

認識變數與基本數學運算

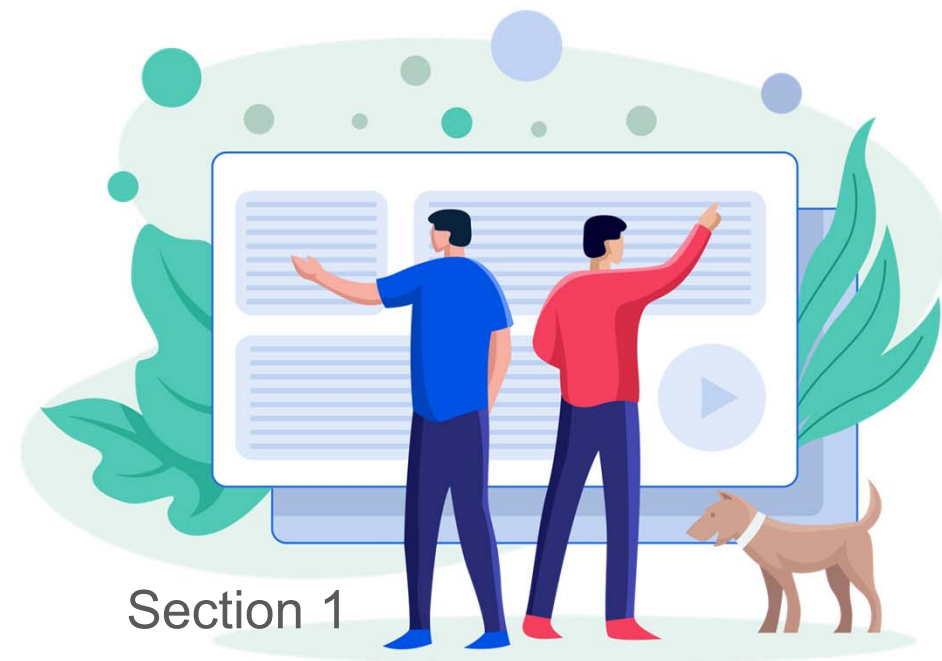
學習目標

- 說明變數的用法
- 介紹算術運算子(Arithmetic Operators)
- 介紹指定運算子(Assignment Operators)



Python 的變數

- 介紹 Python 如何使用變數
- Python 的關鍵字 (Keywords) 或保留字 (Reserved words)



Section 1

Python 識別字

- 識別字是用來識別變數、函數、類別、模組與其他物件
- 第一個字必須是文字或底線 (_)，第二個字以後可以是文數字或底線
- Python 的識別字內不允許特殊字元，例如 @ 、 \$ 、 % ，也不可以有 hyphen 符號 (-)
- Python 為有區分大小寫的程式語言，Hello 與 hello 是不同的識別字

Python 關鍵字(保留字)

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

列出 Python 的保留字

- `import keyword # 匯入 keyword 模組`
- `print(keyword.kwlist)`
- `['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']`
- `keyword.iskeyword('if') # True`
- `keyword.iskeyword('IF') # False`

Python identifier 使用(變數)識別字

- Python 不需要明確宣告變數來保留記憶體空間
- 當【指定】某個值給變數時，Python 就自動【宣告】變數
- 用等號 (=) 來指定值給變數
 - `total = 10` # An integer assignment 指定整數
 - `miles = 477.5` # A floating point 指定浮點數
 - `name = 'John'` # A string 指定字串
 - `x = y = z = 1` # Multiple assignments
 - `a, b, c = 1, 2, "John"` # Multiple assignments
- Python變數名稱的风格一般是以小寫為主，除非是做類別(Class)的定義
- 在Python中，單引號(')與雙引號(")是相同的；使用上必須成對使用

檢查是否為合格識別字

- `print("abc".isidentifier())` # True
- `print("99a".isidentifier())` # False
- `print("_".isidentifier())` # True
- `print("for".isidentifier())` # True

Python identifier

name	valid or invalid
ab10c	valid
abc_DE	valid
_	valid
_abc	valid
99	invalid
x+y	invalid
for	valid (not recommend)
a@	invalid
9abc	invalid

