

## 一、變數與資料型態

1. Python 變數不需要宣告資料型態就可以使用，語法為：

```
變數名稱 = 值
變數名稱 1 = 變數名稱 2 = 值
變數名稱 1,變數名稱 2 = 值 1,值 2
```

2. Python 變數其資料型態會根據變數值決定，例如：

```
score = 80      #資料型態為數值，變數值為「80」
avg = 60.5     #資料型態為浮點數，變數值為「60.5」
fruit = "蘋果" #資料型態為字串，變數值為「蘋果」
fruit = '蘋果' #資料型態為字串，變數值為「蘋果」
fruit = '"蘋果"' #資料型態為字串，變數值為「"蘋果"」
result = True  #資料型態為布林，變數值為「True」
result = False #資料型態為布林，變數值為「False」
```

3. Python 變數命名必須遵守一定的規則：

- 第一個字元必須是大小寫字母、\_、中文
- 只能由大小寫字母、數字、\_、中文組成
- 英文字母大小寫視為不同變數名稱
- 不得使用保留字

```
7eleven      #錯誤！第一個字元不能是數字
George&Mary  #錯誤！不能包含特殊字元
George Mary  #錯誤！不能包含空白字元
if           #錯誤！if 為 Python 保留字
```

4. 若字串中需含有特殊字元，可以使用脫逸字元：

脫逸字元	意義	脫逸字元	意義
\'	單引號「'」	\"	雙引號「"」
\\	反斜線「\」	\n	換行
\t	定位(Tab 鍵)		

例如：

```
str3 = "大家好！ \n 歡迎光臨！"    #「歡迎光臨」會顯示於第二列
```

5. Python 強制資料型態轉換的指令有：

- `int()`：強制轉換為整數資料型態。
- `float()`：強制轉換為浮點數資料型態。
- `str()`：強制轉換為字串資料型態。
- `bool()`：強制轉換為布林資料型態。
- `eval()`：強制將運算式轉換為數值資料型態。

## 二、輸出

1. print 命令能輸出指定項目的內容，語法為：

```
print(項目 1 [, 項目 2,……, sep=分隔字元,end=結束字元])
```

- 項目 1, 項目 2,……：可輸出多筆資料，資料間以逗號隔開。
- sep：設定項目之間的分隔符號，預設值為 1 個空白字元。
- end：設定輸出完畢後的字元，預設值為換列字元(\n)。

例如：

```
print("多吃水果")           #多吃水果
print(100, "多吃水果", 60)   #100 多吃水果 60
print(100, "多吃水果", 60, sep("&")) #100&多吃水果&60 (下一個 print 從下一列顯示)
print(100, 60, sep("&", end="")) #100&60 (下一個 print 於同一列顯示)
```

2. print 支援參數格式化功能，即以「%s」代表字串、「%d」代表整數、「%f」代表浮點數，語法為：

```
print("項目" % (參數列))
```

例如：

```
name = "小明"
score = 60
print("%s 的成績為 %d" % (name,score)) #小明 的成績為 60
```

參數格式化，可以精確控制輸出位置：

- %5d：固定輸出 5 個字元，若少於 5 位數，會在數字左方填入空白字元；若大於 5 位數則全部輸出。
- %5s：固定輸出 5 個字元，若少於 5 字元，會在字串左方填入空白字元；若大於 5 字元則全部輸。
- %8.2f：固定輸出 8 個字元(含小數點)，小數固定輸出 2 位數。若整數少於 5 位數(8-3=5)，會在數字左方填入空白字元；若小數少於 2 位數，會在數字右方填入「0」字元。

```
price = 23.8
print("價格為%8.2f" % price) #價格為△△△23.80 (△代表空白)
```

3. print 亦可以使用 format 做格式化，語法為：

```
print("{引數}".format(參數列))
```

例如：

```
name = "小明"
score = 60
print("{0} 的成績是 {1}分".format(name,score)) #小明 的成績是 60 分
print("{} 的成績為 {}分".format(name,score)) #省略引數
```

**【注意 1】**不能省略的情況有兩種：

1. 左邊的引數，未按照右邊順序來取得值。
2. 左邊的引數，多於右邊的值。

```
print("{1} 的成績是 {0}" .format(name,score)) #60 的成績是 小明
print("{0} 的成績是 {0}" .format(name)) #小明 的成績是 小明
```

**【注意 2】** 引數可用於控制格式：

```
print("{編號:填充字元 對齊方式 欄寬 千分位 精度 資料型態}".format(參數列))
```

其中資料型態如下：整數：**d** (可省略)、字串：**s** (可省略)、浮點數：**f** (不可省略)

例如：

```
sum = 1345
str = "1345"
print("{:6}".format(sum))      #△△1345 (△代表空白)
print("{:06}".format(sum))    #001345
print("{:0>6}".format(str))   #001345
print("{:0<6}".format(str))   #134500
print("{:0^6}".format(str))   #013450
print("{:,}".format(sum))     #1,345
print("{:06,}".format(sum))   #01,345
print("{:09,.2f}".format(sum)) #01,345.00
```

### 三、輸入

1. `input` 命令能輸入指定項目的內容，語法為：

```
變數 = input([提示字串])
```

例如：

```
score = input("請輸入成績")
```

**【注意】** `input` 命令的資料型態預設為 **字串**！如要進行計算，需要搭配資料型態轉換。

例如：

```
score = int(input("請輸入成績:"))
```

2. `input` 命令同時輸入 **多個** 指定項目的內容，語法為：

例如：

```
a,b = input("請輸入成績:")      #輸入資料時，資料之間必須使用「,」隔開
a,b = eval(input("請輸入成績:")) #轉換成數值資料型態，必須使用 eval()
```

3. `input` 命令同時輸入 **特殊格式** 指定項目的內容，語法為：

```
a,b = input("請輸入時間:").split(":") #輸入時間格式「9:45」，分別將 9 指派給 a；45 指派給 b
a,b = map(int,input("請輸入時間:").split(":")) #轉換成數值資料型態，必須使用 map()
```

**【註 1】** `split()` 可指定分隔符號，進行字串切割。例如：「9:45」，兩個值可使用「:」來做切割

**【註 2】** `map()` 可對 `list` 裡面的每一個元素，進行運算或轉換資料型態。