

CHAPTER 5

第五單元「函式」



1. 為什麼要使用函式？

當程式變得越來越複雜時，重複的程式碼會造成維護困難、可讀性降低，這時就需要「函式」來把重複的邏輯封裝成模組。

使用時機：

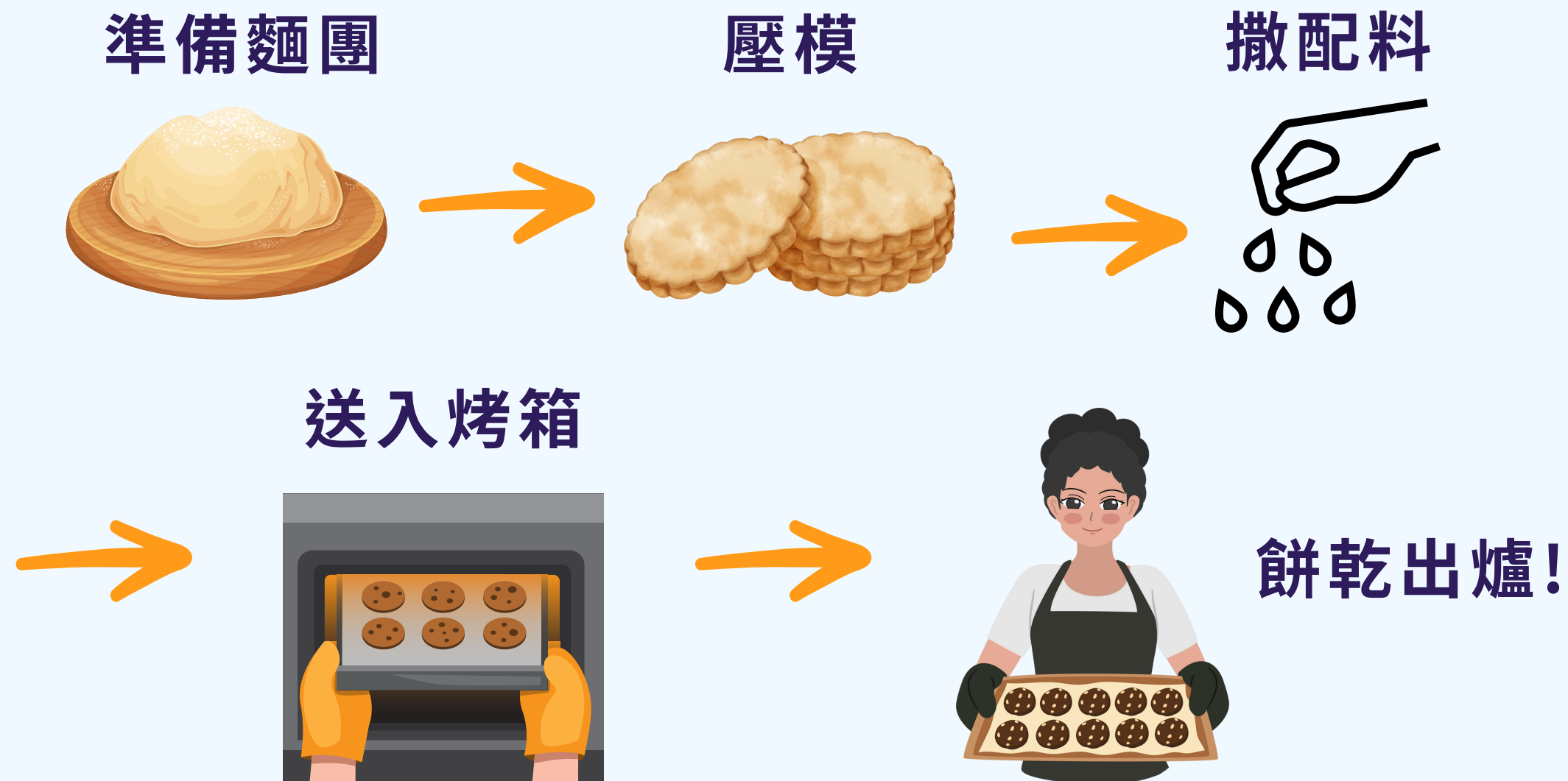
- 同樣的邏輯會被使用多次（例如：輸出格式化文字、計算數值）
- 希望讓程式分段更清楚，便於除錯與修改



1. 為什麼要使用函式？

舉例來說...