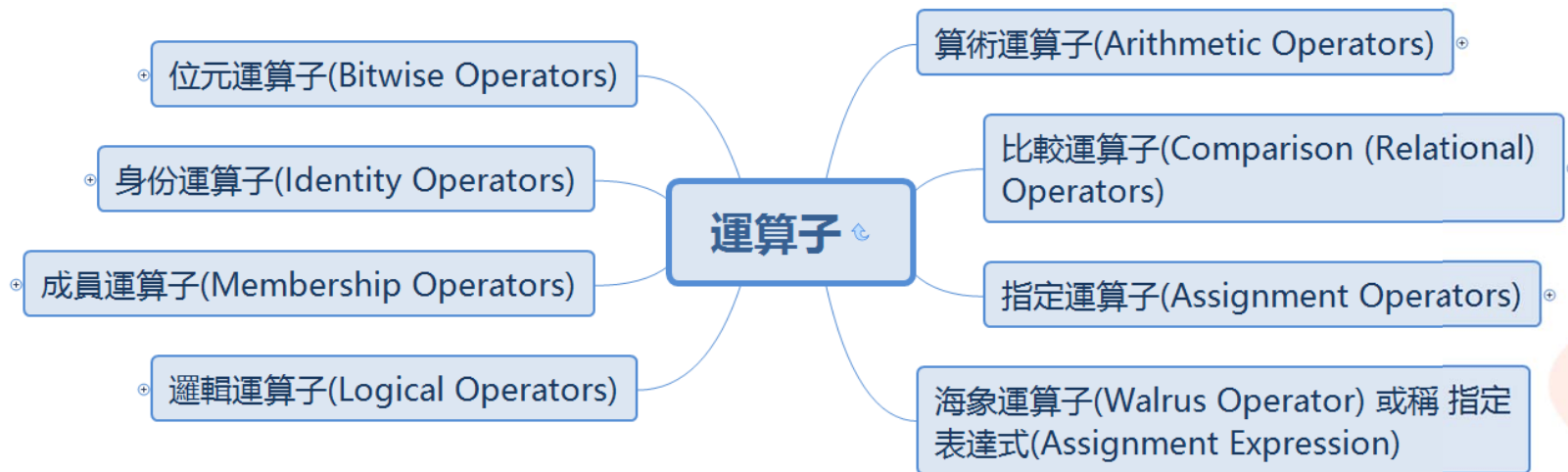
使用變數(Demo)

■使用 Python 互動式環境使用變數



Python算術運算子

■使用算術運算子(Arithmetic Operators)



運算子類型

■Python 語言支援以下運算子

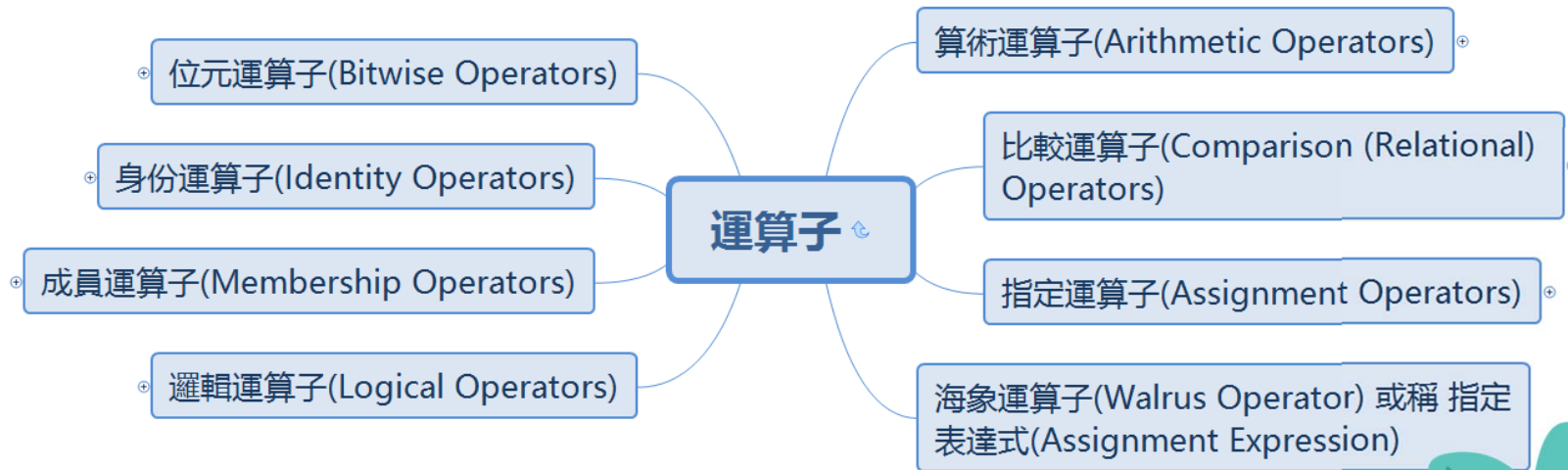
1. 算術運算子(Arithmetic Operators)
2. 比較運算子(Comparison (Relational) Operators)
3. 指定運算子(Assignment Operators)
4. 邏輯運算子(Logical Operators)
5. 成員運算子(Membership Operators)
6. 身份運算子(Identity Operators)
7. 位元運算子(Bitwise Operators)
位元運算子通常是「繪圖」或「硬體控制」等需要處理二進制或十六進制的資料時，才會使用到
8. 海象運算子(Walrus Operator) 或稱 指定表達式(Assignment Expression) (Python 3.8)

