

tutorJr | 青少兒程式設計

# Python 課程

## 初級課程11

【函式：回傳值】



# 初級Python知識點



# 第十一節課|函式：回傳值

## 第十節課|函式|回傳值

上一堂課我們學習了函式的宣告、設計跟使用，輸入資料後產出結果。本堂課我們在學習函式的另一個重點：回傳值。

我們可以設計執行完函式獲得結果的同時，也回傳一個資料給呼叫函式的程式，舉例來說，我們設計了一個販賣機，我們輸入了投入的金額跟要購買的飲料後，結果就是獲得飲料，同時，回傳剩下的錢。



要注意，函式回傳的值也需要處理，可以用變數存放，或是直接印出來。

函式要如何回傳值呢？利用return這個關鍵字

```
def 函式名(參數):  
    程式碼  
    return 回傳值
```

要注意的是當函式執行到return後，後續的程式就不會再執行囉。

同時，return只能回傳單一的值，例如整數、浮點數或是字串等等。

### 英文小教室

飲料的英文是drinks

自動販賣機是 Vending machine



# 第十一節課|函式：回傳值

## 第十節課|函式|設計一個飲料販賣機

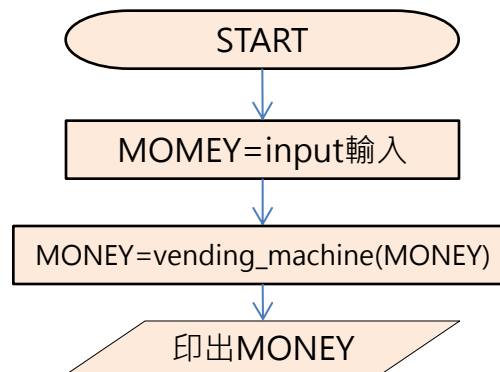
接著就讓我們來設計一個飲料販賣機練習函式跟回傳值吧，先思考這個販賣機有哪些功能：

1. 販賣4種飲料
2. 投入金額後可以判斷可以購買那些飲料同時顯示(印出)給消費者
3. 依據選擇印出購買的飲料
4. 回傳剩下的金額

定義出這四個功能後，這個販賣機的輸入會是錢，輸出會是購買的飲料，同時，回傳剩下的金額。

接著我們試著畫出流程圖出來吧

這是主流程的流程圖，輸入有的金額，同時呼叫販賣機，帶入這個金額作為參數，最後印出還有剩下多少錢。



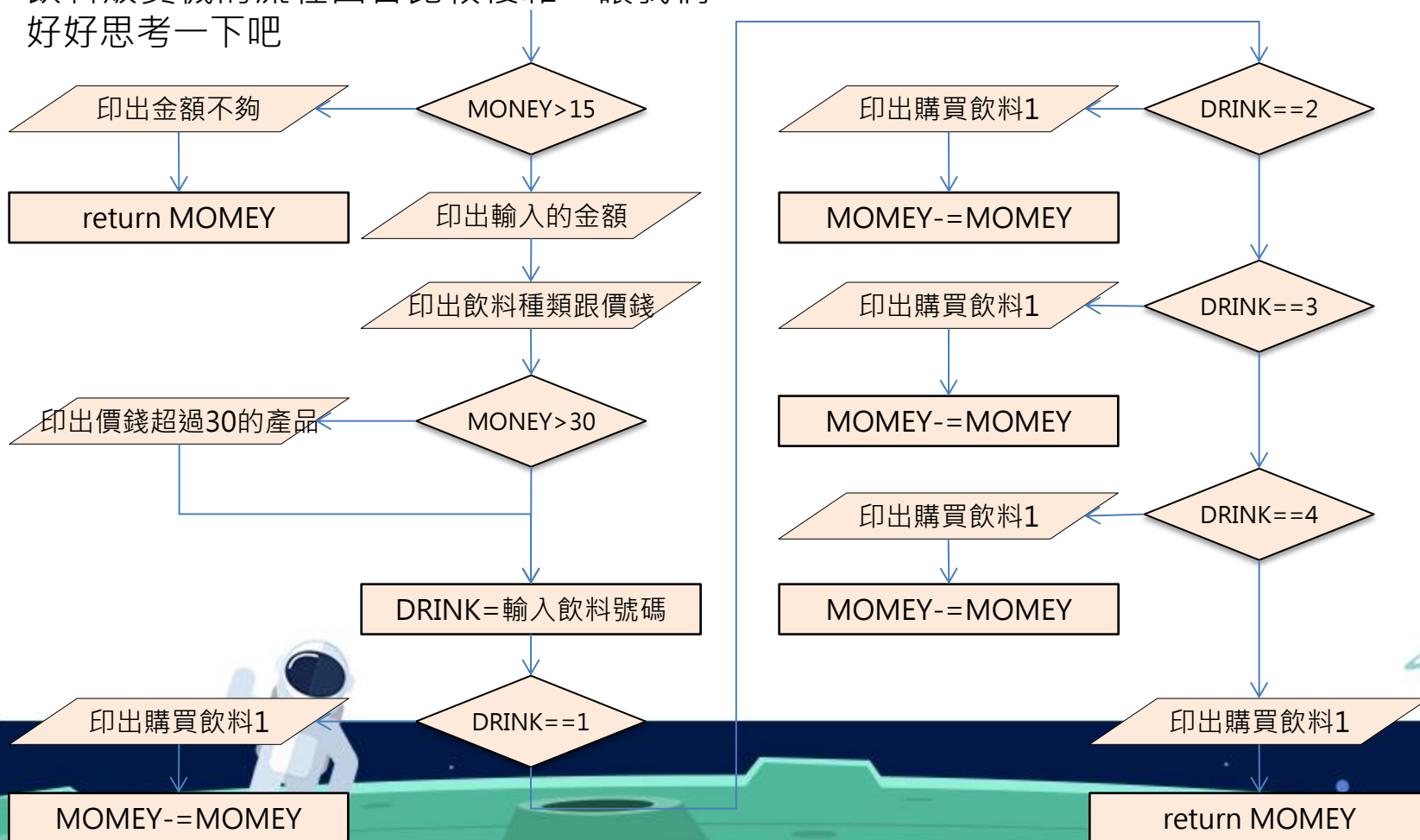
### 程式小提醒

記得輸入的值的型態是字串，如果要轉換為數字要用int()轉換唷

# 第十一節課|函式：回傳值

## 第十節課|函式|設計一個飲料販賣機

飲料販賣機的流程圖會比較複雜，讓我們好好思考一下吧



# 第十一節課|函式：回傳值

## 第十節課|函式|設計一個飲料販賣機

現在讓我們來看看這個程式範例

```
def vending_machine(input_money):  
    if(input_money>15):  
        print("You add $",input_money, " and you can buy below drinks")  
        print("1 Apple juice $20")  
        print("2 Coke          $25")  
        print("3 Lemon juice $15")  
        if(input_money>30):  
            print("4 coffie $45")  
        drink_no=int(input("Please enter the drink number you want to buy"))  
        if(drink_no==1):  
            print("You have bought the Apple juice")  
            input_money-=20  
        elif(drink_no==2):  
            print("You have bought the Coke")  
            input_money-=25  
        elif(drink_no==3):  
            print("You have bought the Lemon juice")  
            input_money-=15  
        elif(drink_no==4):  
            print("You have bought the coffie")  
            input_money-=45  
        else:  
            print("The drink number is not correct, thank you")  
        return input_money  
    else:  
        print("You dont have enough money to buy the dirnks")  
        return input_money
```



# 第十一節課|函式：回傳值

## 第十節課|函式|設計一個飲料販賣機

主程式的部分比較簡單，看看以下的程式範例：

```
my_money=int(input("Please enter the money you want to buy the dirinks"))  
my_money=vending_machine(my_money)  
print(my_money)
```

執行看看，測試的時候可以輸入幾種金額  
確認各種條件都是正確的

Please enter the money you want to buy the dirinks

```
Please enter the money you want to buy the dirinks60  
You add $ 60 and you can buy below drinks  
1 Apple juice $20  
2 Coke $25  
3 Lemon juice $15  
4 coffie $45
```

Please enter the drink number you want to buy

```
Please enter the drink number you want to buy2  
You have bought the Coke  
35
```

# 作品討論與實作

## 討論主題

利用函式創作出自己的作品吧  
你可以加上更多產品到販賣機嗎？

繼續作品實作！





tutorJr | 青少兒程式設計

# Python 課程

## 初級課程12

【模組：利用烏龜模組畫圖】



# 初級Python知識點





# 第十二節課|模組：烏龜畫畫

## 第十二節課|模組|如何使用模組

Python程式的強大在於可以很容易地使用其他人寫好的功能(模組)，例如數據分析有numpy，文件處理的is或是找資料的requests。使用模組的方式如下

1. 確認有安裝模組
2. 在程式裡宣告使用模組

如何安裝模組？在官方網站下載的python環境，使用命令提示字元視窗下，用以下的命令來下載安裝模組

### pip install 模組

看看以下的程式範例

```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\jerylin>pip install juoyter_
```

