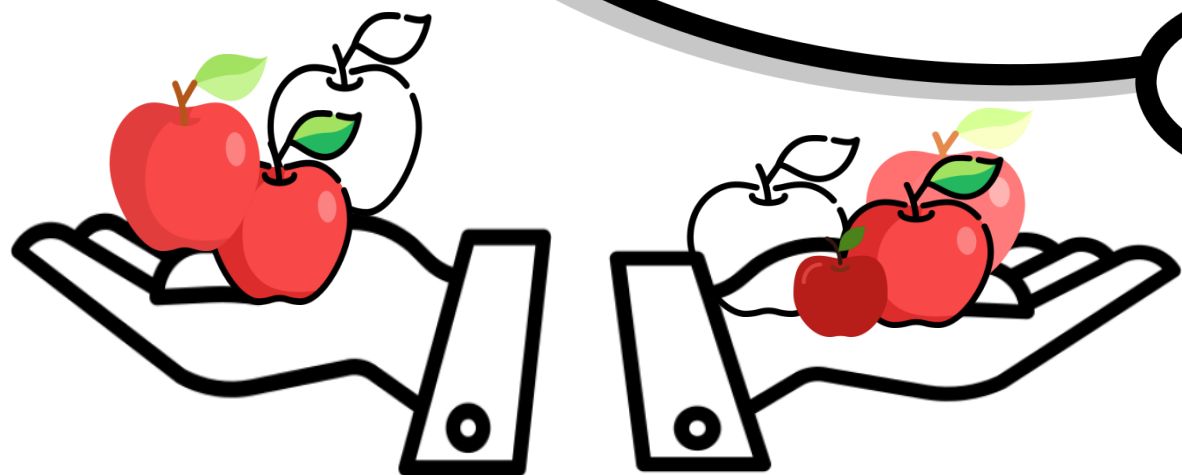


Python

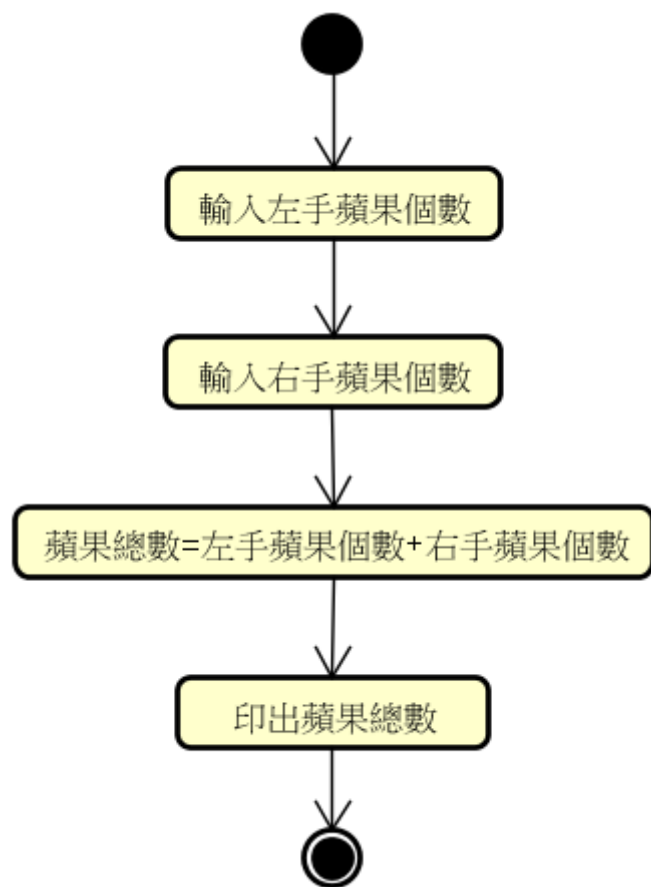
加法

寫一個程式，
從標準輸入取得左手與右手各拿著多少蘋果
請算出一共多少個蘋果？ 運算過程需輸出

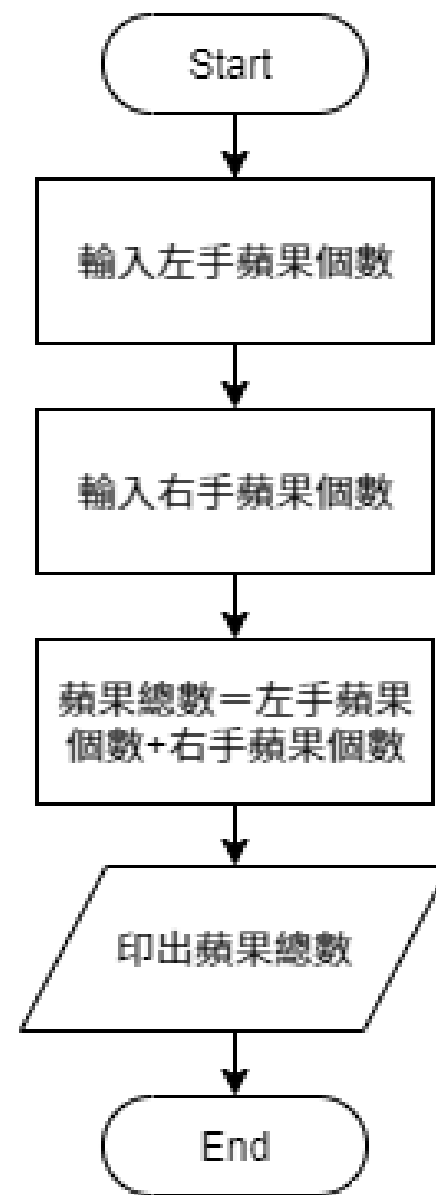
問題



流程



UML流程圖



傳統流程圖

Python程式碼：相加程式碼

```
left = input()
right = input()
apple = left + right
print("{0} + {1} = {2}".format(left, right,
apple))
```

apple會得
left+right的結果

執行結果

我們預期的結果

3 + 4 = 7

實際執行的結果

3+4=34

input() 不負責分辨資料喔！

```
line = input()
print(line)
print(type(line))
line = input()
print(line)
print(type(line))
```

