

函式、參數傳遞與套件

滄海《Python 程式設計初學指引》二版·CH08

考試範圍提醒：老師投影片把「LEGB 規則、遞迴函式（河內塔）、傳函式當參數、匿名函式 lambda、過濾 filter()／映射 map()」都標為（不考）。本講義以考試範圍為主體（內建函式、自定義函式 def／參數／預設值／變數範圍、import 套件），不考內容放最後「補充」一節。

§ 1 名詞速查表

中文	English	一句話定義	例	易混
函式	function	一組命名、可重複使用的指令；分內建／自定義／套件三來源	<code>len(s)</code>	—
內建函式	built-in	Python 自帶、免 import 即可用	<code>max sorted abs</code>	套件函式要 import
<code>def</code>	—	定義自定義函式的關鍵字	<code>def f(): ...</code>	呼叫要在定義之後
參數 / 引數	parameter / argument	定義時的形式參數 / 呼叫時傳入的實際值	<code>def f(x) / f(5)</code>	—
<code>return</code>	—	函式回傳值並結束；可回傳多個（成元組）	<code>return a, b</code>	沒寫回傳 None
預設值參數	default parameter	定義時先給值，呼叫沒給就用預設	<code>def f(x=10)</code>	要放沒預設參數右邊
全域變數	global variable	在最外層主程式建立，整支程式可見	—	函式內同名重新賦值會變區域
區域變數	local variable	在函式內建立，只在函式內有效	—	函式外用會 NameError
套件 / 模組	package / module	別人寫好的功能集，用 <code>import</code> 引入	<code>time random sys</code>	—

中文	English	一句話定義	例	易混
<code>import ... as</code>	—	引入套件並給簡稱	<code>import random as r</code>	—

§ 2 核心概念

核心概念

函式 (function) 是一組命名、可重複使用的指令，用它讓程式好寫、好讀、好維護。Python 的函式有三個來源：① **內建函式** (自帶、免 `import`，如 `len`、`max`、`sorted`、`abs`)；② **自定義函式** (自己用 `def` 寫)；③ **套件**裡的函式 (`import` 別人寫好的模組，如 `time`、`random`)。

寫自定義函式用 `def 名稱(參數):`，函式內容要**縮排**，用 `return` 回傳。兩個關鍵規則：**呼叫必須在定義之後** (否則 `NameError`)；**呼叫時要把參數備好**，缺必填參數會 `TypeError`，可用**預設值**避免。

最容易考也最容易錯的是**變數範圍**：函式內用 `=` **重新賦值**一個變數，會建立只在函式內有效的**區域變數**，不影響外面的全域變數；但若是直接改可變物件的內容 (如 `books[鍵]=值`、沒有重新賦值整個變數)，動到的就是**全域**那一份。

§ 3 主要內容

3.1 一、內建函式群

```
dir(__builtins__)          # 列出所有內建名稱
help(sorted)              # 查某函式的用法說明

zip(name, price)          # 把兩串列配對成元組，回傳「迭代器」，要 list() 才看得到
list(zip(['A', 'B'], [1,2])) # [('A', 1), ('B', 2)]

max([520, 780]); min([520, 780]); sum([520, 780]) # 780 / 520 / 1300
sorted([3, 1, 2])         # [1, 2, 3] 回傳新串列、不改原本
sorted([3, 1, 2], reverse=True) # [3, 2, 1] 降冪
```

