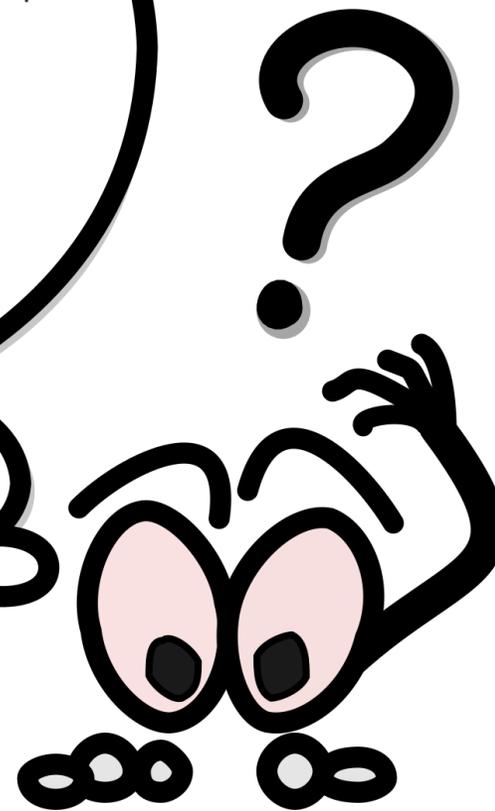
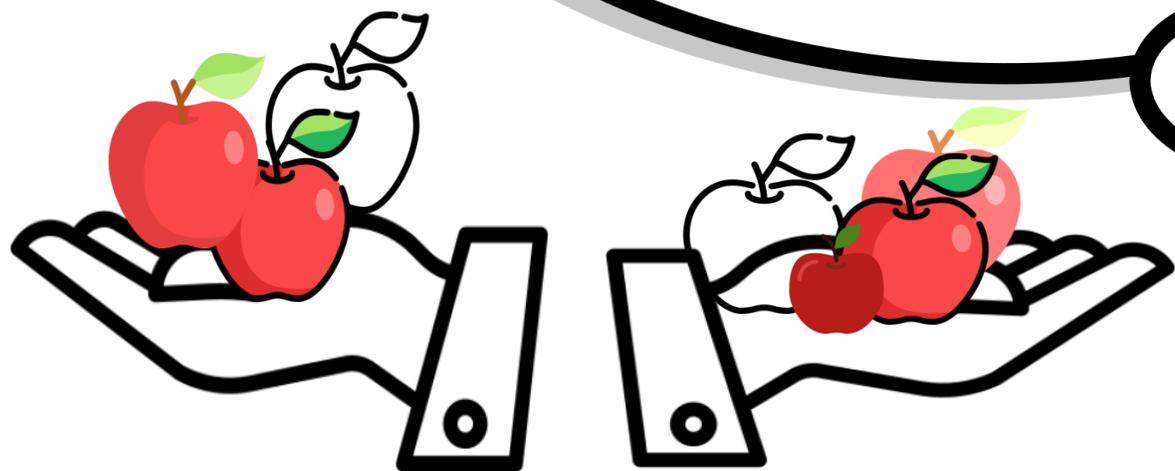


