

Python

運算

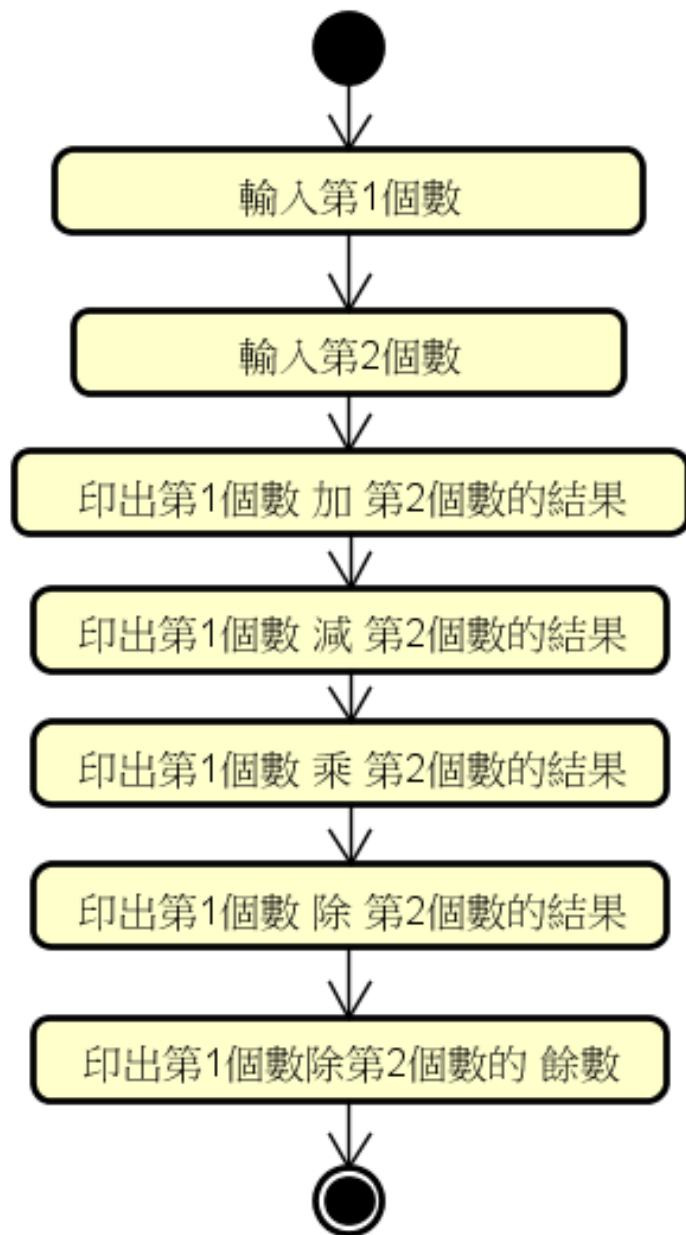


寫一個程式，
輸入2個整數
印出兩數加、減、乘、除與餘數等7則運算
的結果

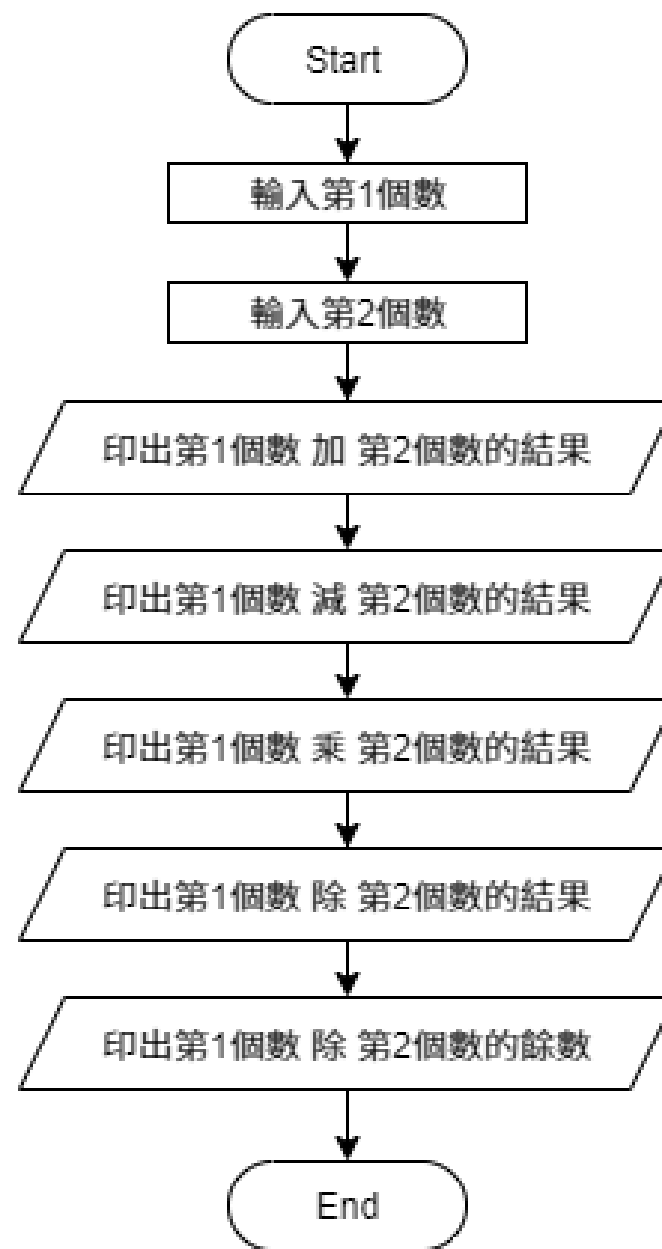
問題



流程



UML流程圖



傳統流程圖

數學的四則運算

符號	功能
+	加
-	減
*	乘
/	除

Python 的 ? 則運算

符號	功能
+	加
-	減
*	乘
/	浮點除法
%	取餘數
**	指數次方
//	整數除法

Python 的 7 則運算

符號	功能	語法	實例
+	加	$x + y$	$10 + 3 = 13$
-	減	$x - y$	$10 - 3 = 7$
*	乘	$x * y$	$10 * 3 = 30$
/	浮點除法 (強迫使用 <code>float()</code> 轉型)	x / y	$10 / 3 = 3.333$
%	取餘數	$x \% y$	$10 \% 3 = 1$
**	指數次方 (x 的 y 次方 = x^y)	$x ** y$	$10 ** 3 = 1000$
//	整數除法 (強迫使用 <code>int()</code> 轉型)	$x // y$	$10 // 3 = 3$