模組安裝好後，就可以在程式裡使用，要如何宣告模組呢？

### import 模組

看看以下的程式範例

```
import turtle # Allows us to use turtles
import math
```

這裡利用import宣告了兩個模組，一個是畫圖用的turtle，另一個是數學的math。

#### 英文小教室

安裝的英文是install  
進口、引入的英文是import





## 第十二節課|模組：烏龜畫畫

### 第十二節課|模組|利用烏龜來練習

我們來利用一個常見的繪圖模組- turtle 烏龜來練習使用模組。首先我們先來創建一個畫布，在turtle模組提供了幾個方法來建立畫布，畫布建立後也有各種方法來運用：

`turtle.screen()`：建立一個畫布

`screen.setup(長,寬)`：設置畫布的長跟寬

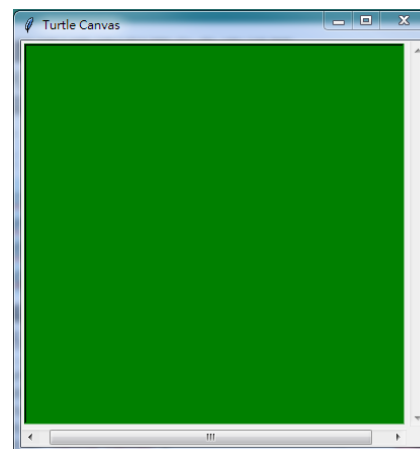
`screen.title(字串)`：設置畫布的名稱

`screen.bgcolor(字串)`：設置畫布的背景色

練習看看，建立一個400x400，顏色為綠色的畫布，看看以下的程式範例

```
import turtle # Allows us to use turtles
import math
screen = turtle.Screen()
screen.setup(400, 400)
screen.title('Turtle Canvas')
screen.bgcolor('green')
screen.exitonclick()
```

執行程式看看，會彈出一個視窗，成功



#### 程式小提醒

建一個變數來裝模組裡的建立的畫布，再利用這個畫部的功能做事情





## 第十二節課|模組：烏龜畫畫

### 第十二節課|模組|利用烏龜來練習

畫布建立好了，讓我們來利用turtle裡的畫筆功能來畫圖吧。

1. 建立畫筆：建立以及設置畫筆的特性
2. 利用畫筆功能畫畫

如何利用turtle建立一支畫筆？

`turtle.Turtle()`

建立一個變數來裝這枝筆就可以了。接著來想一下，一支畫筆會有那些特徵呢？

1. 畫筆的顏色
2. 畫筆的粗細
3. 畫圖的速度

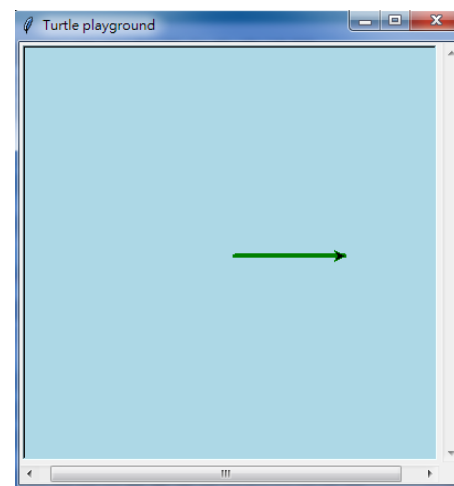
這些都可以透過烏龜畫筆提供的方法來設置



來看看以下的程式範例

```
my_pen=turtle.Turtle()  
my_pen.pencolor("green") #設置畫筆顏色  
my_pen.pensize(4) #設置畫筆尺寸  
my_pen.speed(5) #設置畫筆畫的速度  
  
my_pen.forward(100) #測試畫線
```

執行看看，是否得到以下結果



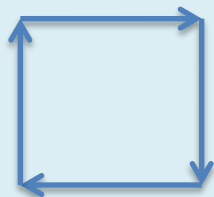


## 第十二節課|模組：烏龜畫畫

### 第十二節課|模組|畫一個正方形

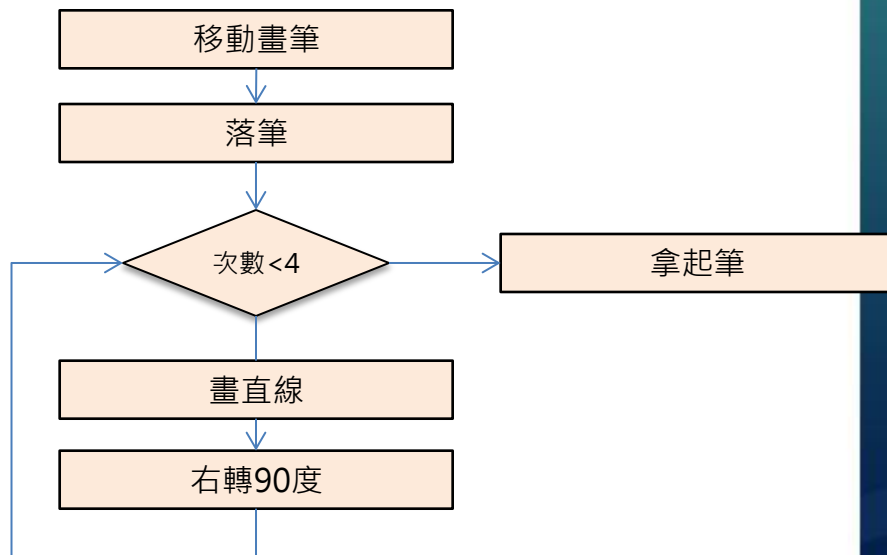
畫布做好了，畫筆也建立好了，讓我們來畫一個正方形練習吧。想想看，你手上有一支畫筆的時候，你會怎麼畫一個正方形？

1. 把筆移動到要畫的位置
2. 落筆
3. 水平移動畫筆
4. 轉動畫筆90度
5. 重複四次
6. 拿起筆



利用turtle畫畫也是一樣的原理唷，這跟我們的習慣一樣，寫起程式也就更輕鬆囉，以上的流程也就是流程圖，試著把流程圖畫出來，同時，看看以下的程式範例

流程圖



試著依照流程圖把程式寫出來







## 第十二節課|模組：烏龜畫畫

### 第十二節課|模組|畫一個正方形

來看看以下的程式範例，你也寫出來了嗎？

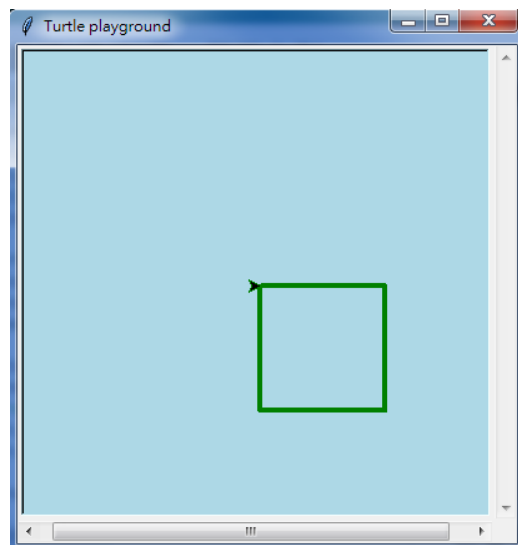
```
import turtle # Allows us to use turtles
screen = turtle.Screen()
screen.setup(400, 400)
screen.title('Turtle playground')
screen.bgcolor('lightblue')

my_pen=turtle.Turtle()
my_pen.pencolor("green") #設置畫筆顏色
my_pen.pensize(4) #設置畫筆尺寸
my_pen.speed(5) #設置畫筆畫的速度

my_pen.goto(0,0) #移動畫筆到座標(0,0)的位置
my_pen.pendown() #放下筆要準備開始畫囉
for i in range(4): #畫四個邊，執行四次
    my_pen.forward(100) #畫直線，長度100
    my_pen.right(90) #往右轉90度

my_pen.penup() #提起筆
screen.exitonclick()
```

執行看看，結果是否正確



自己試著改變畫筆的顏色、尺寸跟速度，畫出各種大小的正方形吧







## 第十二節課|模組：烏龜畫畫

### 第十二節課|模組|寫一個畫矩形的函式

我們在第10堂有學習到函式，接下來我們就練習寫一個可以畫出矩形的函式吧，先設計這個函式需要參數嗎？要幾個參數？我們可以先思考一個矩形有哪些條件？

#### 1. 矩形的長跟寬

接著，想想矩形要怎麼畫？參考正方形的作法，畫出流程圖，記得要注意矩形的長跟寬可能不同，所以for-loop的次數要幾次呢？



看看以下的程式範例

```
def draw_rect(length, width):  
    my_pen.pendown() #放下筆要準備開始畫囉  
    for i in range(2): # 畫四個邊，執行四次  
        my_pen.forward(length) #畫長的邊  
        my_pen.right(90) #往右轉90度  
        my_pen.forward(width) #畫寬的邊  
        my_pen.right(90) #往右轉90度  
    my_pen.penup() #提起筆  
  
draw_rect(100,50)
```

