CONTROL FLOW

Chih-Chung Hsu (許志仲)
Institute of Data Science
National Cheng Kung University
https://cchsu.info

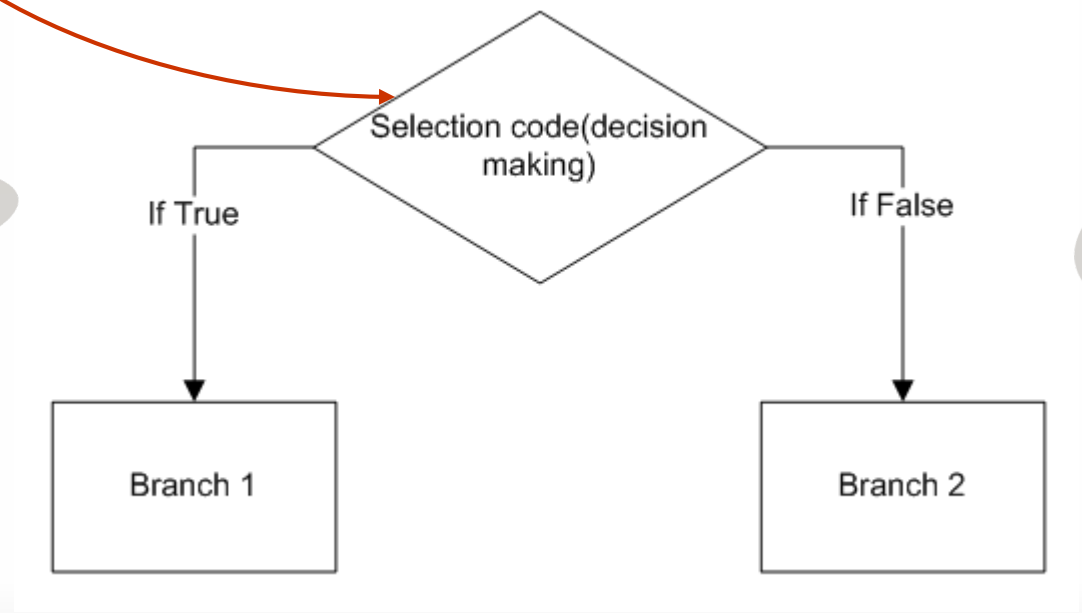
判斷式
PYTHON 程式碼縮排
判斷式

How to determine whether code should we need to run?

Why CONTROL FLOW?

- 寫程式總是會遇到十字路口

How to decide?