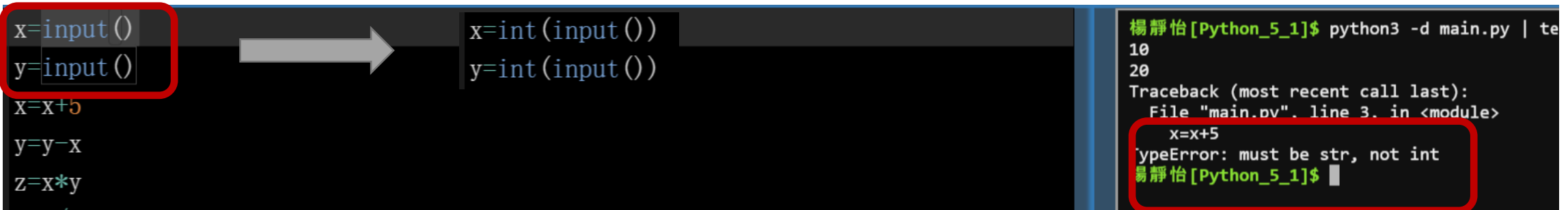
Python 的 7 則運算

符號
+
-
*
/
%
**
//

這 7 種運算符號稱為

算數運算子
(Arithmetic Operators)


```
x=input()  
y=input()  
x=x+5  
y=y-x  
z=x*y
```



```
楊靜怡 [Python_5_1]$ python3 -d main.py | te  
10  
20  
Traceback (most recent call last):  
  File "main.py", line 3, in <module>  
    x=x+5  
TypeError: must be str, not int  
楊靜怡 [Python_5_1]$
```

- Python將所有的輸入都視為字串，也就是將輸入的10與20都是文字，必須轉型成數字才能運算
- 例如：
x=int(input())，將輸入的資料轉為整數
y=float(input())，將輸入的資料轉為浮點數