# Java

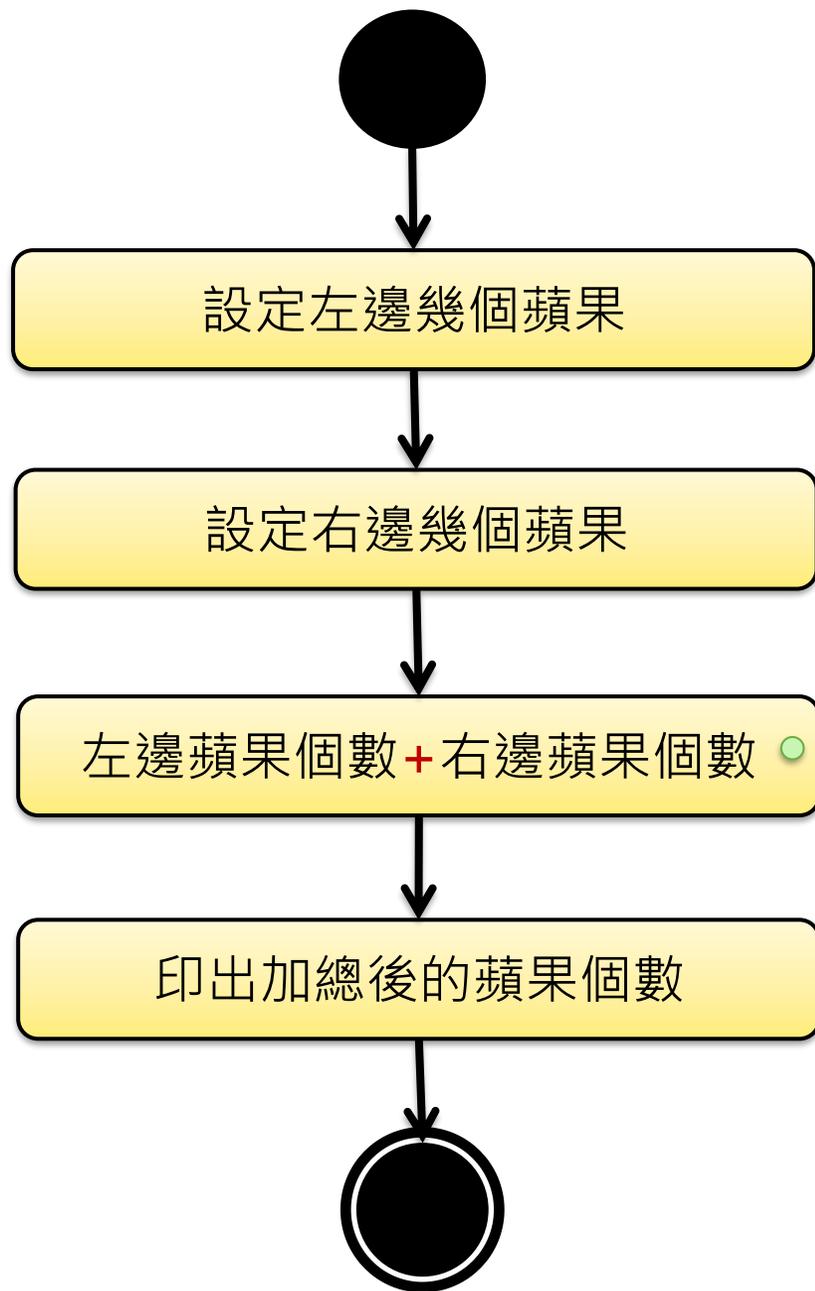
數的加法

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



Java 的加法運算同樣也是用數學運算中的 + 運算子

# Java 程式玩加法

```
import java.util.Scanner;

class web {
    public static void main(String[] args) {
        int left, right, apples;
        Scanner sc = new Scanner(System.in);
        left = sc.nextInt();
        right = sc.nextInt();
        apples = left + right;
        System.out.printf("%d + %d = %d\n", left, right, apples);
    }
}
```

apple會得  
left+right的結果



執行結果

輸入左邊  
幾個蘋果

3

4

3

+

4

=

7

輸入右邊  
幾個蘋果

# JAVA

## 延伸的概念

# 概念1：=，不是等號喔！

```
import java.util.Scanner;

class web {
    public static void main(String[] args) {
        int apples = 10;
        apples = apples + 10;
        System.out.println(apples);
    }
}
```

apples不是等於 apples+10，  
而是 apples 會得到  
apples+10 運算後的值！

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 )

18dice

# 概念2：加法運算(+)遇上指派運算子(=)

```
import java.util.Scanner;  
  
class web{  
    public static void main(String[] args) {  
        int a = 3;  
        int b = 3;  
        a = a + 10;  
        b += 10;  
        System.out.printf("%d <=> %d\n",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)

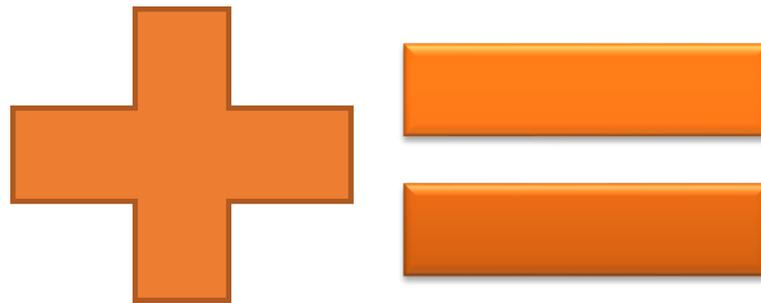


執行結果

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！



# 概念3：輸出參照要清楚

$$3+4=7$$

`%d+%d=%d`

```
System.out.printf("%d + %d = %d\n", left, right, apples);
```

第幾個輸出格式就會對應逗點  
後面的第幾個變數