執行看看結果是否正確

英文小教室

矩形的英文是rectangle



# 作品討論與實作

## 討論主題

如何利用矩形函式劃出一個機器人呢？

繼續作品實作！



tutorJr | 青少兒程式設計



# Python 課程

## 初級課程13

【模組：利用烏龜模組填顏色、畫更多形狀】



# 初級Python知識點





## 第十三節課|模組：烏龜填顏色、更多形狀

### 第十三節課|模組|利用烏龜填顏色

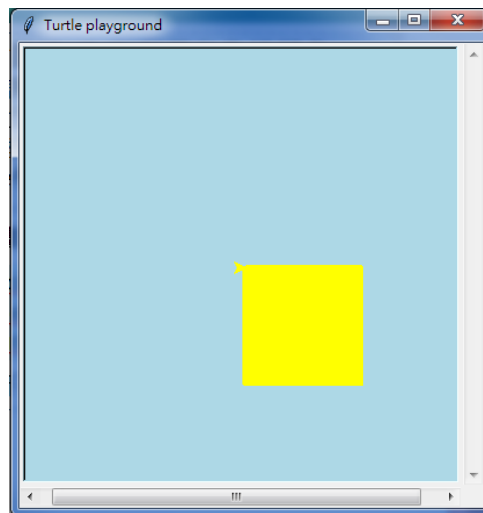
上一堂課我們學習到模組，學習用烏龜來畫圖，也寫了一個函式來畫矩形，那可以用烏龜模組來填顏色嗎？當然可以囉！利用turtle模組塗顏色的邏輯跟我們平時畫圖有點像，先指定要塗的顏色，然後開始塗，接著要畫塗的形狀，然後完成塗色。

```
turtle.color(color) #color用字串表示  
turtle.begin_fill()  
turtle.end_fill()
```

看看以下的程式範例，同時自己試試看

```
my_pen.color("yellow") #設置塗的顏色  
my_pen.begin_fill() #開始塗顏色  
for i in range(4):  
    my_pen.forward(100)  
    my_pen.right(90)  
my_pen.end_fill() #完成塗顏色
```

執行看看，結果是否正確。



#### 程式小提醒

Turtle裡的顏色表示方式是利用字串





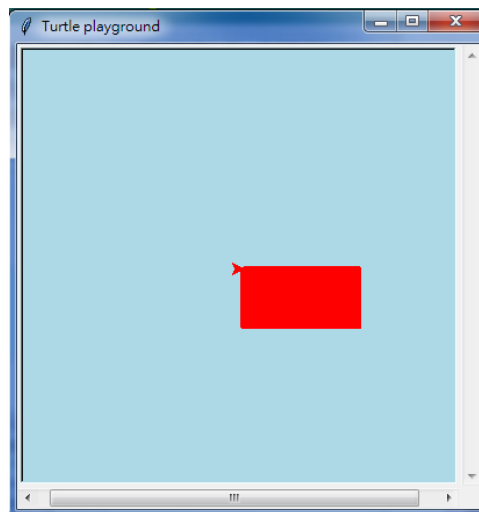
# 第十三節課|模組：烏龜填顏色、更多形狀

## 第十三節課|模組|畫矩形函式調整

接著，我們把塗顏色這個功能加入到我們上堂課寫的矩形函式中，想想看，我們要怎麼調整呢？沒錯，把顏色做為一個參數。參考以下的程式範例

```
def draw_rect(length, width, color):  
    my_pen.pendown() #放下筆要準備開始畫囉  
    my_pen.color(color)  
    my_pen.begin_fill()  
    for i in range(2): # 畫四個邊，執行四次  
        my_pen.forward(length) #畫長的邊  
        my_pen.right(90) #往右轉90度  
        my_pen.forward(width) #畫寬的邊  
        my_pen.right(90) #往右轉90度  
    my_pen.end_fill()  
    my_pen.penup() #提起筆  
  
draw_rect(100,50,"red")
```

執行看看，結果是否正確。



### 程式小提醒

def 函式宣告後要記得加上冒號  
同時要有縮排







## 第十三節課|模組：烏龜填顏色、更多形狀

### 第十三節課|模組|畫六邊形

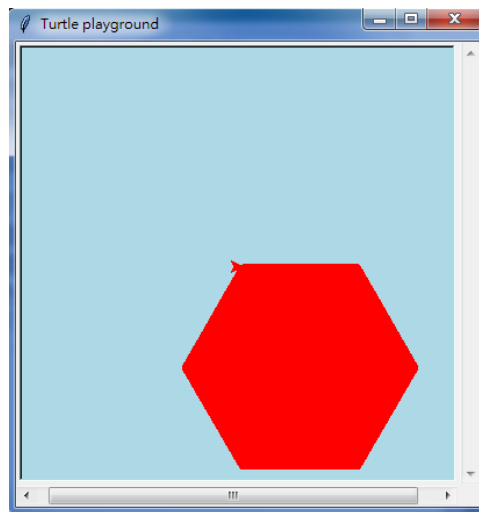
了解turtle模組如何畫出矩形後，想看看要如何畫出正六邊形呢？想想正六邊形的特點

六個邊每個邊長度一樣

聯想一下，畫矩形時有個動作右轉90度，因為四個邊，每個角度都相同，所以計算每個角度是  $360/4$ ，以此類推，六邊形的每個角度應該是多少？沒錯，就是60度  
看看以下的程式範例，自己試試看

```
def draw_hexagon(length,color):  
    my_pen.pendown() #放下筆要準備開始畫囉  
    my_pen.color(color)  
    my_pen.begin_fill()  
    for i in range(6): # 畫四個邊，執行四次  
        my_pen.forward(length) #畫長的邊  
        my_pen.right(60) #往右轉90度  
    my_pen.end_fill()  
    my_pen.penup() #提起筆  
  
draw_hexagon(100,"red")
```

執行看看，結果是否正確。



自己試著畫出三角形、八邊形跟其他不同的形況看看







# 第十三節課|模組：烏龜填顏色、更多形狀

## 第十三節課|模組|畫圓形

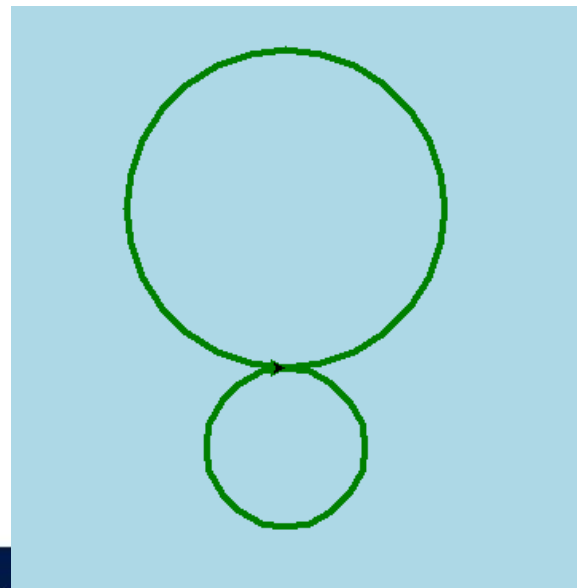
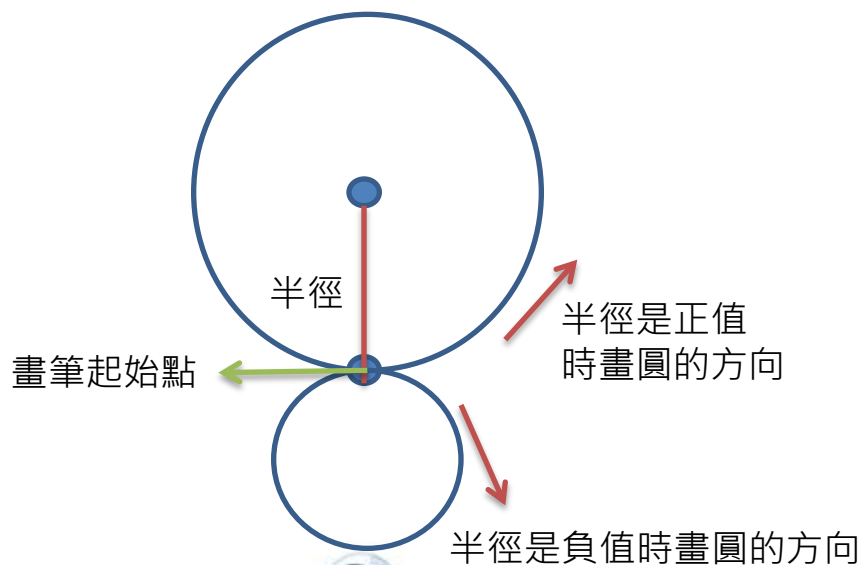
要如何用turtle畫出圓形呢？Turtle提供一個方法可以直接話出圓形，方法如下

```
turtle.circle(半徑)
```