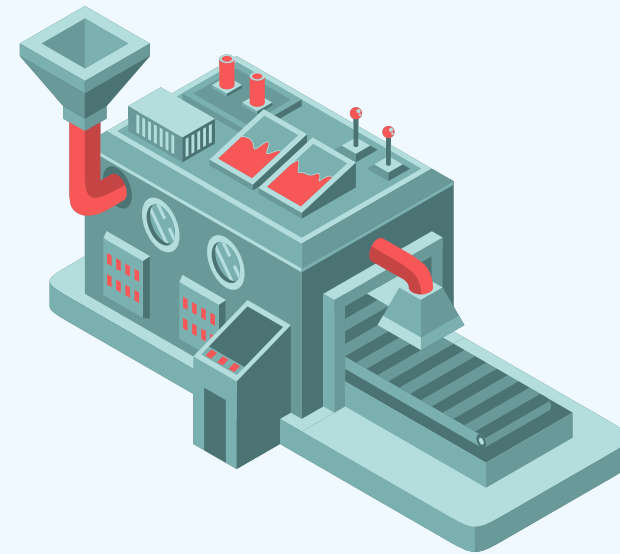
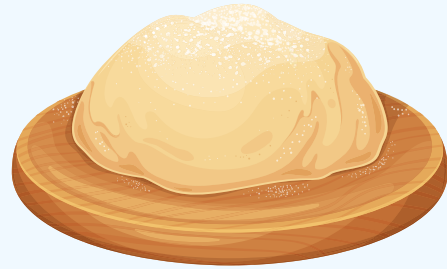
如果我們想要烤餅乾，
我們需要不斷重複右邊的步驟



1. 為什麼要使用函式？

現在有一台機器可以幫我們
一次完成中間的步驟

準備麵團



壓模
撒配料
送入烤箱



餅乾出爐！



像這樣可以把重複的動作(邏輯)
封裝成機器(模組)就稱為函式！

2. 定義與呼叫函式

語法格式：

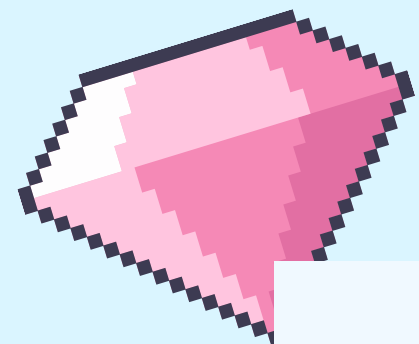
```
def 函式名稱():  
    # 要執行的程式
```

範例：

```
def say_hello():  
    print("Hello, Python!")  
  
say_hello()
```

注意事項：

- 定義函式使用 **def** 關鍵字
- 函式名稱後面要有括號 **()** 和冒號 **:**
- 函式內的程式碼要縮排



3. 加入參數：讓函式更有彈性

語法格式：

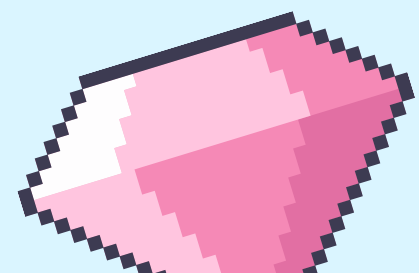
```
def greet(name):  
    print("Hi,", name)  
  
greet("吉多")
```

使用參數的好處：

- 可以傳入不同的值，讓函式功能更強大
- 提高程式的可重用性

但是，參數數量要匹配，否則會出現錯誤：

```
def greet(name):  
    print("Hi", name)  
  
greet() # × 會出錯，少了必要的參數
```



3. 加入參數：讓函式更有彈性

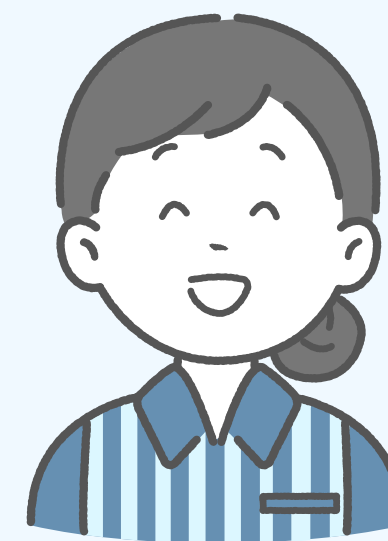
舉例來說...