通關密語

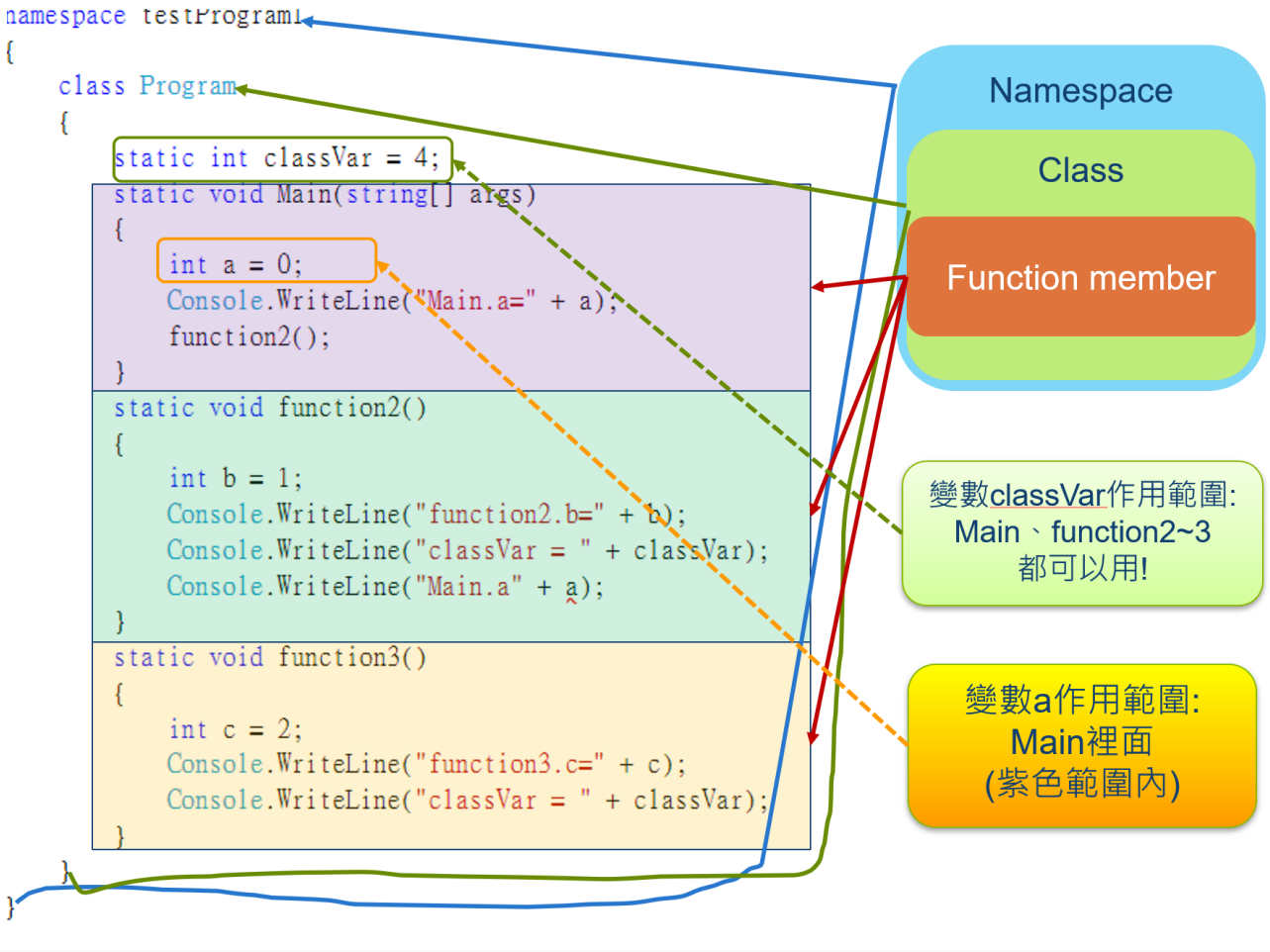
只算必要的部分

不同條件有不同程式

避免數值問題

How to know “Activate Space”

- 以前是這樣，看起來還算整齊



How to know “Activate Space”

- 但也會寫成這樣

```
1  //Rextester.Program.Main is the entry point for your code. Don't change it.
2  //Microsoft (R) Visual C# Compiler version 2.9.0.63208 (958f2354)
3
4  using System;
5  using System.Collections.Generic;
6  using System.Linq;
7  using System.Text.RegularExpressions;
8
9  namespace Rextester
10 {
11 public class Program
12 {
13 public static void Main(string[] args)
14 {
15     //Your code goes here
16 Console.WriteLine("Hello, world!");
17 }
18
19 public static goodmorning()
20 {
21 Console.WriteLine("Good morning, you fool!")
22 }
23 }
24 }
```

Python 程式碼縮排

- 不好意思，我就是需要整齊點
  - 要告訴我範圍，請縮排!!

- Python 程式碼縮排格式
  - Python 語言以冒號「:」及縮排來表示程式區塊，縮排為 1 個 Tab 鍵或 4 個空白鍵 (建議，但非強制)，且不得 “混用”
  - 例如：

Control Flow

- 程式流程控制
  - Python 流程控制命令分為兩大類：
    - **判斷式**：根據關係運算或邏輯運算的條件式來判斷程式執行的流程，若條件式結果為 True，就執行跳躍。判斷式命令只有一個：

      ```
      if…elif…else
      ```

    - **迴圈 (Next time!!)**：根據關係運算或邏輯運算條件式的結果為 True 或 False 來判斷，以決定是否重複執行指定的程式。迴圈指令包括下列兩種：（迴圈將在第 4章詳細說明）