看看以下的程式範例，畫兩個圓，不同的半徑，正值跟負值的差異

```
my_pen.circle(100)  
my_pen.circle(-50)
```

看執行結果，了解不同點了嗎？





## 第十三節課|模組：烏龜填顏色、更多形狀

### 第十三節課|模組|設計一個機器人吧

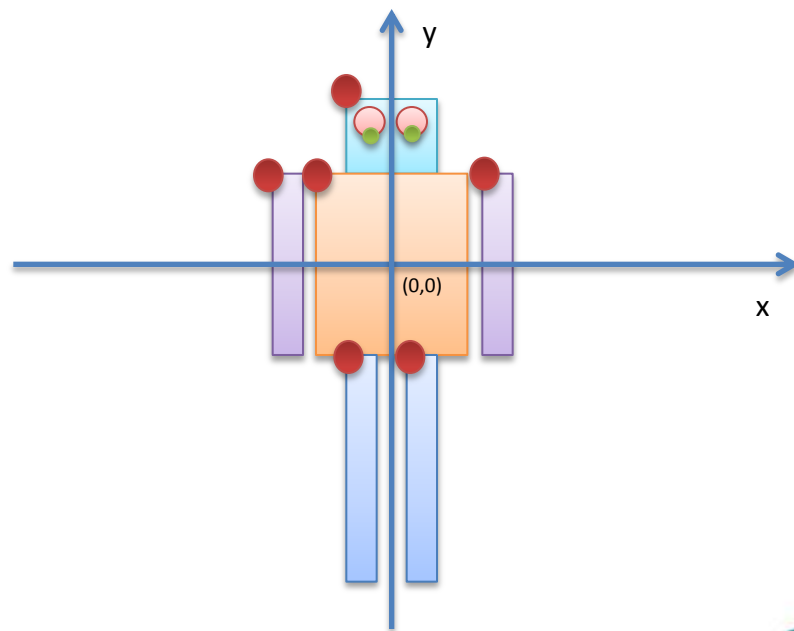
學習如何利用turtle模組化出各種形狀後，讓我們來設計一個機器人，並且用turtle模組化出來吧？想看看你的機器人有哪些部位？

1. 一個頭 (正方形，裡面有兩個圓形眼睛)
2. 一個身體 (長方形)
3. 兩隻手 (長方形)
4. 兩隻腳 (長方形)

看起來像是這樣，包含6個矩形跟2個圓形



接下來我們要思考每個圖形的起始位置，這就跟座標有關囉，請看座標圖



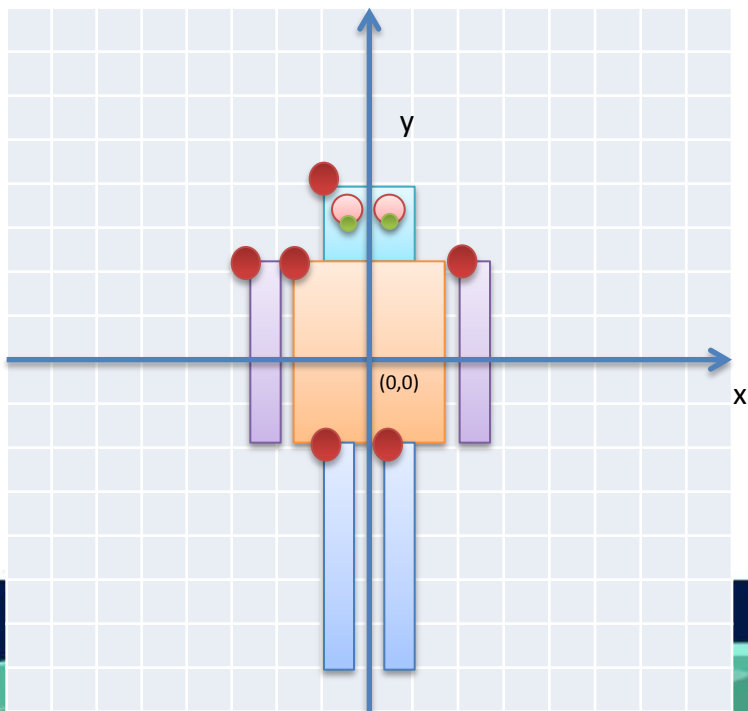
訂出6個紅點(矩形)的座標跟2個綠點(圓形)的座標，各矩形的長跟寬以及圓形的半徑



## 第十三節課|模組：烏龜填顏色、更多形狀

### 第十三節課|模組|turtle座標

在訂出機器人各部位前我們先了解一下turtle的座標，中心點是(0,0)，往右x越來越大，往上y越來越大，請看下圖，每個小方塊都是20長寬的話，可以計算每個點的座標



部位	長	寬	x	y	顏色
頭	40	40	-20	80	goldenrod
身體	80	80	-40	40	thistle
左手	80	20	-60	40	goldenrod
右手	80	20	40	40	goldenrod
左腳	100	20	-30	-40	goldenrod
右腳	100	20	10	-40	goldenrod

部位	半徑	x	y	顏色
左眼	5	-10	60	black
右眼	5	10	60	black





# 第十三節課|模組：烏龜填顏色、更多形狀

## 第十三節課|模組|完成機器人吧

接著讓我們畫出流程圖，依順序把機器人的每個部位畫出來，這裡有個畫眼睛的函式請自己練習寫出來。



依據流程圖寫出程式，參考以下程式範例

```
my_pen.goto(-20,80)
draw_rect(40,40,"goldenrod")

my_pen.goto(-40,40)
draw_rect(80,80,"thistle")

my_pen.goto(-60,40)
draw_rect(20,80,"goldenrod")

my_pen.goto(40,40)
draw_rect(20,80,"goldenrod")

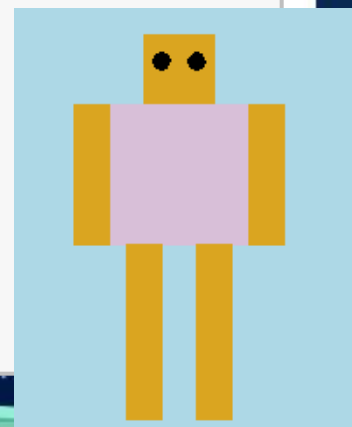
my_pen.goto(-30,-40)
draw_rect(20,100,"goldenrod")

my_pen.goto(10,-40)
draw_rect(20,100,"goldenrod")

draw_eye(-10,60,"black",5)
draw_eye(10,60,"black",5)

my_pen.hideturtle()

screen.exitonclick()
```



# 作品討論與實作

討論主題

把自己的機器人畫得更多細節吧？

繼續作品實作！



tutorJr | 青少兒程式設計

# Python 課程

## 初級課程14

【模組：遞迴-畫出各種形況的萬花筒】



# 初級Python知識點





# 第十四節課|模組：遞迴-畫出各種形狀的萬花筒

## 第十四節課|模組|烏龜萬花筒

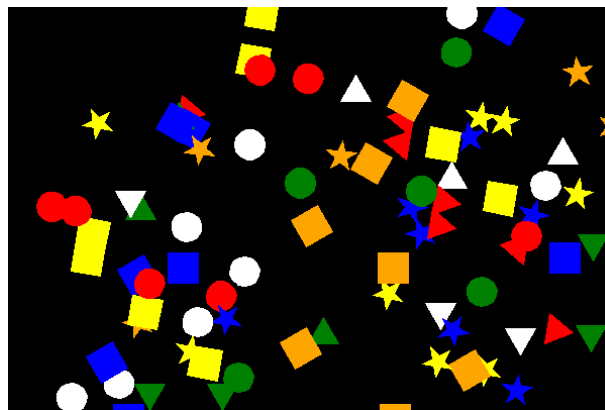
上堂課我們利用烏龜模組來填顏色，也順利的畫了一個機器人，你利用烏龜畫了哪些有趣的作品呢？

讓我們來複習烏龜模組的幾個重要功能：

```
goto(int, int),  
pencolor(str)  
pensize(int),  
speed(int)  
pendown()  
penup()  
right(int)  
forward(int)  
fillcolor(str)  
circle(int)  
goto(int, int)  
end_fill()
```

利用這些功能可以畫出很多漂亮的圖案唷。

這堂課，讓我們應用控制元跟運算子，還有迴圈，做出充滿繽紛色彩各種形狀的萬花筒吧。



### 程式小提醒

要使用烏龜模組前記得要先import  
這個烏龜模組唷 `import turtle`



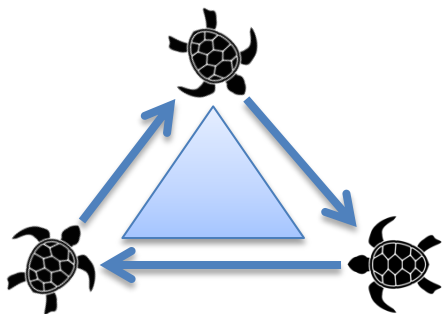
# 第十四節課|模組：遞迴-畫出各種形狀的萬花筒

## 第十四節課|模組|畫出各種形狀

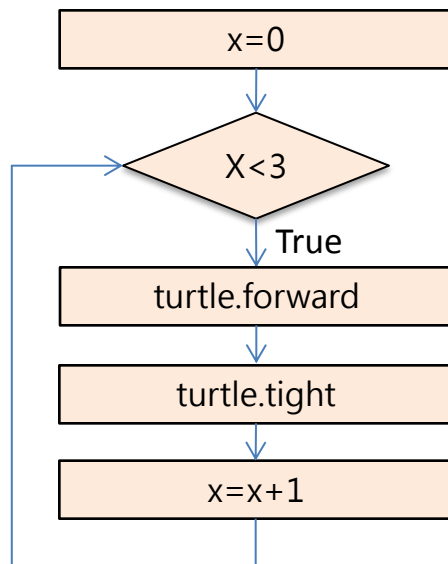
想看看，在萬花筒中最重要的功能是什麼？  
沒錯，就是畫出各式各樣彩色的圖案。請  
思考2個問題