算術運算子

運算子	說明	舉例
+	加法或正號	+7; 12 + 34
-	減法或負號	-7; 34 - 12
*	乘法	16.4 * 5
/	除法	17 / 6
%	求餘數	17 % 6
**	次方	9 ** 2
//	整數除法，結果為整數，捨棄小數部分 如果有一個運算元為負數，則結果為地板除法(floored)，取較小的結果	9 // 2 = 4; 9.0 // 2.0 = 4.0 -9 // 2 = -5; -9.0 // 2 = -5.0

Python指定運算子

■使用指定運算子(Assignment Operators)



指定運算子

運算子	說明	舉例
=	單純指定運算	c = a + b
+=	加法與指定運算	c += a 相當於 c = c + a
-=	減法與指定運算	c -= a 相當於 c = c - a
*=	乘法與指定運算	c *= a 相當於 c = c * a
/=	除法與指定運算	c /= a 相當於 c = c / a
%=	求餘數與指定運算	c %= a 相當於 c = c % a
**=	次方與指定運算	c **= a 相當於 c = c ** a
//=	地板與指定運算	c //= a 相當於 c = c // a

Examples

■ $a += 5$

相當於

$a = a + 5$

■ $a /= 10$

相當於

$a = a / 10$

■ $a *= b$

相當於

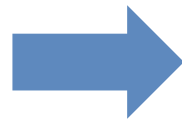
$a = a * b$

海象運算子(Walrus Operator) 或稱 指定表達式(Assignment Expression)

- 海象運算子(:=) (The Walrus Operator) 的功能是 可以在使用未賦值的變數時，使用海象運算子(:=)來賦值

未使用海象運算子寫法

```
boolUser = False  
print(boolUser) # Fasle
```



使用海象運算子寫法

```
print(boolUser := True) # True
```

海象運算子(Walrus Operator) 或稱 指定表達式(Assignment Expression)

未使用海象運算子寫法

```
list1 = list()
current = input('Write something: ')
while current != 'quit':
    list1.append(current)
    current = input('Write something: ')
```

使用海象運算子寫法

```
list1 = list()
while (current := input('Write something: ')) != 'quit':
    list1.append(current)
```

不使用海象算子的情境

- 如果要賦值，直接使用「指定運算子」
- 簡化程式(在python的function內賦值)，使用海象運算子

```
name:='Benjamin'
```

```
File "C:\Users\Benjamin\AppData\Local\Temp\ipykernel_26892\136562968.py", line 1
    name:='Benjamin'
          ^
SyntaxError: invalid syntax
```

要賦值請用指定運算子，若直接使用海象運算子會有語法錯誤

```
print(name:='Benjamin')
```

Benjamin

海象運算子僅適用於執行Python函數時(如print)，同時賦值的情境

基本數學運算(Demo)

■使用 Python 互動式環境做基本算術運算



```
■>>> 23 + 45
```

```
■>>> 59 - 22
```

```
■>>> 43 * 3
```

```
■>>> 8 ** 2
```

```
■>>> 16 / 4
```

```
■>>> 16 // 4
```

```
■>>> 16 // 4.0
```

```
■>>> 15 % 4
```

```
■>>> 15 % 4.0
```

```
■>>> x = 5
```

```
■>>> x
```

```
■>>> x = x + 1
```

```
■>>> x
```

```
■>>> x += 1
```

```
■>>> x
```

```
■>>> --x
```

```
■>>> x--
```

```
■>>> y = 3
```

```
■>>> x, y
```

```
■>>> y **= x
```

```
■>>> x, y
```

```
■>>> 33 ** 999
```

```
■>>> 33.0 ** 999
```

重點精華回顧

- Python 保留字
- Python (變數)識別字
- Python 算術運算子
- Python 指定運算子
- Python 海象運算子

