

# 0\_1\_DataType

## Data Type

Name	Type	Note
integer	int	整數 1, 2
floating point	float	(浮點數, 有小數點) 1.2
boolean	bool	布林值 True或False
string	str	字串 "aaa" '123'

10、100

文字”你好”

數值型態 (Numeric type) : int, float, bool

字串型態 (String type) : str

## Numeric Operation

混合運算：會取範圍較廣的運算方式  
避免小數點不見

不同型態間的運算，自動轉換成範圍比較大的型態

int < float

```
1 | print(4 + 2)      # 6
2 | print(4.0 + 2)    # 6.0
3 | print(4.0 + 2.0)  # 6.0
4 | print(2 / 5)      # 0.4
5 | print(2.0 / 5)    # 0.4
6 | print(2 / 5.0)    # 0.4
7 | print(2.0 / 5.0)  # 0.4
8 | print(2 // 5)     # 0
```

/：產生的結果一定是浮點數，即使剛好整除  
//：只需要商的整數部分，須用兩條斜線

## Data Type Example

```

1  iv = 10    # 放入變數 iv (整數 int)
2  fv = 12.3  # 放入變數 fv (浮點數 float)
3  sv = 'hello python' #放入變數 sv (字串 str)
4  bv = True  # 放入變數 bv (布林值 bool)
5
6  print('iv :', iv, '; fv :', fv, '; sv :', sv, '; bv :', bv)
7  print(type(iv)) # type(iv)會得到 iv 的形態 ==> int
8  print(type(fv)) # type(fv)會得到 fv 的形態 ==> float
9  print(type(sv)) # type(sv)會得到 sv 的形態 ==> str
10 print(type(bv)) # type(bv)會得到 bv 的形態 ==> bool

```

```

iv : 10 ; fv : 12.3 ; sv : hello python ; bv : True
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>

```

## str - - string type (字串型態)

字串：一連串的字元

可用單引號、或是雙引號，把它包起來

用 `'...'` 和 `"..."` 來宣告字串  
字串，顧名思義就是一串字

```

1  str1 = '12345'      # 用 '...' 宣告字串 str1
2  str2 = "hello world" # 用 "..." 宣告字串 str2
3  print(f"str1 is {str1} {type(str1)}") # 印出 str1 的內容與型態
4  print(f"str2 is {str2} {type(str2)}") # 印出 str2 的內容與型態

```






內外用不同引號區分

```

str1 is 12345 <class 'str'>
str2 is hello world <class 'str'>

```

## Example String?

Example	String?
"2" (with double-quotes)	Yes 
'3.6' (with single-quotes)	Yes 
"True" (also in quotes)	Yes 
7 (integer)	No 
Hello (no quotes)	No 

True (boolean)

No ❌

## 跳脫字元(Escape Character)

\字元 具有特殊函義  
\n 換行  
\t tab 空格  
\" 在字串中表示單純的 "  
' 在字串中表示單純的 '

\"

"字串" 中有 "

```
1 | "It is a word "test"." # 用 "..." 宣告的字串有 "
```

```
File "<ipython-input-2-07ecadeb4421>", line 1
    "It is a word "test"." # 用 "..." 宣告的字串有 ""
                        ^
SyntaxError: invalid syntax
```

程式中的第一個引號 " 視為字串的開始，並將第二個引號 " 視為字串的結束

\'

'字串' 中有 '

```
1 | 'It is a word 'test'.' # 用 '...' 宣告的字串有 '
```

```
File "<ipython-input-4-dc7342a6fb9d>", line 1
    'It is a word 'test'.' # 用 '...' 宣告的字串有 '
                        ^
SyntaxError: invalid syntax
```

程式中的第一個引號 ' 視為字串的開始，並將第二個引號 ' 視為字串的結束

"字串" 中有 " 使用 \"

```
1 | "It is a word \"test\")." # 用 "...\" 宣告的字串有 "
```

```
'It is a word "test").'
```