1. 要幾種形狀？
2. 要幾種顏色？

決定後，先利用烏龜畫出這些形狀，例如  
三角形、正方形跟原形要怎麼畫呢？



聯想一下，我們讓烏龜走三角形狀，是讓  
烏龜重複前進一個距離後，往右轉120度。  
程式要如何寫呢？先想看看流程圖



英文小教室

三角形的英文是triangle  
正方形的英文是square  
圓形的英文是circle

# 第十四節課|模組：遞迴-畫出各種形狀的萬花筒

## 第十四節課|模組|畫出各種形狀

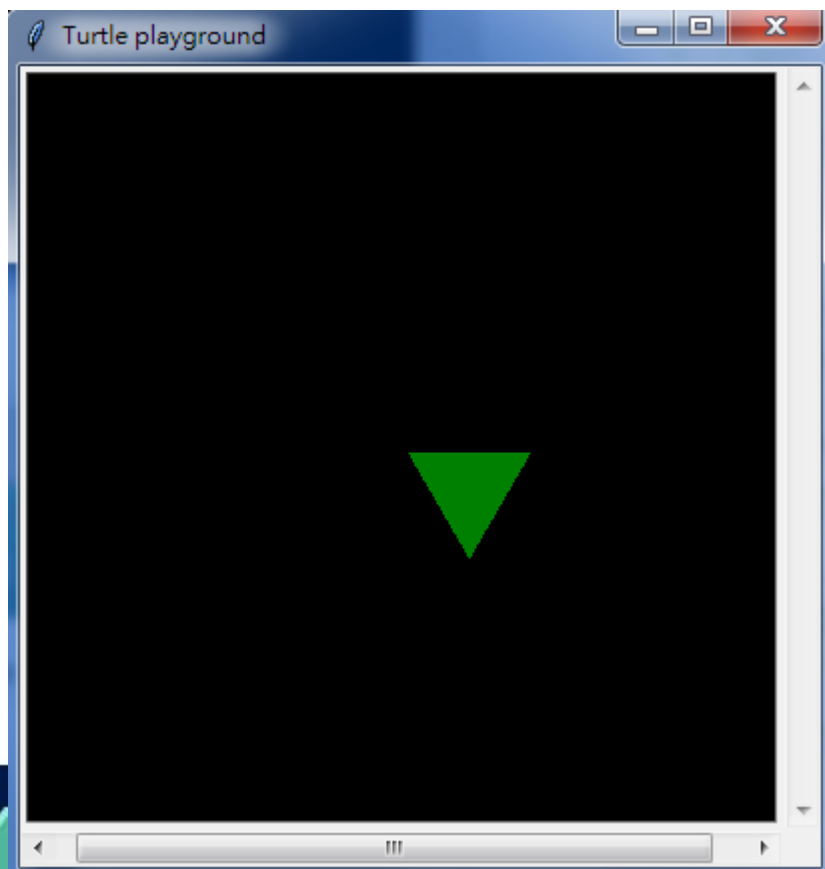
接下來想看看程式要怎麼寫？參考以下的程式範例

```
import turtle # Allows us to use turtles
screen = turtle.Screen()
screen.setup(400, 400)
screen.title('Turtle playground')
screen.bgcolor('black')

my_pen=turtle.Turtle()
my_pen.pencolor("green") #設置畫筆顏色
my_pen.pensize(1) #設置畫筆尺寸
my_pen.speed(10) #設置畫筆畫的速度
my_pen.hideturtle()
my_pen.goto(0,0)

my_pen.fillcolor("green")
my_pen.begin_fill()
for i in range(3):
    my_pen.forward(60)
    my_pen.right(120)
my_pen.end_fill()
```

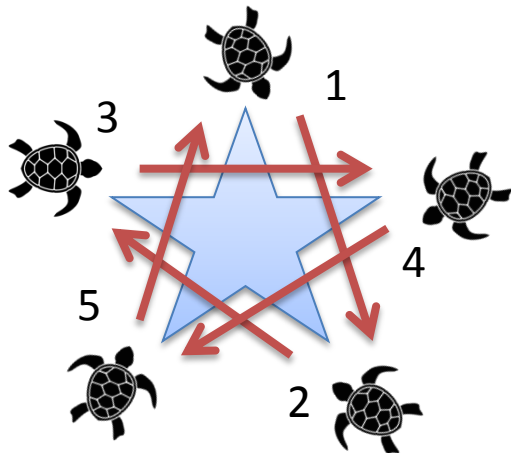
執行看看是否可以畫出一個三角形。  
自己試著畫出圓形跟正方形吧



## 第十四節課|模組：遞迴-畫出各種形狀的萬花筒

### 第十四節課|模組|畫出多邊形

想看看三角形的畫法，要如何畫出星形跟多變形狀呢？來看看烏龜怎麼走路的



烏龜直線走了一個距離後會右轉一個角度，那個角度是多少？

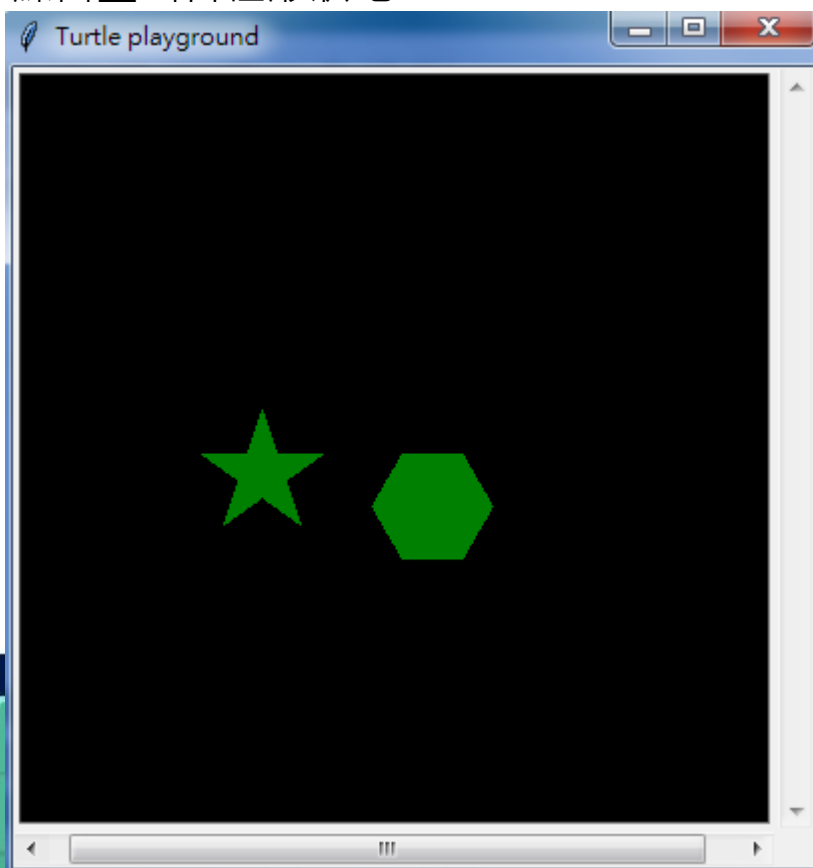
$$180-(180/5)$$

試著自己畫出流程圖跟寫出程式吧

那等邊六邊形呢？不同的是烏龜右轉的角度，思考一下，這個角度要如何算出來？

$$360/6$$

練習畫出各種形狀吧



## 第十四節課|模組：遞迴-畫出各種形狀的萬花筒

### 第十四節課|模組|畫萬花筒的函式

我們已經知道如何畫出各種圖樣，接著我們先把要畫的形狀跟顏色紀錄下來。之前我們學習for-loop時有用到list來計錄順序，現在我們也可以用list來紀錄形狀跟顏色，請看看以下的程式範例：

```
colors=["red","blue","green","yellow","white","orange"]  
sharps=["circle","square","triangle","star"]
```

