

## Design a Program 程式設計

## Syntax Error

所謂syntax error說的是在給電腦指令的表達上不符合程式在語法上的要求。

例一: `print 1`則給予error message。

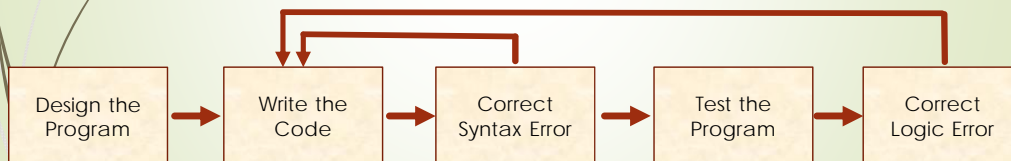
```
>>> print(1)
1
>>> print 1
SyntaxError: Missing parentheses in call to 'print'
>>> |
```

例二: `'3'*3`會產生無法evaluate的錯誤

```
>>> 3*3
'333'
>>> '3'*3
'333'
>>> '3'*'3'
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    '3'*'3'
TypeError: can't multiply sequence by non-int of type 'str'
>>> |
```

## The Program Development Cycle

1. **Design** the Program (程式設計)
2. **Write/Change** the **Code** (程式撰寫)
3. **Correct Syntax** Error (更正語法錯誤)
4. **Test** the Program (程式測試)
5. **Correct Logic/Semantic** Error (更正邏輯/語意錯誤)



## Logic Error

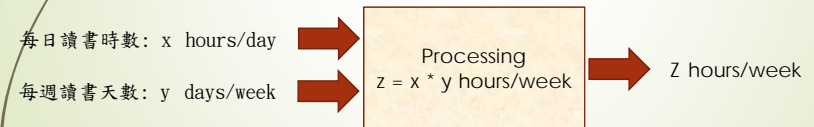
- 所謂logic error是說雖然你的syntax都是正確的，但是邏輯上是不正確的。例如說請你計算6是否是偶數，結果計算結果說6不是偶數，就表示你的程式執行的結果是錯誤的。

## 學習步驟

	語言(English)	程式語言(Python)
Lexical 語彙	單字、片語 <b>Ex.</b> Cat、clothes、phone	語言中的token <b>Ex.</b> int、bool、str、if、.(Reserve words) 1、2、3.14..... (numbers)
Syntax 語法	S+V+O 主詞+動詞+受詞 <b>Ex.</b> I like you. (o) I favorite you. (x)	If <condition>: True Blocks <b>Ex.</b> if a>0:           if a>0 a = a - 1       a = a - 1 (o)               (x)
Semantic 語意	邏輯上的正確性 <b>Ex.</b> Cat eats mouse. (o) Dog eats mouse. (x)	邏輯上的正確性 <b>Ex.</b> 程式判斷6是否為 偶數，答案說不是
Pragmatics 意境	作文程度 <b>Ex.</b> 修辭、押韻	程式的好壞程度，可讀性，執行效率等 <b>Ex.</b> O(n)、O(log n) 之 program

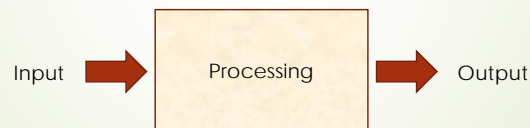
## Input, Processing, Output 實例

假設你每天有固定的讀書時數，也知道一星期讀幾天的書，希望電腦幫你計算每週的讀書時數，則我們程式的輸入(input)是『每日讀書時數』和『每週讀書天數』。處理(processing)的過程將兩數相乘，得到的『每週讀書時數』就是的輸出值(output)了。



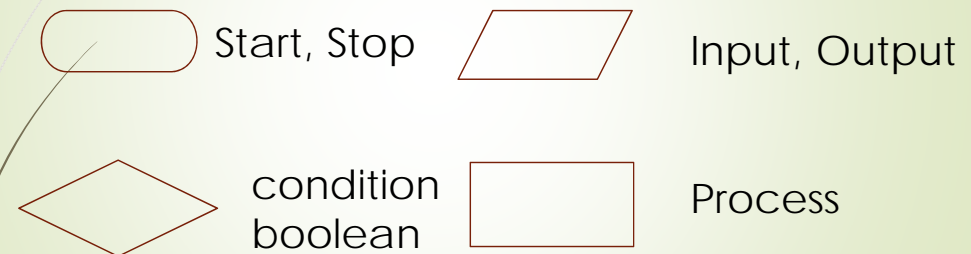
## Input, Processing, Output

電腦主要的功能是利用它快速計算與大量記憶的能力來幫你處理資料。所以輸入資料(input)、處理(processing)、輸出(output)這三個步驟就是程式的主要架構。



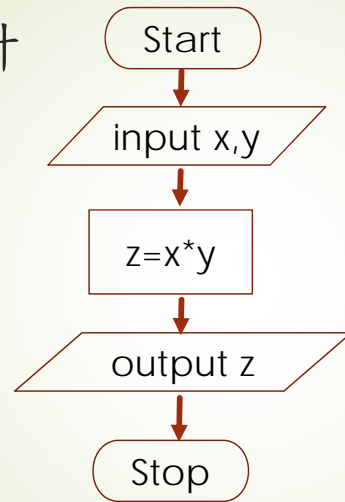
## 程式設計

流程圖幫助您在程式設計時先了解程式的執行順序。每個不同形狀的圖代表不同的意義。





## 程式設計



## Read and Print

- 目前我們使用print將所要的資料輸出到螢幕
- 我們使用input來將資料讀入

```
m = input()
```

```
m = input('Please input m: ')
```

- 讀入的值會是一個字串，並將他assign給變數m

```
>>> m = input('Please input m: ')
Please input m: 23
>>> print(m)
23
>>> |
```