'字串' 中有 ' 使用 \"

```
1 | 'It is a word \"test\').' # 用 '...\" 宣告的字串有 '
```

```
"It is a word 'test')."
```

## 字串內外符號不同

```
1 | str1 = "It is a word 'test')." # 用 "...\" 宣告的字串有 '  
2 | str2 = 'It is a word "test").' # 用 '...\" 宣告的字串有 "  
3 | print(str1, "\t", str2)
```

```
It is a word 'test'.      It is a word "test").
```

## 字串 +

字串中的 + 號：連接字串用

字串 + 字串 ==> 字串串接

```
1 | name = "May"  
2 | print("Hello " + name + " :)")  
3 |  
4 | gretting = "Hi, " + name  
5 | print(gretting + " :)")
```

```
Hello May :)  
Hi, Amy!  :)
```

## 字串 \*

\*：重複字串用

字串 \* n ==> 字串重複 n 次

```
1 hi = "Hi. " # 這是字串 hi
2 hi3 = hi * 3 # 重複3次 hi 的內容
3 print("hi :", hi)
4 print("hi3 :", hi3)
```

```
hi : Hi.
hi3 : Hi. Hi. Hi.
```

## in 運算符號

"字串A" in "字串B": 字串A 是否在 字串B 中

```
1 string = "Emma is there"
2 print("Emma" in string) # "Emma" 是否在 string 中
3 print("emma" in string) # "emma" 是否在 string 中
```

```
True
False
```

## String method

附屬在字串後面的小功能

[The Python Standard Library » Built-in Types str](#)

[Python 標準函式庫內建型別 str](#)

[Python String Methods w3schools](#)

## String method

strip : 幫忙去除字串頭尾多餘的空白字元

```
1 s = "    hello world.  "
2 print(f"-{s}-")
3 print(f"-{s.strip()}-") # 去除字串頭尾的空白
4 print(f"{s.strip().capitalize()}") # 第一個字母變成大寫
5 print(f"{s.strip().title()}")    # 轉換成為標題字串
```

```
-    hello world.  -
-hello world.-
```

```
Hello world.  
Hello World.
```

## String method

len : 可以取的字串長度 = 有幾個字元

```
1 s = "Hello World"  
2 print(s)  
3 print(len(s))      # s 的長度  
4 print(s.split())    # 將 s 依照空格分開  
5 print(s.lower())    # 轉小寫  
6 print(s.upper())    # 轉大寫  
7 print(s.isdigit())  # 判斷是否由數字組成
```

split :

```
Hello World  
11  
['Hello', 'World']  
hello world  
HELLO WORLD  
False
```

## String method

```
1 s = "Hello World"  
2 print(s)  
3 print(s.isnumeric()) # 判斷是否為合法的數值  
4 print(s.isalpha())   # 判斷是否由字母組成  
5 print(s.find('World')) # 找出 s 中 world 出現的位置  
6 print(s.replace('World', 'Python')) # 將 s 中的 world 換成 python
```

```
Hello World  
False  
False  
6  
Hello Python
```

replace : 可以字串中的某一部份換成另外的內容  
建立全新的字串  
就的字串無法被更改

## Python 字串無法被改變

```
1 word = 'Python'  
2 word[0] = 'J'
```

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-2-0dd74ce406e6> in <module>
      1 word = 'python'
----> 2 word[0]='j'

TypeError: 'str' object does not support item assignment

```

## 取字串 String select and slice

	H	E	L	L	O	,	W	O	R	L	D	
INDEX	0	1	2	3	4	5	6	7	8	9	10	
INDEX	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	
	0	1	2	3	4	5	6	7	8	9	10	11
	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	

`s[index]` : 取出 `s` 字串裡面的第幾個數字的字元，索引值(index)是從 0 開始