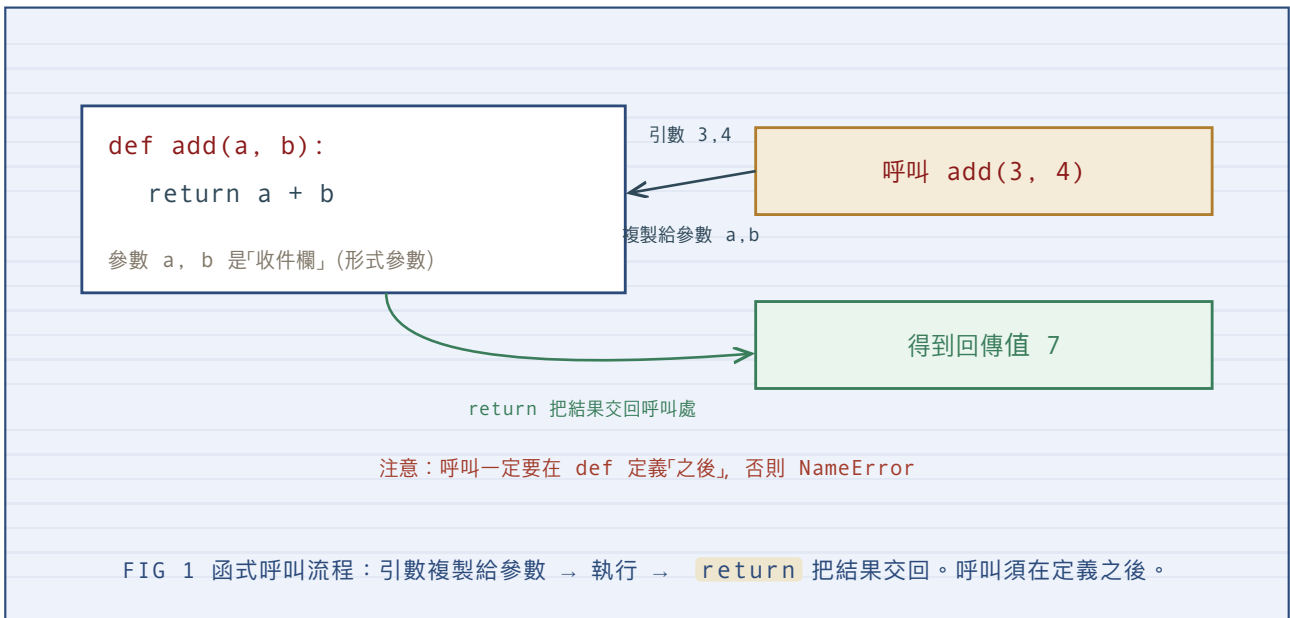
型別轉換內建函式 (`int` `float` `str` `list` `hex` `oct` `chr`) 與運算函式 (`abs` `divmod`)：

```
int(520.1)      # 520
float(520)     # 520.0
hex(520)       # '0x208'      oct(520) # '0o1010'
list('Py')    # ['P', 'y']
abs(-520)     # 520
divmod(1200, 780) # (1, 420) 回傳 (商, 餘數)
```

型別轉換陷阱

`int('520')` 可以 (字串內容是數字) ; 但 `int('Python')` 會 `ValueError: invalid literal for int()` 。轉換前要先確定字串內容真的能轉。

3.2 二、自定義函式：def、呼叫順序



```
def most_expensive(products):          # 通用函式：可重用於任何「名稱：價格」字典
    name = list(products.keys())
    price = list(products.values())
    return max(list(zip(price, name)))

books = {'Python手冊': 520, 'Java聖經': 780}
print(most_expensive(books))          # (780, 'Java聖經')
# most_expensive() 少傳 products → TypeError: missing 1 required positional argument
```

3.3 三、參數預設值

```
def most_expensive(products={'Python手冊': 520}): # 給預設值
    ...
most_expensive() # 沒傳 → 用預設值，不會 TypeError
most_expensive(books) # 有傳 → 用傳入的
```

有預設值的參數，要放在沒預設值參數的**右邊**。

3.4 四、變數範圍（招牌考點：重新賦值 vs 改內容）

```

全域 books = {... 'Python手冊': 520 ...}

```

```

def change_local():
    books = {...} # 重新賦值
    books['Python手冊']=1000
→ 建立「區域」books, 全域不變

```

```

def change_global():
    books['Python手冊']=1000
(沒有重新賦值整個變數)
→ 直接改「全域」books 的內容

```

關鍵分辨

- 用 = 重新賦值一個名稱 → 在函式內造出「區域變數」，外面不受影響
- 直接改可變物件內容 (books[鍵]=值, 沒重新賦值) → 動到的是全域那一份
- 函式內建立的區域變數, 函式「外」用會 **NameError**

FIG 2 變數範圍：= 重新賦值造區域變數；直接改可變物件內容動到全域。對應範例 8-16 / 8-17。

3.5 五、套件 import (time / random / sys)

```

import random as r          # 引入並給簡稱
import time as t

r.choice(['A','B','C'])     # 隨機抽一個
r.randint(0, 4)             # 隨機整數, 含 0 含 4
r.shuffle(students)        # 原地洗牌 (改變原串列)

t.asctime()                 # 'Sun Jan 27 18:42:33 2020'
t.localtime()              # struct_time, 可取 .tm_year/.tm_mon/...
t.time()                    # 1970 起的秒數, 常用來算執行時間
t.sleep(2)                  # 暫停 2 秒

import sys as s
s.version                   # Python 版本資訊