不論輸入什麼，input() 收到的資料就是一堆文字(字串)，這堆文字裡面是否有整數、小數還是特殊符號資料的，input() 是不知道的喔！

執行結果

```
hello
hello
<class 'str'>
10
10
<class 'str'>
```

使用int轉型成整數：num = int(input())

```
num = input()
print(num)
print(type(num))
num = int(input())
print(num)
print(type(num))
```

使用前面教過的 int() 轉型，就可以讓輸入的資料轉成整數喔！

執行結果

```
10
10
<class 'str'>
10
10
<class 'int'>
```

Python程式碼：兩數相加程式碼

```
left = int(input())  
right = int(input())  
apple = left + right  
print("[0] + [1] = [2]".format(left, right,  
apple))
```

apple會得
left+right的結果

執行結果

3 + 4 = 7

使用float轉型成浮點數：



```
num = float(input())
```


PYTHON

延伸的概念

概念1: = , 不是等號喔!

```
a=int(input())  
a = a + 10  
print("{0}".format(a))
```

a 不是等於 a+10 ,
而是 a 會得到 a+10的
結果

這個等號是將右邊的內
容指派給左邊的變數

概念1: = , 不是等號喔!

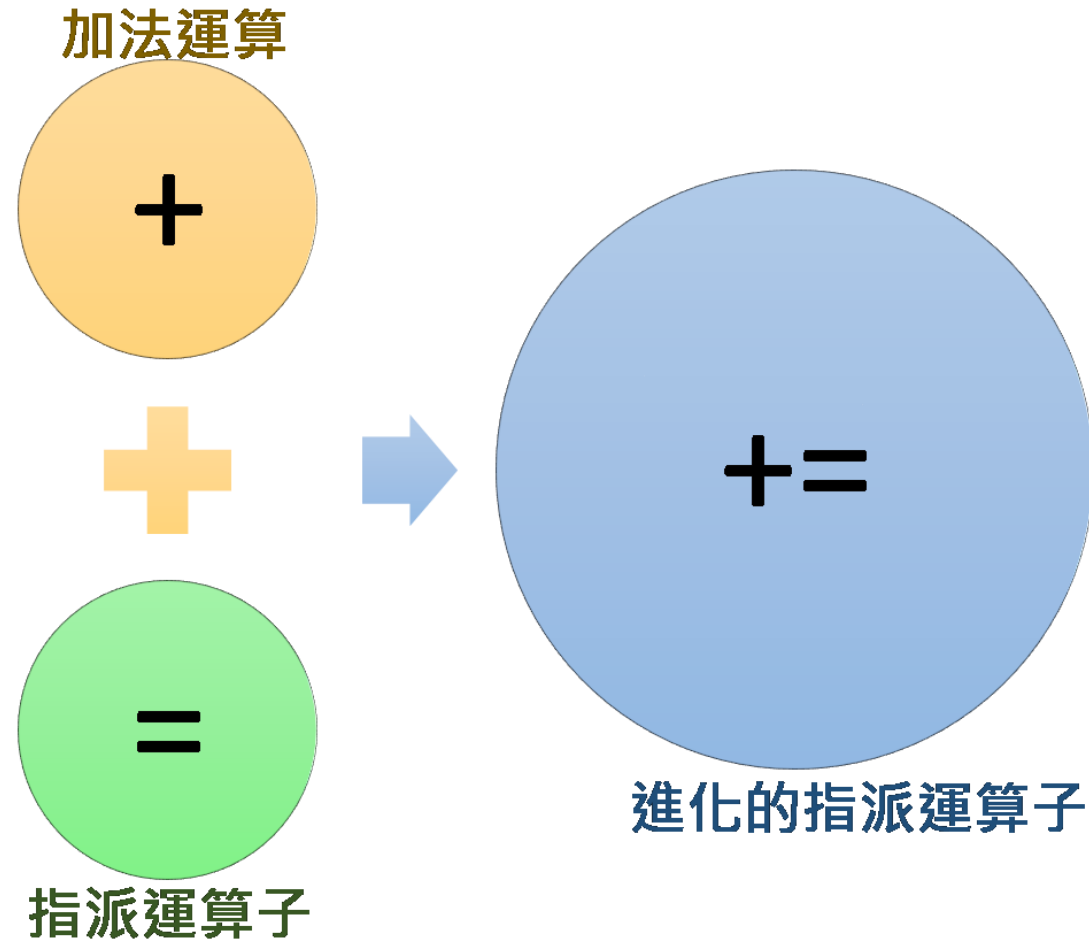
右邊指派給左邊

```
a = 3
a = a + 10
b = 4
b = b + 11
c = a + b
```

這些等號，我們稱為

指派運算子
(Assignment Operators)

概念2: 加法運算(+)遇上指派運算子(=)



概念2: 加法運算(+)遇上指派運算子(=)

```
a = 3
```

```
a = a + 10
```

```
b = 3
```

```
b += 10
```

```
print("{0} <=> {1}".format(a, b))
```

其實 `a+=10` 就是 `a=a+10`
`+=` 不過就是懶人進階版的寫法喔！
(可以少寫一個變數)

執行結果

```
13 <=> 13
```

概念2: 加法運算(+)遇上指派運算子(=)

- $+=$ 就是做完加法後再進行指派
- 所以， $+=$ 本質仍然是一種指派運算子喔！