PYTHON

延伸的概念

概念1: 運算的優先順序，與數學運算是一樣的

- 跨號優先
- 乘、除(整數除法/符點除法)與取餘數優先
- 同一運算式含有數個加與減，由左至右逐一運算

概念2: 整數除法 vs 浮點除法

- 整數除法: //

- 以整除結果為準
- 得到的值必為**整數**
- 即結果會是強迫使用 `int()` 轉型後的值
- $10 // 3 = 3$

- 浮點除法: /

- 結果以**浮點數**表示
- 即結果會是強迫使用 `float()` 轉型後的值
- $10 / 2 = 5.0$

概念2: 整數除法 vs 浮點除法

```
print(10 / 2)
```

```
print(10 / 3)
```

浮點除法

```
print(10 // 2)
```

```
print(10 // 3)
```

整數除法

執行結果

```
5.0
```

```
3.3333333333333335
```

浮點除法

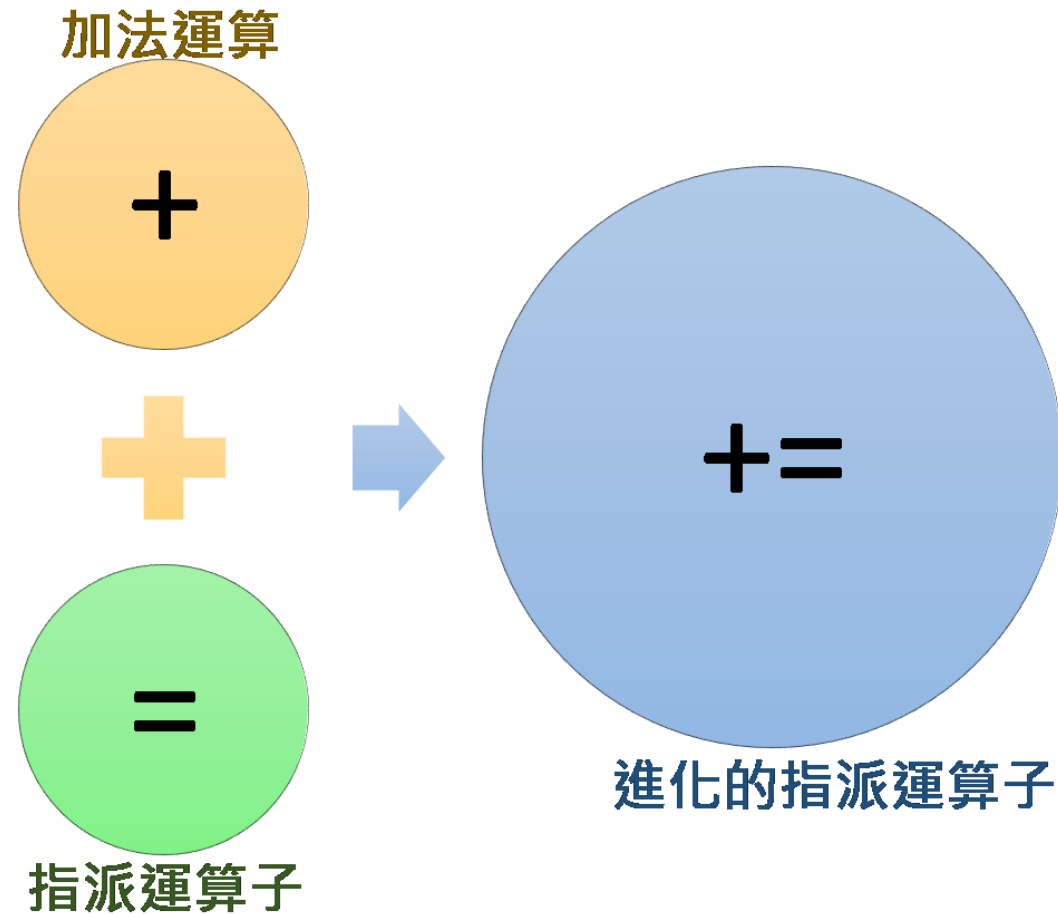
```
5
```

```
3
```

整數除法

概念 3: 更多的指派運算子

