

## 第三章、字串與格式化輸出入

print()函式可以將電腦運算的結果在螢幕上輸出顯示, print()函式是 python 語言最常用的輸出函式, 可以將數值、字串…等資料, 以指定的格式在螢幕上顯示出來。使用 print()輸出函式時, 在小括弧()內可以使用多種引數, 其常用的語法如下:

```
print(v1, v2, v3, ..., sep = '間隔字串', end = '結尾字串')
```

### 練功坊

```
01 print('Python', 3.8)
02 print('台北', '台中', '台南', sep=',')
03 print('高雄', '屏東', sep='\t')
04 print('價目表:', end="")
05 money=30
06 print('陽春麵', money, '元')
```

print()函式要顯示的字串資料, 是用單引號或雙引號前後框住, 其中可以由一般字串、轉換字串和逃出序列三個部分組合而成。

1. **一般字串**：即為任何可顯示的字元組合, 如:A-z、a-z、0~9、!\*#\$△&…以及中文字元。

print()函式會將一般字串做完整的輸出。

2. **轉換字串**：所謂的格式字串輸出, 就是在一般字串輸出時, 在字串內指定位置插入指定的資料, 如此資料可套用格式輸出。其作法是在字串的指定位置用轉換字串, 轉換字串是由%轉換字元和型別字元所組合。Python 建議用字串物件的 format()方法來格式化字串, 但是轉換字串使用者仍然很多, 轉換字串位置用來插入引數串列對應的資料。語法如下：

%[修飾字元][寬度][.小數位數]型別字元

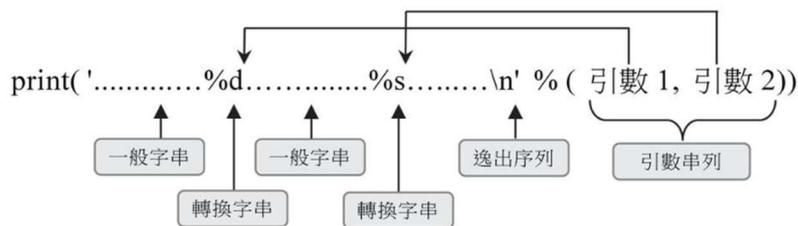
資料型別	%型別字元	說明
字元 字串	%c	顯示單一字元。
	%s	顯示字串。
	%%	顯示%字元。
整數	%d、%i	以十進位顯示整數。
	%o	以八進位顯示整數。
	%x、%X	以十六進位顯示整數。
浮點數	%f	以十進位顯示浮點數, 小數部分與設 6 位。
	%e、%E	以十進位科學記號顯示浮點數, 數值預設寬度為 8 位, 小數部分預設 6 位, 而 e 指數位數預設佔 2 位。
	%g、%G	

修飾字元：可以進一步設定輸出的格式, 常用的修飾字元如下表所示：

修飾字元	說明
#	配合二、八和十六進制，設定顯示 0b、0o、0x 等進制符號。
0	數值資料前面多餘的寬度補 0。
-	靠左對齊，預設值是靠右。
空白字元	會保留一個空格。

3. **逸出序列**：若格式字串內需要顯示一些特殊控制字元，如：雙引號「"」、單引號「'」、控制游標移動字元(跳格或跳到下一行行首)，可在特殊字元前加逸出字元「\」，當逸出字元加上控制字元就構成了逸出序列。下表為逸出序列的使用說明：

逸出序列	使用說明
\a	發出系統聲音。
\b	倒退鍵(Backspace)，游標會由\b所在位置向左移一個字元。
\f	換頁。
\n	換行，游標會由逸出序列所在位置跳到下一行的行首。
\r	移到行首，會刪除掉該行逸出序列所在位置前面的所有字元。
\t	水平跳格，相當於按 Tab 鍵。
\\	顯示倒斜線「\」字元。
\'	顯示單引號「'」字元。
\"	顯示單引號「"」字元。



### 練功坊

```
print('%f % 12345.67)
print('%f % -12.345)
print('%2f % 12.345)
print('%8.2f % -12.3456)
print('%3.1f % 123.45)
print('%#x' % 12345)
print('%08d' % 12345)
print('%-8d' % 12345)
print('% d' % 12345)
print('%s %d 打 %d 瓶' % ('汽水', 2, 24))
```

## 字串與運算子

使用「+」運算子，可以將兩個字串合併成一個字串。在 python 中，字串變數是屬於不可變動，所以使用「+」運算之後其實會產生新的字串。另外，字串不能用「+」運算子和其他型別資料合併，其他型別資料必須先用 str()函式進行型別轉換。

```
print('汽水：'+str(24)+'元')
>>>汽水：24 元
```