建立兩個list，一個紀錄顏色，另一個紀錄形狀，這樣我們也可以依照順序畫出各種顏色的形狀囉。

接著來想想這個畫出萬花筒的函式需要哪些參數？

1. 大小； 2. 角度； 3. 位移

第二個參數跟第三個參數是讓每個形狀都能夠在不同的地方。

接下來，讓我們思考這個函式要做的事情有哪些？

1. 照list畫出形狀跟填顏色。
2. 移動到另一個地方
3. 呼叫畫萬花筒的函式再畫其它形狀

第三件事情需要利用函式來呼叫同一個函式，也就是遞迴(Recursion)的做法。

萬花筒裡有成千上萬各種形狀跟顏色的圖案，我們拆解為畫一個顏色形狀的函式，然後反覆呼叫，就是簡單的遞迴操作唷

#### 程式小提醒

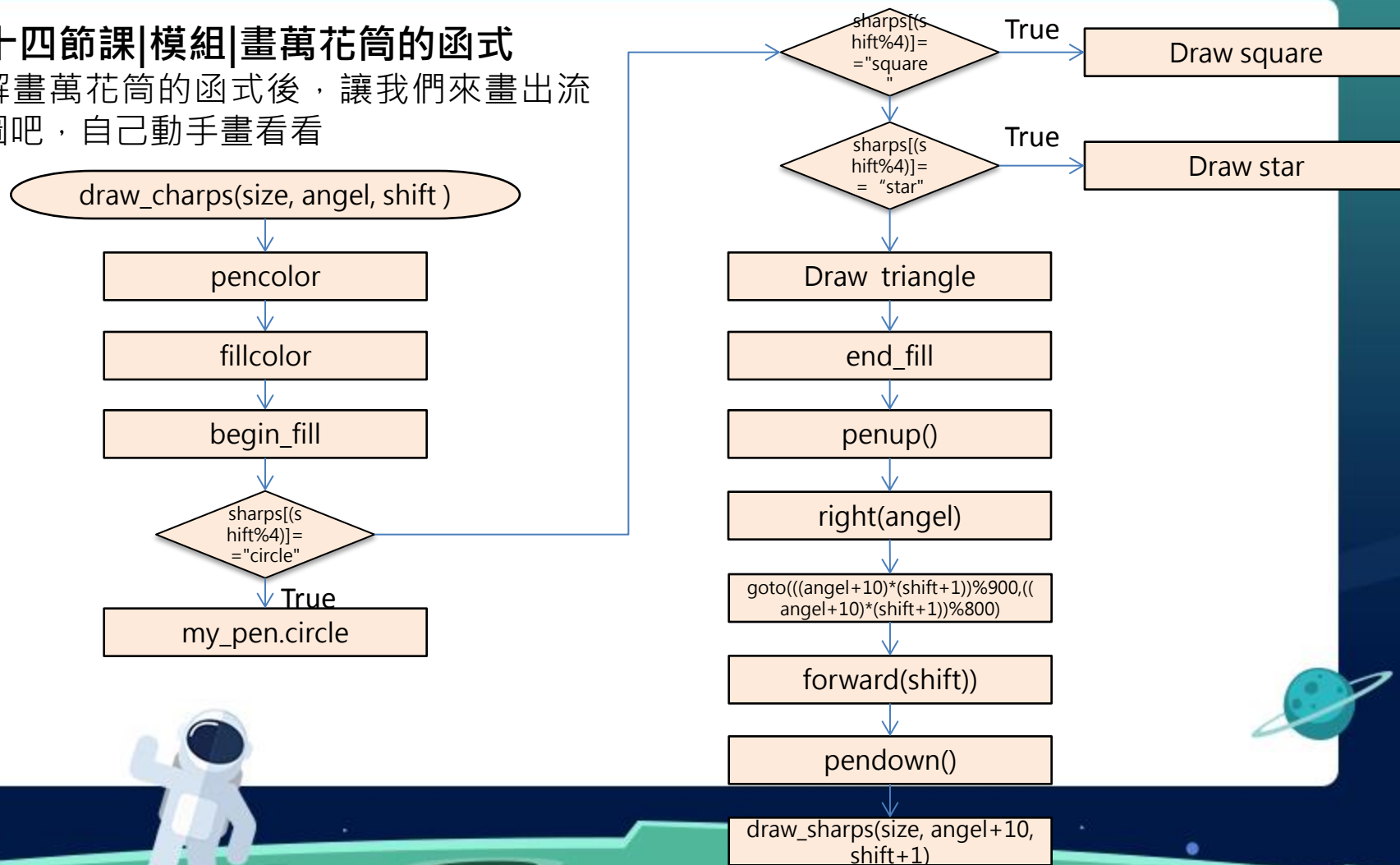
要取出list中的資料，可以用list[x]，例如第1項就是list[0]，因為list是從0開始唷



# 第十四節課|模組：遞迴-畫出各種形狀的萬花筒

## 第十四節課|模組|畫萬花筒的函式

了解畫萬花筒的函式後，讓我們來畫出流程圖吧，自己動手畫看看

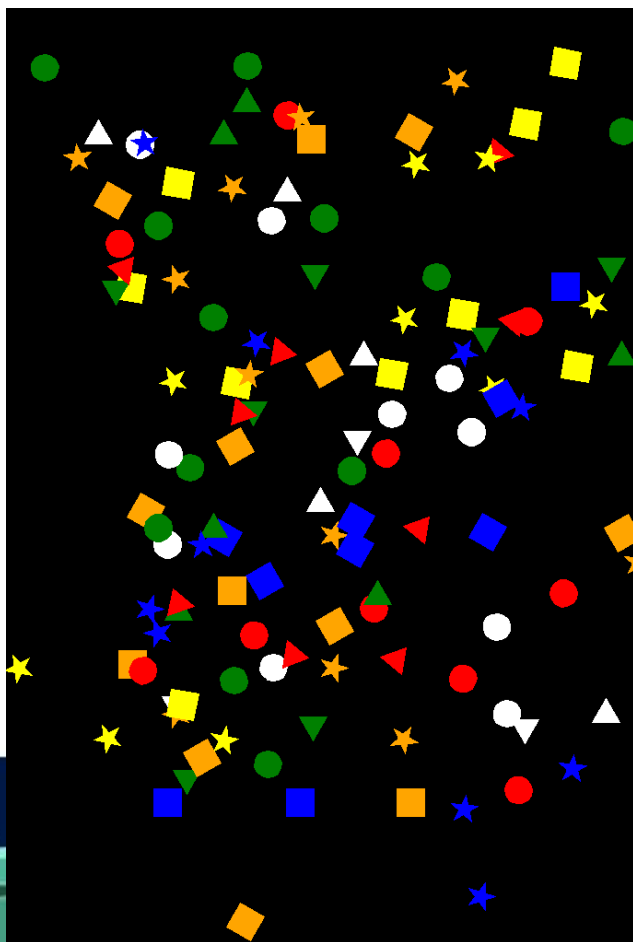




# 第十四節課|模組：遞迴-畫出各種形狀的萬花筒

## 第十四節課|模組|畫萬花筒的函式

開始照著流程圖動手寫程式吧，參考以下的程式範例



```
import turtle # Allows us to use turtles
screen = turtle.Screen()
screen.setup(400, 400)
screen.title('Turtle playground')
screen.bgcolor('black')

my_pen=turtle.Turtle()
my_pen.pencolor("green") #設置畫筆顏色
my_pen.pensize(1) #設置畫筆尺寸
my_pen.speed(10) #設置畫筆畫的速度
my_pen.hideturtle()
my_pen.goto(0,0) #移動畫筆到座標(0,0)的位置

colors=["red","blue","green","yellow","white","orange"]
sharps=["circle","square","triangle","star"]

def draw_sharps(size, angle, shift):
    my_pen.pencolor(colors[(shift%6)])
    my_pen.fillcolor(colors[(shift%6)])
    my_pen.begin_fill()

    if(sharps[(shift%4)]=="circle"):
        my_pen.circle(size/2)
    elif(sharps[(shift%4)]=="square"):
        for i in range(4):
            my_pen.forward(size)
            my_pen.right(90)
    elif(sharps[(shift%4)]=="star"):
        for i in range(5):
            my_pen.forward(size)
            my_pen.right(180-(180/5))
    else:
        for i in range(3):
            my_pen.forward(size)
            my_pen.right(120)

    my_pen.end_fill()
    my_pen.penup()
    my_pen.right(angle)
    my_pen.goto(((angle+10)*(shift+1))%900,((angle+10)*(shift+1))%800)
    my_pen.forward(shift)
    my_pen.pendown()
    draw_sharps(size, angle+10, shift+1)

draw_sharps(30,0,1)

screen.exitonclick()
```