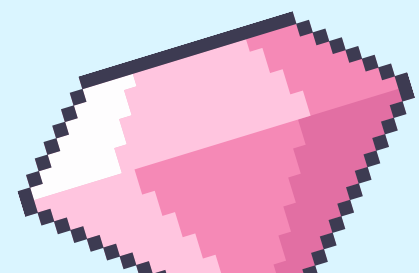
如果**函式**是一間**速食店**，我們需要告訴櫃台**餐點名稱**
(**參數**)

吉多點的**薯條**、**雞塊**和**可樂**就是使用者提供的不同**參數**！

我想要薯條、
可樂、雞塊

一份薯條、可
樂、雞塊馬上
來！





3. 加入參數：讓函式更有彈性

但是如果吉多甚麼都不點
(不給參數)，櫃檯就不知道
我們的需求(缺少參數)，也
無法回應(輸出)



我想要...

(他想要甚麼
呢?)



4.使用 return 回傳結果

函式除了執行動作（例如印出東西），也可以「回傳」一個值，讓你在主程式中使用這個結果。

```
def square(x):  
    return x * x  
  
result = square(5)  
print(result)  # 25
```

💡 為什麼要用 return ？

- 可以把值傳回來給主程式做更多運算或判斷
- print() 是印出結果，而 return 是帶回結果

注意：return 一執行就會結束函式內部執行流程。



5. 多參數與運算實作

函式可以有多個參數，搭配計算與邏輯處理：

```
def bmi(weight, height):  
    return weight / (height ** 2)  
  
print(bmi(70, 1.75)) # 22.86...
```

💡 小提醒：

- 寫函式時，參數的順序與意義要清楚，不然會導致誤判。
- 呼叫函式時的引數要對應正確順序！

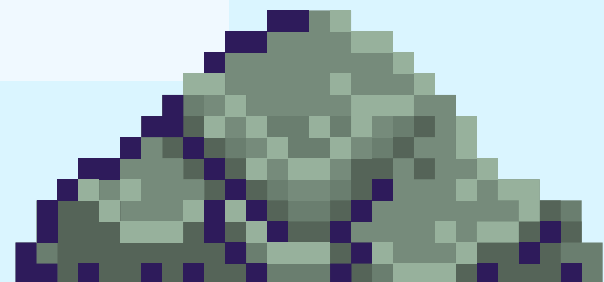
6. 函式的命名與設計原則

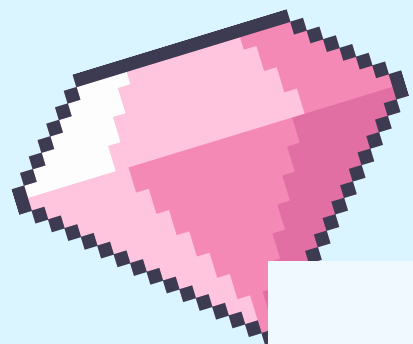
建議命名方式：

- 使用有意義的英文單字，並用小寫 + 底線分隔：
`calculate_area`、`print_info`
- 避免使用 `sum`, `list`, `input` 等 Python 內建關鍵字！

設計原則：

- 一個函式只做「一件事」：單一職責原則（Single Responsibility）
- 讓函式名稱能清楚說明它的用途
- 如果太長或做太多事情，應該拆成多個函式





7. 常見錯誤與除錯技巧

錯誤	原因	解法
<code>TypeError: greet() missing 1 required positional argument</code>	呼叫函式時少帶一個參數	確認定義與呼叫時參數數量相符
<code>None</code> 被當作回傳值	函式沒 <code>return</code> ， 預設回傳 <code>None</code>	使用 <code>return</code> 傳回結果
呼叫函式但沒加小括號	忘了 <code>()</code> ， 函式不會被執行	寫成 <code>my_func()</code> 而不是 <code>my_func</code>

準備好了嗎？
已經學會「函式」了嗎？
遇到困難記得找我求助喔~嘎嘎！



Let's
GoGoGo!!!