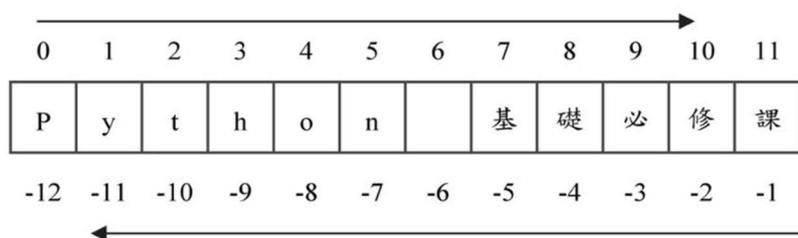
使用「\*」運算子，可以複製字串。對字串型別的資料乘上某一正整數時，就能讓字串重複該整數次數。

```
s1 = '-'
print(s1 * 10)
>>>-----
```

使用「in」運算子，若存在傳回值為 True；否則傳回 False。如果是使用 not in 運算子，就是反向查詢，傳回值會是相反的，若存在傳回值為 False；不存在則傳回 True。

```
print('a' in 'abcd')
>>>True
```

使用「[]」運算子可以分割字串，來取得單一字元或部分字串。[]必需配合字串註標值來運作，註標值可以使用正數或逆數(負數)來表示。如果是採正數，是從左向右由第 1 個元素從 0 開始編註標。若是採逆數，是從右向左由最後 1 個元素從 -1 開始逆數編註標。中文字和英文字母一樣是以 1 個元素計算。請看下面圖例，字串之上的數字是正註標，下方的數字是逆註標



使用「[]」運算子配合註標值讀取字串指定的單一字元，要注意註標值如果超出字串長度會產生 IndexError，語法如下：

字串變數名稱[註標值]

### 練功坊

```
s1 = 'Python 基礎必修課'
print(s1[3])
print(s1[-2])
print(s1[16])
```

除了取得單一的字元，也可以取得指定段落的字串，可以使用以下語法：

## 字串變數名稱[起始註標:結束標註:遞增值]

1. 起始註標、結束註標、遞增值皆可省略。
2. 起始註標是指定擷取的開始註標值，若省略代表從 0 開始。
3. 會擷取到結束註標的前一個字元，例如 3 表示取到註標值 2 的字元。結束註標如果省略，此時若遞增值為正值，代表擷取到字串尾端；反之若是負值，代表擷取到字串開頭。
4. 遞增值若省略，內定值為 1，代表連續擷取；如果是 2，代表間隔 1 個元素，也就是間隔值等於遞增值減 1。如果遞增值是正值，代表由左向右取出元素；反之是負值，則由右向左取出元素。
5. 註標值若超過字串範圍時，並不會產生 `IndexError`

0	1	2	3	4	5	6	7	8	9	10	11
P	y	t	h	o	n		基	礎	必	修	課

	輸出結果	說明
<code>s[:]</code>	Python 基礎必修課	字串元素從頭擷取到尾
<code>s[7:]</code>	基礎必修課	字串由註標值 7 擷取到字串結尾
<code>s[:6]</code>	Python	字串元素從頭擷取到註標值 5
<code>s[5:8]</code>	n 基	字串元素從註標值 5 擷取到 7
<code>s[9:6:-1]</code>	必修基	字串元素從註標值 9 逆向擷取到 7
<code>s[::2]</code>	Pto 礎修	字串從頭間隔 1 個字元擷取
<code>s[::-1]</code>	課修必修基 nohtyP	字串從尾到頭逆向擷取
<code>s[::-2]</code>	課必基 nhy	字串間隔 1 個元素逆向擷取

## 格式化字串 `str.format()` 方法

Python2.6 版後提供字串的 `format()` 方法，也可以用來格式化輸出內容。`format()` 是屬於字串物件的方法，其語法如下：

```
print("xxx{n}xxx".format(變數或常值))
```

字串內要代入的文字以大括弧「`{n}`」作代表，`n` 值由 0 開始依序編寫，執行時會依照編號，依序代入括弧內的引數；亦可不填編號，程式執行時會讀取對應引數列依序代入。引數列中的引數，可以使用常值或是變數。

### 練功坊

```
s1 = '電車月票'  
s2 = 1280  
print('項目:{0} 金額:{1}'.format(s1, s2))  
>>>項目：電車月票 金額：1280
```

大括弧內的編號或欄位名稱後，可以加上轉換字串來指定輸出格式，轉換字串於「:」後開始，包含了修飾字元、寬度、小數位數、型別字元，來強化輸出的功能，語法如下：

```
print('xxx{編號或欄位名稱:轉換字串}xxx'.format(參數列))  
print('xxx{編號或欄位名稱:[修飾字元][寬度][.小數位數]型別字元}xxx'.format(參數列))
```

