
變數的概念與型態

變數 **VARIABLE**

名稱 + 型態 + 值



變數型態

- 數值型態 **Numeric** : **int, float, boolean(bool), complex**
- 字串型態 **String** : **string(str)**
- 容器型態 **Container** : **tuple, list, dict, set**



```
type(3.14)
```

數值型態

Int, Float, Boolean(Bool), Complex

算術運算

運算子	說明	範例
+	加	$3 + 4 = 7$
-	減	$3 - 4 = -1$
*	乘	$3 * 4 = 12$
/	浮點數除法	$3 / 4 = 0.75$
//	整數除法（無條件捨去）	$3 // 4 = 0$
%	相除後求餘數	$3 \% 4 = 3$
**	次方	$3 ** 4 = 81$

算術運算

- $5 // 3$
- $3.14 // 2$
- $2^{**} 0$
- $0^{**} 0$
- $2^{**} .5$
- $2^{**} -.5$
- $-1 \% 3$
- $5.5 \% 3$
- $-5.5 \% 3$
- $2.5 \% .5$

算術運算

- $0 / 1$
 - $0 * .1$
 - $2 ** 2$
 - $5 // 2$
 - $5.5 // 2$
 - $4 \% 2$
 - $2.5 \% .5$
 - $5 / \text{True}$
 - $\text{False} / \text{True}$
 - $0 * 1.5 + 0 * 3j$
-

字串型態

String(Str)

字串運算

運算子	說明	範例
+	加	‘a’ + ‘b’ = ‘ab’
*	乘	a’ * 3 = ‘aaa’

容器型態

Tuple, List, Dict, Set

LIST 串列

- 可重複
- 可修改
- 有順序性（可使用序號索引）

```
empty_list_1 = []  
empty_list_2 = list()  
fruits = ['apple', 'banana']  
animals = [  
    'cat',  
    'cat',  
    'dog',  
]  
  
print(fruits[0])    # apple  
fruits[0] = 'orange'  
print(fruits[0])    # orange
```

TUPLE 數組


- 可重複
- 不可修改
- 有順序性（可使用序號索引）

```
empty_tuple_1 = ()
empty_tuple_2 = tuple()
fruits = ('apple', 'banana')
animals = (
    'cat',
    'cat',
    'dog',
)

print(animals[0])    # cat
animals[0] = 'dog'   # error
```

DICTIONARY 字典


- **Key** 不可重複
- 可修改
- 無順序性（可使用 **Key** 索引）



```
empty_dict_1 = {}  
empty_dict_2 = dict()  
  
fruit = {  
    'name': 'apple',  
    'price': 20,  
}  
  
print(fruit['name'])    # 'apple'  
print(fruit['price'])   # 20
```

SET 集合

- 不可重複
- 可修改
- 無順序性（無法索引）



```
empty_set = set()

fruits = set(
    'apple',
    'banana',
    'banana',
)

print(fruits)
# 'apple', 'banana'
```

型態轉換

TYPE CONVERSION(CASTING)

數值型態與字串型態轉換



```
int('123')    # 123
int(3.14)     # 3
float(123)    # 123.0
str(123)      # '123'
str(3.14)     # '3.14'
```

程式結構

區塊概念

- 區塊內變數共享，分成全域與區域
- 區塊執行順序由全局開始（由上而下，由左而右）
- 定義區塊並不會被執行，條件區塊依條件被執行

```
2
3
4 def echo(message):
5     print(message)
6     return message
7
8
9 def main():
10     # 印出今天日期
11     print(datetime.date.today())
12     # 簡單迴圈與條件判斷示範
13     num = 5
14     for i in range(1, 10):
15         if num > i:
16             echo("num > i")
17         else:
18             echo("num < i")
19
20
21 if __name__ == "__main__":
22     main()
```

縮排規定

- **Python** 以縮排劃分區塊，不可有縮排錯誤情形
- 相同區塊，相同縮排數量
- 子區塊比父區塊多一個縮排
- 縮排為四個空格，按 **Tab** 會自動轉換

```
2
3
4 def echo(message):
5     print(message)
6     return message
7
8
9 def main():
10     # 印出今天日期
11     print(datetime.date.today())
12     # 簡單迴圈與條件判斷示範
13     num = 5
14     for i in range(1, 10):
15         if num > i:
16             echo("num > i")
17         else:
18             echo("num < i")
19
20
21 if __name__ == "__main__":
22     main()
```