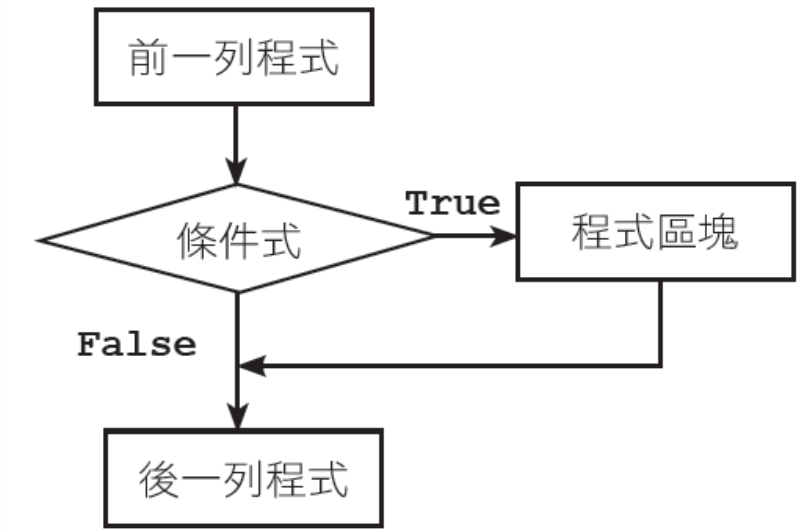
      ```
      for
      while
      ```

Control Flow-1

- 單向判斷式（if…）
  - 「if…」為單向判斷式，是 if 指令中最簡單的型態，語法為：

```
if （條件式）:
    程式區塊
```

  - 以下是單向判斷式流程控制的流程圖：

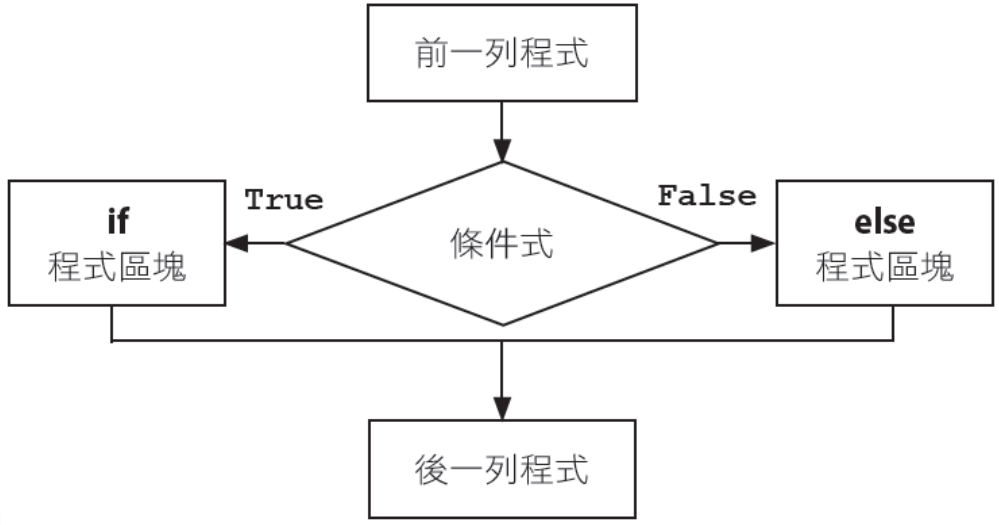


Control Flow-2

- 雙向判斷式（if…else）
  - 「if…else…」為雙向判斷式，語法為：

```
if (條件式):
    程式區塊一
else:
    程式區塊二
```