範例	輸出結果	說明
'{:2f}'.format(0.666666)	0.67	浮點數，顯示兩位小數點。
'{:,}'.format(1000 * 1000)	1,000,000	以逗號分隔的數字格式。
'{:2%}'.format(0.666666)	66.67%	百分比的數字格式。
'{:0f}'.format(0.666666)	1	小數點以下四捨五入。
'{:6}'.format("字串")	字串	至少預留 6 位字元的空間，字串會靠左對齊。
'{:6}'.format(1000)	1000	至少預留 6 位字元的空間，數值會靠右對齊。
'{:>6}'.format("字串")	字串	「>」靠右對齊
'{:<6}'.format(1000)	1000	「<」靠左對齊
'\${:6}'.format(1000)	\$1000\$	「^」置中輸出，並且在空白處填滿「\$」。
'{0}{1}'.format('顯示{}')	顯示{}{}	顯示大括弧，可由參數代入或使用「{}」。

Python3, 6 版之後新增了格式化字串常值(Formatted String Literal)簡稱為 f-strings, 可以直接將運算式嵌入在字串常值中, 大大簡化格式化輸出的寫法, 其語法如下:

```
print(f'字串{變數名稱:轉換字串}')
```

### 練功坊

```
x, y = 10, 15
print (f' {x} + {y} = {x + y} ')
name = 'jack'
print (f'我的名字是 {name.capitalize()}')
```

## format() 函式

format() 函式是 Python 內建的函式, 可以用來格式化輸出內容, 其語法如下:

format(value, 格式字串)

- (1) value 引數是要格式化的資料, 資料型別可以是數值或字串。
- (2) 格式字串引數是指定格式的字串, 其用法和前面介紹字串的 format() 方法大致相同。

### 練功坊

```
print(format(123.45, '.1f'))
print(format('Python', '>8s'))
```

### 小試身手

一、words 變數值為 "ABCDEFGHJKLMNOPQRSTUVWXYZ", 請使用下列代碼回答下列問題的執行結果。

(A)DEFGHIJKLMN      (B)PMJGD      (C)CDEFGHIJKLMNO  
(D)ONMLKJIHGFE      (E)DGJM      (F)ZWTQNKHEB

(1) words[3:14] 的輸出結果為何?  
(2) words[3:14:3] 的輸出結果為何?

(3) words[15:2:-3]的輸出結果為何?

(4) words[::-3]的輸出結果為何?

解答：(1) A (2) E (3) B (4) F

二、餐廳需要一個服務評分程式，程式需要執行下列程式工作：

- 接受輸入評分(score)
- 傳回評分的平均(average)
- 平均分數四捨五入到小數兩位

請回答下列問題來完成程式：

```
total = num = stop = 0
```

```
average = 0.0
```

```
while stop != -1:
```

```
    score = ____ (1) ____
```

```
    if score == -1:
```

```
        break
```

```
    total += score
```

```
    num += 1
```

```
average = float(total/num)
```

```
____ (2) ____ + ____ (3) ____
```

(1) 請從下列選項中選取正確程式碼填寫至\_\_\_\_ (1) \_\_\_\_。

(A) float(input("請輸入 1-10 的評分(-1 離開)"))

(B) print("請輸入 1-10 的評分(-1 離開)")

(C) input("請輸入 1-10 的評分(-1 離開)")

(D) input"請輸入 1-10 的評分(-1 離開)"

(2) 請從下列選項中選取正確程式碼填寫至\_\_\_\_ (2) \_\_\_\_。

(A) printline("平均給分為：" (B) console.input("平均給分為:")

(C) print("平均給分為：" (D) output("平均給分為:")

(3) 請從下列選項中選取正確程式碼填寫至\_\_\_\_ (3) \_\_\_\_。

(A) {average, '.2f}) (B) format.average. {.2f})

(C) format(average, '.2d') (D) format(average, '.2f )

解答：(1) A (2) C (3) D

三、下列程式碼執行後的輸出結果會是哪一個選項？

```
a = "薯條"
```

```
b = "熱咖啡"
```

```
c = "雞腿堡"
```

```
s = "{1}加{0}加{2}"
```

```
print(s.format(a, b, c))
```

(A) 熱咖啡 加 薯條 加 雞腿堡 (B) 薯條 加 熱咖啡 加 雞腿堡

(C) 雞腿堡 加 薯條 加 熱咖啡 (C) 薯條 加 雞腿堡 加 熱咖啡

四、您正在建立一個產品售價輸入程式，並以逗點分隔格式輸出資料，輸出格式需求如下：

- 以雙引號括住字串資料。
- 不以引號或其他字元括住數值資料。
- 以逗點分隔項目。

```
itemName = input('請輸入產品名稱:')
```

```
price = input('請輸入售價:')
```

請問下列哪兩個敘述片段，符合上述輸出格式需求？

(A) `print("{0}", {1}'.format(itemName, price))`

(B) `print("{0}, {1}".format(itemName, price))`

(C) `print(""+itemName+", "+price)`

(D) `print(itemName+', '+price)`

解答：A、C