```
1 | s[0] # 取出 index = 0 的字，第一個字 H
```

`s[star_index:stop_index]` : 取出某一個範圍的字串，**不包含**`stop_index` 的字元

```
1 | s[0:3] # 取出 index 0 至 2 的字串，HEL
```

```

1 | s = "HELLO , WORLD"
2 | print('s :', s)
3 | print('s[0] ', s[0]) # 取 index = 0 第一個字 H
4 | print('s[-11]', s[-11]) # 取 index = -11 第一個字 H
5 | print('s[4] ', s[4]) # 取 index = 4 第五個字 O
6 | print('s[-1] ', s[-1]) # 取 index = -1 最後一個字 D

```

```

s : HELLO , WORLD
s[0] H
s[-11] H
s[4] O
s[-1] D

```

```

1 | s = "HELLO , WORLD"
2 | print('s :', s)
3 | print('s[2:5]', s[2:5]) # 取 index 2-5 [2,3,4] 第2到第4
4 | print('s[2:] ', s[2:]) # 從 index 2到底
5 | print('s[:5] ', s[:5]) # 從 開頭取到第5
6 | print('s[:-2]', s[:-2]) # 從 開頭取到倒數第2

```

```

s : HELLO , WORLD
s[2:5] LLO
s[2:] LLO , WORLD
s[:5] HELLO
s[:-2] HELLO , WOR

```

## Type Convert 型態轉換

不同的形態不能運算  
例如：字串+數值 ...

```

1 | a = 123      # a is int
2 | b = '456'   # b is str
3 | print(a+b)  # a+b ==> int + str has error

```

```

-----
TypeError                                Traceback (most recent call last)
<ipython-input-1-895d3406aa84> in <module>
      1 a = 123
      2 b = '456'
----> 3 print(a+b)

TypeError: unsupported operand type(s) for +: 'int' and 'str'

```

## Type Convert

Operation	Result
int(x)	x converted to integer
float(x)	x converted to floating point
str(x)	x converted to string



```
1 a = 123          # a is int
2 b = '456'        # b is str
3 print(a+int(b))  # int(b) ==> int + int
4 print(str(a)+b)  # str(a) ==> str + str
```

```
579
123456
```

---

## 計算成績 example

---

```
1 student_name = '李大中' # 學生名稱
2 ch_score     = 98       # 國文成績
3 en_score     = 95       # 英文成績
4 math_score   = 90       # 數學成績
5
6 # 計算總分和平均
7 total = ch_score + math_score + en_score # 總分
8 average = total / 3 # 平均
9
10 # 顯示學生姓名和成績
11 print('學生姓名:', student_name)
12 print('國英數成績:', ch_score, en_score, math_score)
13 print('平均成績:', average)
```

```
學生姓名: 李大中
國英數成績: 98 95 90
平均成績: 94.33333333333333
```

---

## 練習 - 四則運算

---

請寫一支Python程式，能讀取兩個整數，並把他們的四則運算印出來。四捨五入取到小數點第二位

---

## 練習 - 計算 BMI

---

身體質量指數 (Body Mass Index, BMI)  
BMI = 體重 (公斤) 除以身高 (公尺) 的平方

請計算體重 (70公斤)和身高(170公分)的BMI

請宣告體重，身高，BMI三個變數

BMI請四捨五入取到小數點第二位

---

## 四捨五入 `round()`

---

`round(number)` # 預設無小數

`round(number, 小數後幾位)`

```
1 print('round(59.4)', round(59.4))
2 print('round(59.5)', round(59.5))
3 sc = 5954/100
4 print('sc', sc, 'round(sc)', round(sc))
5 print('sc', sc, 'round(sc,1)', round(sc, 1))
6 print('sc', sc, 'round(sc,2)', round(sc, 2))
```

```
round(59.4) 59
round(59.5) 60
sc 59.54 round(sc) 60
sc 59.54 round(sc,1) 59.5
sc 59.54 round(sc,2) 59.54
```