >= b$	True
$<=$	小於等於	$a <= b$	False
$==$	相等	$a == b$	False
$!=$	不相等	$a != b$	True

$x = 2 > 3$

False

$y = 2 < 3$

True

運算子	說明	範例	輸出結果
and	運算子兩邊都是True 判斷出來才會是True	x and y	False
or	運算子其中一邊是True 判斷出來就會是True	x or y	True
not	True、False反轉	not(x)	True

通常根據關係運算式、邏輯運算式，判斷出的結果，來執行不同的程式碼

例題：

輸入一個景氣分數，判斷現在的景氣燈號，規則如下，若輸入錯誤請輸出，輸入錯誤

分數	燈號	代表意義
9到16分	藍燈	景氣低迷
17到22分	黃藍燈	景氣轉向
23分到31分	綠燈	景氣穩定
32分到37分	黃紅燈	景氣轉向
38分到45分	紅燈	景氣熱絡

```
1  # -*- coding: utf-8 -*-
2  score = 25
3
4  if score >= 9 and score <= 16:
5      print("藍燈")
6  elif score >= 17 and score <= 22:
7      print("黃藍燈")
8  elif score >= 23 and score <= 31:
9      print("綠燈")
10 elif score >= 32 and score <= 37:
11     print("紅黃燈")
12 elif score >= 38 and score <= 45:
13     print("紅燈")
14 else:
15     print("輸入錯誤")
```

- 從if開始判斷，找到符合條件的就離開
if...elif...else判斷
- 以縮排表示程式碼的歸屬

輸出結果

綠燈

```
1  # -*- coding: utf-8 -*-
2  score = 100
3
4  if score >= 9 and score <= 16:
5      print("藍燈")
6  elif score >= 17 and score <= 22:
7      print("黃藍燈")
8  elif score >= 23 and score <= 31:
9      print("綠燈")
10 elif score >= 32 and score <= 37:
11     print("紅黃燈")
12 elif score >= 38 and score <= 45:
13     print("紅燈")
14 else:
15     print("輸入錯誤")
```

輸出結果

輸入錯誤

需要通過多個層級的判斷，會使用巢狀結構

例題：

男性健康要為建議為90公分，女性為80公分

```
1  #-*- coding: utf-8 -*-
2  gender = "男"
3  waistline = 85
4
5  if gender == "男":
6      if waistline > 90:
7          print("腰圍過大，請注意健康")
8      else:
9          print("腰圍正常，請繼續保持")
10 elif gender == "女":
11     if waistline > 80:
12         print("腰圍過大，請注意健康")
13     else:
14         print("腰圍正常，請繼續保持")
```

輸出結果

腰圍正常，請繼續保持

```
1  # -*- coding: utf-8 -*-
2  gender = "女"
3  waistline = 85
4
5  if gender == "男":
6      if waistline > 90:
7          print("腰圍過大，請注意健康")
8      else:
9          print("腰圍正常，請繼續保持")
10 elif gender == "女":
11     if waistline > 80:
12         print("腰圍過大，請注意健康")
13     else:
14         print("腰圍正常，請繼續保持")
```

輸出結果

腰圍過大，請注意健康

除了以下幾種，放入if條件判斷的都會是True，所以放入這些值以外的，都會是True

例如檔案處理時，藉次判斷這個值是否有取得下一行的文字，進而判斷跳出迴圈

項目	內容
整數	0
布林值	False
空值	None
空字串	""
空容器	[] 、 {} 、 set() 、 ()
虛數部分為 0 的複數	0j

確定次數的迴圈，使用for迴圈撰寫

例題：請計算1到100的加總

```
1 # -*- coding: utf-8 -*-
2 total = 0
3 for i in range(1, 101):
4     total += i
5 print(total)
```

輸出結果

5050

巢狀for迴圈如同時鐘一般，內圈圍長針，外圈圍短針，長針走完一圈後，短針前進一格，長針又繼續從頭開始旋轉，直到短針走完一圈

例題：有78元需要找零，硬幣為50元、10元、5元、1元

請列出所有找零的方法

```
1  #-*-coding:utf-8-*-
2  money = 78
3  # 該硬幣最多有幾個
4  max_50 = money // 50
5  max_10 = money // 10
6  max_5 = money // 5
7  max_1 = money
8
9  for coin_50 in range(max_50):
10     for coin_10 in range(max_10):
11         for coin_5 in range(max_5):
12             for coin_1 in range(max_1):
13                 if coin_50 * 50 + coin_10 * 10 + coin_5 * 5 + coin_1 == money:
14                     print(f"{coin_50}個50元、{coin_10}個10元、{coin_5}個5元、{coin_1}個1元")
```