  - 以下是雙向判斷式流程控制的流程圖：



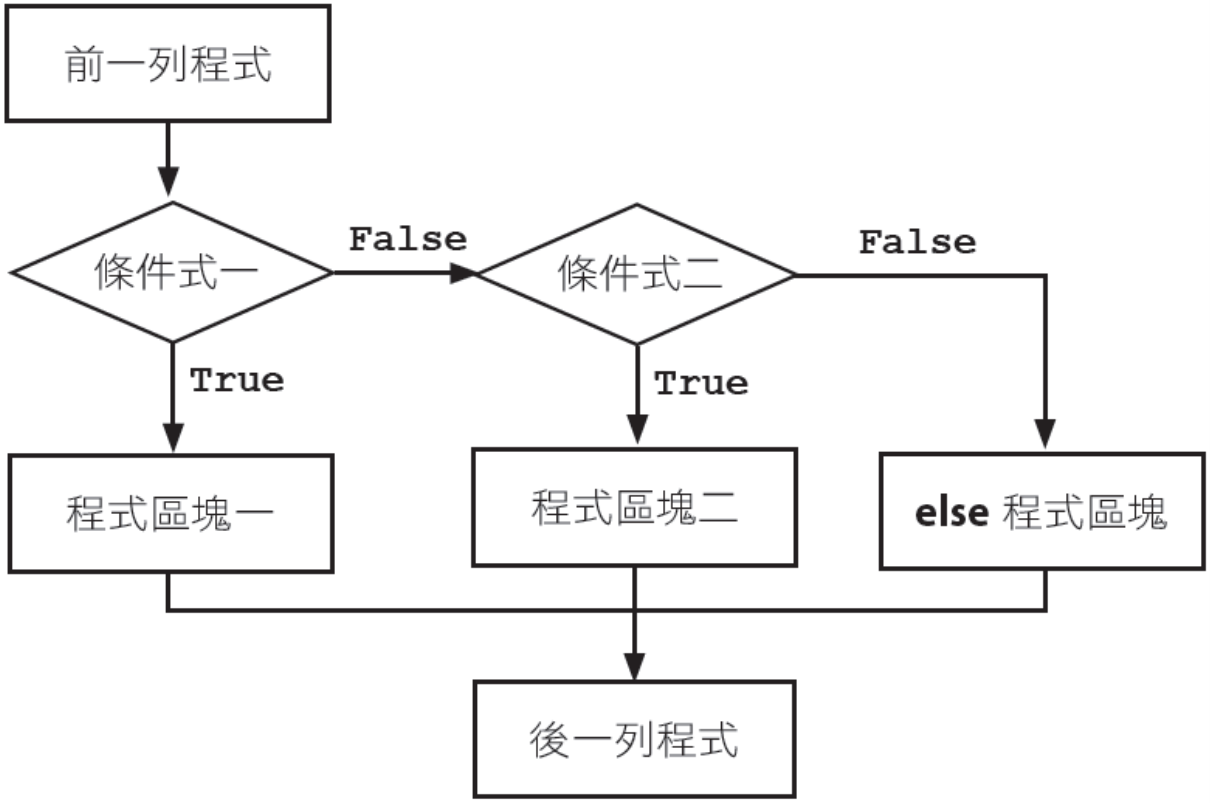
Control Flow-3 (多重條件)

- 多向判斷式（if…elif…else）
  - 「if…elif…else」的語法為：

```
if (條件式一) :
    程式區塊一
elif (條件式二):
    程式區塊二
elif (條件式三):
    .........
[else:]
    程式區塊else
```

多向判斷式流程控制的流程圖

(以設定兩個條件式為例):



Simplified Code for Control Flow

- x = 0
- If (condition-1):
  - Code-1 for x
- Else:
  - Code-2 for x

- 如果Code只有一行，或是簡單的計算，可以改寫成為單行描述式
  - x = (Code-1) if (condition-1) else Code-2
- 範例計算公式
  - y = 1/(log(x))
  - y = 1 / log(x) if x>0 else None

More than this

- a = 330
- b = 330

```
print("A") if a > b else print("=") if a ==
b else print("B")
```

- 判斷變數什麼型態怎麼辦?
  - Isinstance(比較對象，目標類別) [returns True/False]

```
x = isinstance(5, int)
```

- 判斷是不是數字怎麼辦？
  - String.isdigit() [returns True/False]

```
txt = "50800"

x = txt.isdigit()
```

Summary

- 在本章中介紹了下面的重點：
  - Python 語言以冒號「:」及縮排來表示程式區塊，縮排可為 1 個 Tab 鍵或4 個空白鍵。
  - 有條件跳躍：根據比較運算或邏輯運算的條件式來判斷程式執行的流程，若條件式結果為True，就執行跳躍。
  - 迴圈：根據比較運算或邏輯運算條件式的結果為 True 或 False 來判斷，以決定是否重複執行指定的程式。
  - 「if…」為 單向判斷式，當條件式為 True 時，就會執行程式區塊的敘述；當條件式為 False 時，則不會執行程式區塊的敘述。

```
if (條件式):
    程式區塊
```

Summary

- 「if…else…」為 雙向判斷式，當條件式為 True 時，會執行 if 後的程式區塊一；當條件式為 False 時，會執行 else 後的程式區塊二。

```
if (條件式):
    程式區塊一
else:
    程式區塊二
```

- 「if…elif…else」為 多向判斷式，在多項條件式中如果為 True 時，就執行相對應的程式區塊，如果都是 False，則執行 else 後的程式區塊。

```
if (條件式一) :
    程式區塊一
elif (條件式二):
    程式區塊二
    .....
[else:]
    程式區塊else
```

- 在判斷式之內可以包含判斷式，稱為巢狀判斷式。