# 作品討論與實作

討論主題

做出有自己特色的萬花筒吧！

繼續作品實作！



tutorJr | 青少兒程式設計



# Python 課程

## 初級課程15

【迴圈控制：break / continue 的使用】



# 初級Python知識點





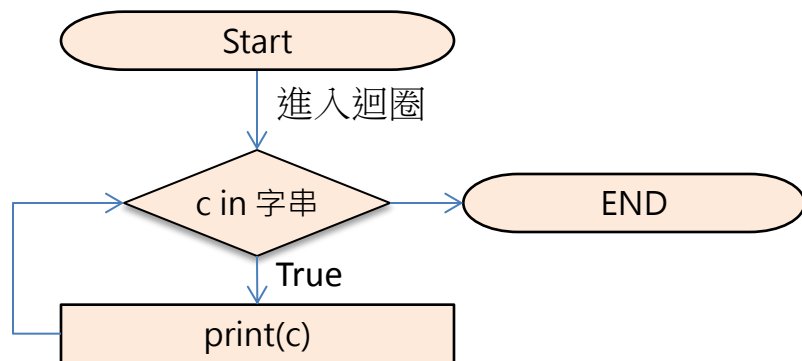
# 第十五節課|迴圈控制：如何中斷迴圈

## 第十五節課|迴圈控制|複習for-loop

利用迴圈可以讓程式重複執行一段程式碼，把程式寫出來，參考以下程式範例並執行。直到滿足某個條件，常用迴圈如for-loop

**for x in 順序:**  
要執行的事情

讓我們用一個簡單的例子來複習，請印出字串"hello python"裡的每個字元，先想想流程圖唷



```
for c in "hello python":  
    print(c)
```

h  
e  
l  
l  
o  
  
p  
y  
t  
h  
o  
n

### 程式小提醒

迴圈條件後記得加上**冒號**：同時注意程式縮排





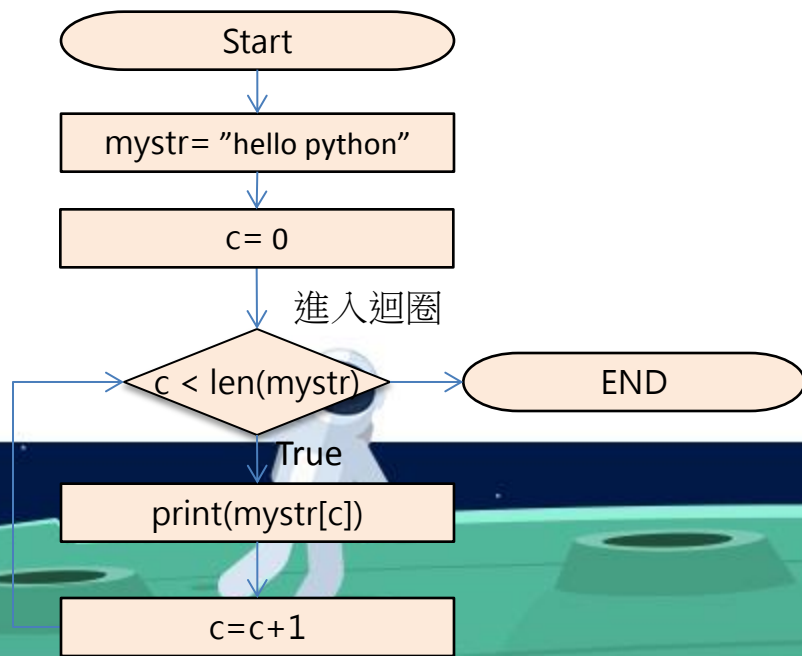
# 第十五節課|迴圈控制：如何中斷迴圈

## 第十五節課|迴圈控制|複習while-loop

還有另一個常用迴圈就是while-loop，程式會一直執行迴圈內的程式直到某個條件成立，用法如下

while 結束條件描述：  
迴圈內的程式

讓我們用同一個例子來複習while-loop，請印出字串"hello python"裡的每個字元



把程式寫出來，參考以下程式範例並執行。

```
mystr="hello python"
c=0
while (c<len(mystr)):
    print(mystr[c])
    c+=1
```

h  
e  
l  
l  
o  
  
p  
y  
t  
h  
o  
n

程式小提醒

清單的開始第一個元素預設編號是 0





## 第十五節課|迴圈控制：如何中斷迴圈

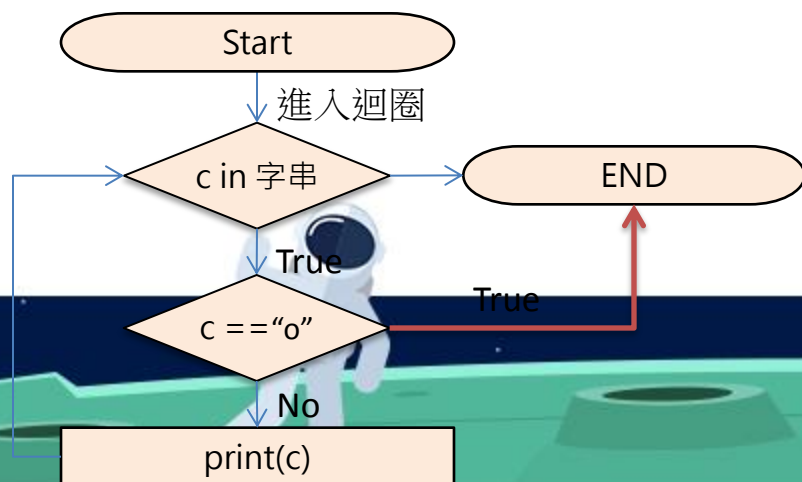
### 第十五節課|迴圈控制|中斷迴圈-break

迴圈裡的程式何時會結束而離開迴圈？一般都是當進入迴圈的條件滿足後進入，不滿足後離開。如果我們要再迴圈裡強制離開迴圈的話，我們就需要中斷迴圈，這裡介紹兩種方式來中斷迴圈：

#### break

讓我們用同一個例子來看看break如何中斷迴圈，請印出字串"hello python"裡的每個字元，但遇到"o"時停止。

先看看流程圖怎麼畫



把程式寫出來，參考以下程式範例並執行。

```
#break
for val in "hello python":
    if val=="o":
        break
    print(val)
print("The end")
```

```
h
e
l
l
The end
```

原本應該要印出"o python"的部分因為要印的字元滿足是"o"所以就被break中斷後續的程式，跳出迴圈執行下一個程式碼。





## 第十五節課|迴圈控制：如何中斷迴圈

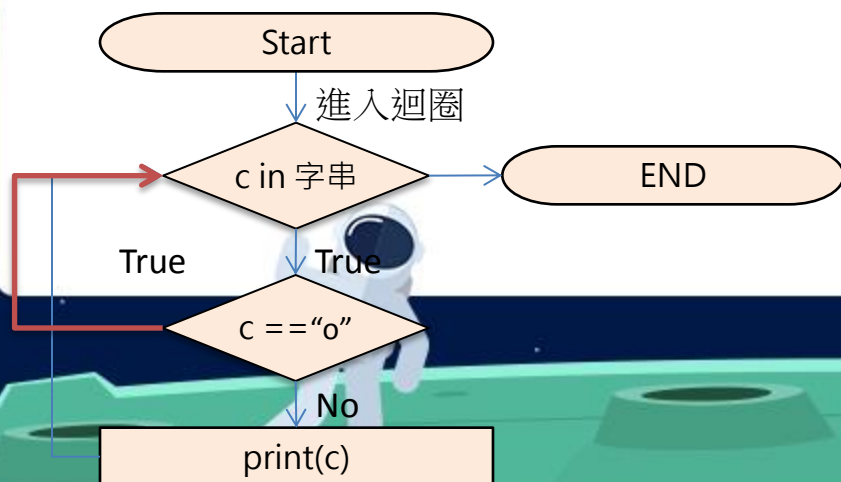
### 第十五節課|迴圈控制|中斷迴圈-continue

另一種情況是只有當迴圈裡的某一個條件成立時中斷，但仍會持續做剩下的程式當條件不成立時，這時就要使用另一個指令

#### continue

讓我們用同一個例子來看看continue如何中斷迴圈以及跟break的不同，請印出字串"hello python"裡的每個字元，但遇到"o"時不要印"o"。

先看看流程圖怎麼畫。



把程式寫出來，參考以下程式範例並執行。

```
#continue
for val in "hello python":
    if val=="o":
        continue
    print(val)
print("The end")
```

```
h
e
l
l

p
y
t
h
n
The end
```