巢狀for迴圈

輸出結果



AI.FINTECH

AI 金融 科技 中心

1	0個50元、0個10元、1個5元、73個1元	29	0個50元、2個10元、0個5元、58個1元	51	0個50元、4個10元、0個5元、38個1元
2	0個50元、0個10元、2個5元、68個1元	30	0個50元、2個10元、1個5元、53個1元	52	0個50元、4個10元、1個5元、33個1元
3	0個50元、0個10元、3個5元、63個1元	31	0個50元、2個10元、2個5元、48個1元	53	0個50元、4個10元、2個5元、28個1元
4	0個50元、0個10元、4個5元、58個1元	32	0個50元、2個10元、3個5元、43個1元	54	0個50元、4個10元、3個5元、23個1元
5	0個50元、0個10元、5個5元、53個1元	33	0個50元、2個10元、4個5元、38個1元	55	0個50元、4個10元、4個5元、18個1元
6	0個50元、0個10元、6個5元、48個1元	34	0個50元、2個10元、5個5元、33個1元	56	0個50元、4個10元、5個5元、13個1元
7	0個50元、0個10元、7個5元、43個1元	35	0個50元、2個10元、6個5元、28個1元	57	0個50元、4個10元、6個5元、8個1元
8	0個50元、0個10元、8個5元、38個1元	36	0個50元、2個10元、7個5元、23個1元	58	0個50元、4個10元、7個5元、3個1元
9	0個50元、0個10元、9個5元、33個1元	37	0個50元、2個10元、8個5元、18個1元	59	0個50元、5個10元、0個5元、28個1元
10	0個50元、0個10元、10個5元、28個1元	38	0個50元、2個10元、9個5元、13個1元	60	0個50元、5個10元、1個5元、23個1元
11	0個50元、0個10元、11個5元、23個1元	39	0個50元、2個10元、10個5元、8個1元	61	0個50元、5個10元、2個5元、18個1元
12	0個50元、0個10元、12個5元、18個1元	40	0個50元、2個10元、11個5元、3個1元	62	0個50元、5個10元、3個5元、13個1元
13	0個50元、0個10元、13個5元、13個1元	41	0個50元、3個10元、0個5元、48個1元	63	0個50元、5個10元、4個5元、8個1元
14	0個50元、0個10元、14個5元、8個1元	42	0個50元、3個10元、1個5元、43個1元	64	0個50元、5個10元、5個5元、3個1元
15	0個50元、1個10元、0個5元、68個1元	43	0個50元、3個10元、2個5元、38個1元	65	0個50元、6個10元、0個5元、18個1元
16	0個50元、1個10元、1個5元、63個1元	44	0個50元、3個10元、3個5元、33個1元	66	0個50元、6個10元、1個5元、13個1元
17	0個50元、1個10元、2個5元、58個1元	45	0個50元、3個10元、4個5元、28個1元	67	0個50元、6個10元、2個5元、8個1元
18	0個50元、1個10元、3個5元、53個1元	46	0個50元、3個10元、5個5元、23個1元	68	0個50元、6個10元、3個5元、3個1元
19	0個50元、1個10元、4個5元、48個1元	47	0個50元、3個10元、6個5元、18個1元		
20	0個50元、1個10元、5個5元、43個1元	48	0個50元、3個10元、7個5元、13個1元		
21	0個50元、1個10元、6個5元、38個1元	49	0個50元、3個10元、8個5元、8個1元		
22	0個50元、1個10元、7個5元、33個1元	50	0個50元、3個10元、9個5元、3個1元		
23	0個50元、1個10元、8個5元、28個1元				
24	0個50元、1個10元、9個5元、23個1元				
25	0個50元、1個10元、10個5元、18個1元				
26	0個50元、1個10元、11個5元、13個1元				
27	0個50元、1個10元、12個5元、8個1元				
28	0個50元、1個10元、13個5元、3個1元				

不確定次數的迴圈，使用while迴圈撰寫

例題：猜數字遊戲

範圍是1到100，讓使用者重複輸入，直到達對，每次答錯
會重新提示範圍，最終計算猜了幾次

while迴圈



```
1  #-*-coding:utf-8-*-
2  import random as rd
3  answer = rd.randint(1,100) # 隨機答案
4
5  count = 0 # 猜測次數
6  guess = 0 # 猜測值的變數
7  # 範圍是1到100，根據猜測變動範圍
8  lower = 1 # 下限
9  upper = 100 # 上限
10 # 用while迴圈執行到猜中為止
11 while guess != answer:
12     guess = int(input("請輸入要猜的數字：")) # 讓玩家輸入猜測的數字
13     count += 1 # 累積猜測次數
14     if guess < answer: # 沒猜中，提示並更動下限
15         print("太小了！")
16         lower = guess
17         print(f"範圍是{lower}到{upper}")
18     elif guess > answer: # 沒猜中，提示並更動上限
19         print("太大了！")
20         upper = guess
21         print(f"範圍是{lower}到{upper}")
22     else:
23         print(f"猜中了！總共猜了{count}次") # 猜中後提示猜了幾次
```

輸出結果

請輸入要猜的數字：50
太小了！
範圍是50到100
請輸入要猜的數字：75
太大了！
範圍是50到75
請輸入要猜的數字：60
太大了！
範圍是50到60
請輸入要猜的數字：55
太小了！
範圍是55到60
請輸入要猜的數字：58
太小了！
範圍是58到60
請輸入要猜的數字：59
猜中了！總共猜了6次

需要重複使用之程式，可以寫成自訂函數(副程式)，
傳入不同的參數，可以根據相同的邏輯，得到不同的結果

例題：

周年慶推出優惠活動

消費滿999，現折100

消費滿499，現折50

請撰寫程式，傳入不同的金額，計算折扣後的金額


```
1  # -*- coding: utf-8 -*-
2  def discount_fun(total):
3      discount = 0
4      discount_999 = total // 999 * 100
5      discount_499 = total % 999 // 499 * 50
6      ...
7      discount += discount_999 + discount_499
8      result = total - discount
9      return result
10
11 customer_A = discount_fun(2000)
12 print(f"顧客A折扣後金額為{customer_A}")
13
14 customer_B = discount_fun(2500)
15 print(f"顧客B折扣後金額為{customer_B}")
```

輸出結果

顧客A折扣後金額為1800
顧客B折扣後金額為2250

章節到此結束，有任何問題歡迎提出來討論！