```

3.6 補充（老師標「不考」，僅供理解）

- **變數範圍 LEGB 規則**：Python 找變數的順序 Local → Enclosing → Global → Built-ins，都找不到才 NameError。
- **遞迴函式**：函式呼叫自己（要有終止條件），如河內塔。
- **傳函式當參數**：def f(data, func): return func(data)，把 max/min/len 當引數傳入。
- **匿名函式 lambda**：lambda x: x*0.8 一行運算式的小函式。

- `filter()` / `map()`：第一個參數是函式；`filter` 篩選、`map` 逐一套用轉換。

§ 4 語法與方法速查

```
# 內建函式
dir(__builtins__) help(f) zip(a,b) max(v) min(v) sum(v)
sorted(v) sorted(v, reverse=True)
int(x) float(x) str(x) list(x) hex(x) oct(x) abs(x) divmod(a,b)

# 自定義函式
def name(arg1, arg2=預設值): # 有預設值的放右邊
    ...
    return 值 # 可回傳多個 → 元組
# 呼叫要在定義之後；缺必填參數 → TypeError

# 變數範圍
x = 1 # 全域
def f():
    x = 2 # 重新賦值 → 區域（全域不變）
    d[鍵] = 值 # 改可變物件內容（沒重新賦值）→ 動全域

# 套件
import 套件 import 套件 as 簡稱
# time: asctime() localtime() time() sleep()
# random: choice() randint(a,b) shuffle()
# sys: version
```

§ 5 常見錯誤

常見錯誤

- **呼叫寫在 def 之前** → `NameError: name '...' is not defined`：先定義再呼叫。
- **缺必填參數** → `TypeError: missing 1 required positional argument`：補參數或給預設值。
- **預設值參數放錯位置**：有預設值的參數要放沒預設值參數的右邊。
- `int('Python')` → `ValueError`：轉換前確認字串內容能轉成數字。
- **變數範圍搞混**：函式內 `x = ...` 是區域變數（外面看不到）；要在函式外用就別期待它存在（`NameError`）。
- **誤以為函式內 = 能改全域**：`books = {...}` 在函式內是建區域變數；要改全域內容用 `books[鍵]=值`（不重新賦值整個變數）。
- `sorted()` **當成原地排序**：`sorted()` 回傳新串列、不改原本（與串列的 `.sort()` 不同）。
- `zip()` **直接 print 看不到內容**：它回傳迭代器，要 `list(zip(...))`。

§ 6 練習題

練習 1 (一般題)：zip + sorted 連動

引導步驟

1. 兩個串列 `names`、`prices`，用 `zip` 配對、`list()` 看結果。
2. 把 `(price, name)` 的配對用 `sorted()` 排序。

解答

```
names = ['Python手冊', 'Java聖經', 'C入門']
prices = [520, 780, 450]
print(list(zip(prices, names)))
print(sorted(zip(prices, names))) # 依價格升冪
```


練習 3 (重要題)：自定義函式 + return

引導步驟

1. 寫 `def price_stats(d)`: 回傳字典所有值的 (最高, 最低, 總和)。
2. 用 `return max(...), min(...), sum(...)` 一次回傳多個。
3. 呼叫並印出。

解答

```
def price_stats(d):  
    v = list(d.values())  
    return max(v), min(v), sum(v)  
  
books = {'Python手冊': 520, 'Java聖經': 780, 'C入門': 450}  
print(price_stats(books))    # (780, 450, 1750)
```

練習 4（一般題）：參數預設值與缺參數

引導步驟

1. 寫 `def greet(name='同學')`：印招呼語。
2. 分別 `greet()`（不給）與 `greet('阿堂')`（給），看差別。
3. 若把預設值拿掉，`greet()` 會怎樣？

解答

```
def greet(name='同學'):  
    print('你好，' + name)  
  
greet()           # 你好，同學（用預設值）  
greet('阿堂')    # 你好，阿堂  
# 若 def greet(name): 然後 greet() → TypeError: missing 1 required positional  
argument
```


§ 7 自我檢核

- 能說出函式三來源：內建、自定義（def）、套件（import）。
- 會用 `dir/help/zip/max/min/sum/sorted` 與型別轉換、`abs/divmod`。
- 知道 `sorted()` 回傳新串列不改原本、`zip()` 要 `list()` 才看得到。
- 會用 `def` 定義函式、`return` 回傳（含回傳多個）。
- 知道呼叫要在定義之後（否則 `NameError`）、缺必填參數 `TypeError`、可用預設值避免。
- 能分辨變數範圍：`=` 重新賦值造區域變數；改可變物件內容動全域。
- 會用 `import 套件 as 簡稱`，並認得 `time/random/sys` 常用函式。
- 知道 LEGB、遞迴、`lambda`、`filter/map` 老師標不考，有概念即可。