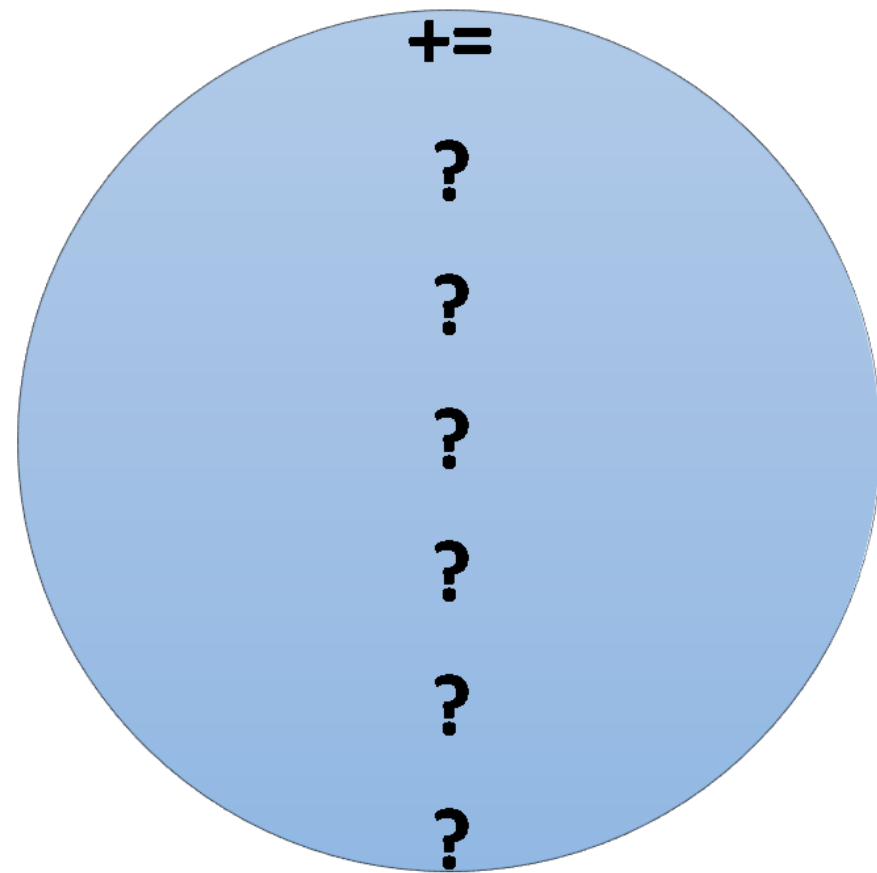
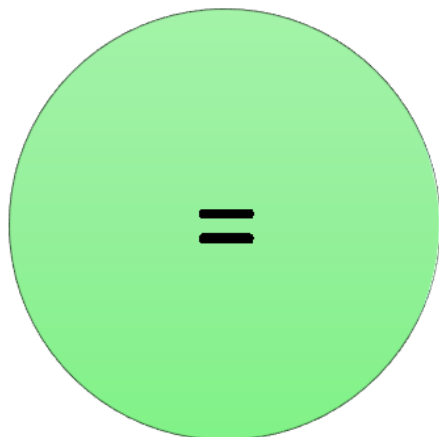
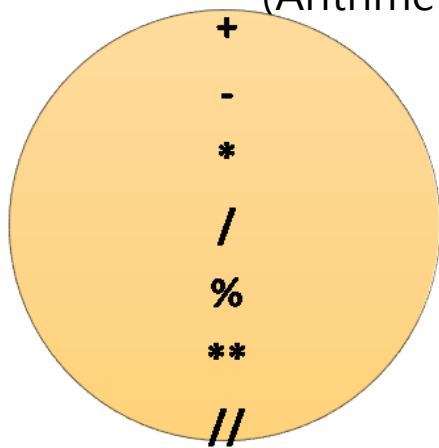
- 還記得**加法運算(+)**遇上**指派運算子(=)**變成什麼嗎？



概念 3: 更多的指派運算子

當其它的
算數運算子
(Arithmetic Operators)
碰上
指派運算子
(Assignment Operators)
又會變成什麼呢？

算數運算子
(Arithmetic Operators)



指派運算子
(Assignment Operators)

概念 3: 更多的指派運算子

算數運算子	用法
+	$x = x + y$
-	$x = x - y$
*	$x = x * y$
/	$x = x / y$
%	$x = x \% y$
**	$x = x ** y$
//	$x = x // y$

進化

指派運算子	用法
+=	$x += y$
-=	$x -= y$
*=	$x *= y$
/=	$x /= y$
%=	$x \% = y$
**=	$x ** = y$
//=	$x // = y$