當發現字元"o"之後，中斷原本要印出字元的程式，但是不離開迴圈，而是跳過要執行的程式繼續迴圈。





# 第十五節課|迴圈控制：如何中斷迴圈

## 第十五節課|迴圈控制|移除不要的數字

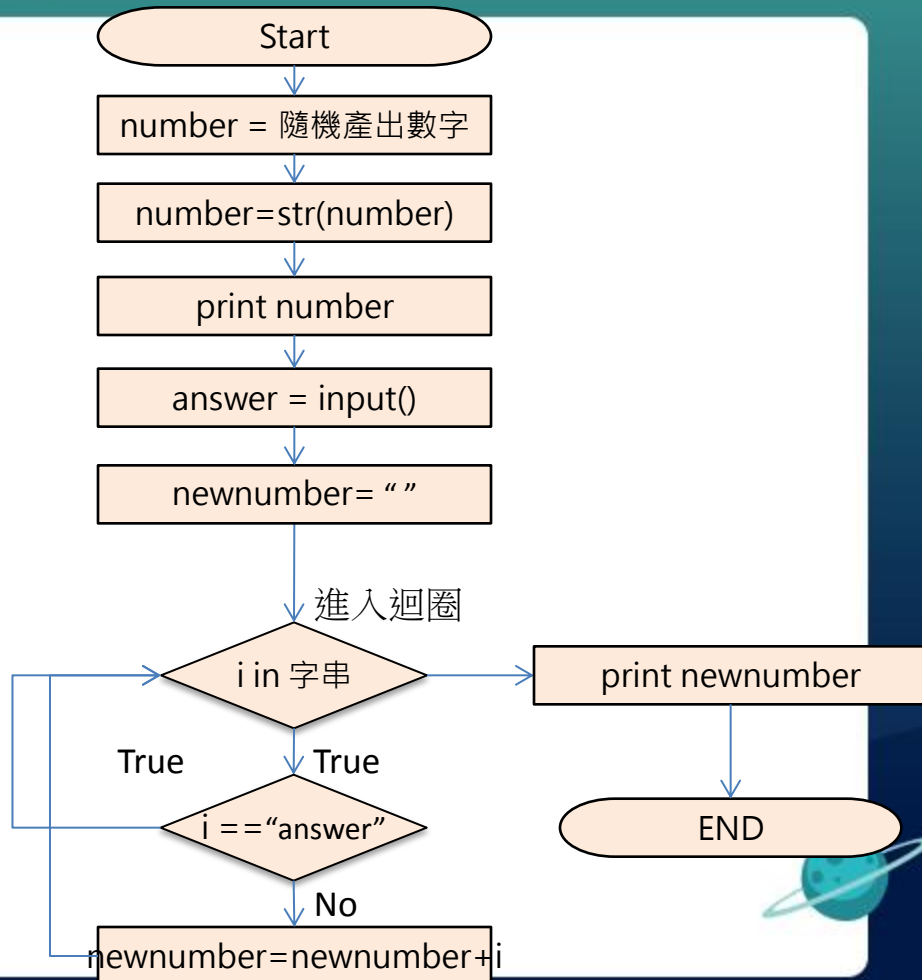
學習如何中斷迴圈後，我們來練習，如何從一串數字移除某個數字？程式重點看看

1. 利用隨機產生1組10位數整數並顯示
2. 輸入一個要移除的數字
3. 從隨機整數中移除輸入的數字

先別急著寫，先畫出流程圖吧

### 程式小提醒

1. 資料類型要轉換，把整數轉換為字串
2. 可以利用字串運算子把兩個字串串起來





## 第十五節課|迴圈控制：如何中斷迴圈

### 第十五節課|迴圈控制|移除不要的數字

開始寫程式吧，參考以下的程式範例

```
#從一組數字中移除某個數字
import random
number=int(random.random()*1000000000)
print("Hi, I have a number ", number)
answer=input("Please input the number you do not want ")
print("The number you do not want is", answer)
number=str(number)
newnumber=""
for i in number:
    if i==answer:
        continue
    newnumber=newnumber+i;
print(number)
print(newnumber)
```

```
Hi, I have a number  388790233
Please input the number you do not want 3
The number you do not want is 3
388790233
887902
```

再接再厲，寫一個程式隨機產生一組數字不包含某個數字。自己畫流程圖並且參考以下的程式範例

```
#隨機產生一組數字不包含某個數字
import random
number=""
number_length=0
throw_number=int(input(""))
while(number_length<10):
    x=int(random.random()*10)
    if x==throw_number:
        continue
    number=number+str(x)
    number_length+=1
print(number)
```

```
3
7424087078
```





# 第十五節課|迴圈控制：如何中斷迴圈

## 第十五節課|迴圈控制|電腦自己猜數字

還記得第7堂課我們寫的猜數字作品嗎？電腦隨機產生一個數字讓我們來猜，電腦只會顯示猜的數字太大還是太小。今天讓我們來寫個程式讓電腦自己來猜數字吧

1. 輸入最大的數字
2. 電腦隨機再最大的數字跟0間產生要猜的數字
3. 演算法練習：猜最大跟最小數字的中間值
4. 依據結果調整最大跟最小的數字

先別急著寫，先試著自己畫出流程圖吧

### 程式小提醒

`randint(min, max)` 可以隨機產出min跟max之間的整數

#電腦自己猜數字

```
import random
max_number=int(input("請輸入最大的數字"))
number=random.randint(1, max_number)
guess_number=int(max_number/2)
print(number)
print("====開始猜囉====")
r_number=0
l_number=max_number
guess_count=0
while 1:
    guess_count+=1
    if(guess_number==number):
        break
    elif(guess_number>number):
        print(guess_number," 太大")
        l_number=guess_number
        guess_number=int(r_number+(l_number-r_number)/2)
    else:
        print(guess_number," 太小")
        r_number=guess_number
        guess_number=int(r_number+((l_number-r_number)/2))
print("I guess right and i use ", guess_count, "times")
```

請輸入最大的數字1000

862

====開始猜囉====

500 太小

750 太小

875 太大

812 太小

843 太小

859 太小

867 太大

863 太大

861 太小

I guess right and i use 10 times

# 學習報告與作品討論

## 學習報告

掃QR Code看看自己的學習報告



## 討論主題

發揮創意，想更多地球溫度的解決方式

繼續作品實作！



tutorJr | 青少兒程式設計

# Python 課程

## 初級課程16

【清單操作：清單的基本使用】



# 初級Python知識點







# 第十六節課|清單操作：清單的基本操作

## 第十六節課|清單操作|什麼是清單

什麼是清單？當我們有很多東西要存放的時候，我們會需要一個包包來存放，而程式裡，清單list就是這樣的一個資料類型(包包)，用來存放很多不同類型的資料。

建立一個清單

```
my_list=[elements]
```

試著建立以下清單並且印出來：空清單、整數清單、字串清單跟複合資料清單，參考以下程式範例。

```
#all kind of list
my_empty_list=[]
my_int_list=[1,2,3]
my_str_list=["hi","my", "friends"]
my_mixed_list=[1, "hello",True, 1.1]
print(my_int_list)
print(my_str_list)
print(my_mixed_list)
```

執行程式看結果。

```
[1, 2, 3]
['hi', 'my', 'friends']
[1, 'hello', True, 1.1]
```

清單裡可以放另一個清單嗎？  
自己試試看並且執行看看

```
my_int=[1,2,3,[4,5,6],7]
print(my_int)

[1, 2, 3, [4, 5, 6], 7]
```

可以的，這也叫做nested list

### 程式小提醒

清單的開始第一個元素預設編號是0



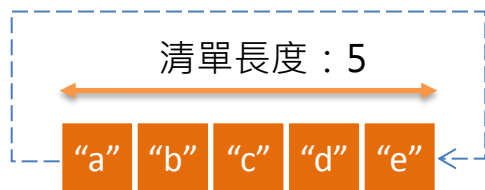




# 第十六節課|清單操作：清單的基本操作

## 第十六節課|清單操作|找到資料

原來清單可以想成是可大可小裝任何資料的大袋子，那要如何從清單中找到資料呢？首先我們要先了解資料在清單中是怎麼排放的，看看底下的圖



index 編號： 0 1 2 3 4

negative index： -5 -4 -3 -2 -1

我們可以利用以下方式拿到某一筆資料

清單[index]

### 程式小提醒

清單的第一筆資料編號是0，右邊的是1、2...，左邊的是清單最後一筆資料，所以編號就是-1

建一個清單，試著印出某一筆資料看看。

```
#access a range of a list
#print a list with for-loop
my_list=[1,2,3,4,5,6,7,8]
for i in range(5):
    print(my_list[i])

for j in range(len(my_list)):
    print(my_list[j])

print(my_list[1:4])
```

```
1
2
3
4
5
6
7
8
[2, 3, 4]
```



## 第十六節課|清單操作：清單的基本操作

### 第十六節課|清單操作|替換資料

如果想要把清單裡的資料替換掉，要如何操作呢？其實很簡單，把清單裡的每一筆資料所放的地方都想成是一個變數，找出來後直接指定新的資料就可以囉。

如果要替換的一個範圍內的資料，只要找出這些資料後指定相同數量的資料。看看以下的程式範例

```
#change an element in the list
#change all elements in the range in the list
my_abc=["a","b","c","d","e"]
print(my_abc)
for i in range(5):
    print(my_abc[i])
my_abc[1]="f"
print(my_abc)
my_abc[0:3]=["x","y","z"]
print(my_abc)
```

執行看看結果吧

```
['a', 'b', 'c', 'd', 'e']
a
b
c
d
e
['a', 'f', 'c', 'd', 'e']
['x', 'y', 'z', 'd', 'e']
```

#### 程式小提醒

表示清單裡的範圍資料用  
清單[開始index: 結束index]





# 第十六節課|清單操作：清單的基本操作

## 第十六節課|清單操作|新增資料

我們有了可以裝很多資料的清單，那要如何把資料放到清單中呢？

清單提供了幾個方式來增加資料到清單

`append(element)`：加一筆資料

`extend(list)`：加一個清單

`+`：加一個清單

`*`：重複加上一筆資料

參考以下的程式範例，練習以上各種方法來新增資料到清單中吧

```
#append and extend elements to a list
#operation to a list and lists
my_abc=["a","b","c","d","e"]
print(my_abc)
my_abc.append("f")
print(my_abc)
my_abc.extend(["g","h","i"])
print(my_abc)
# +
my_abc=my_abc+["j","k","l"]
print(my_abc)
# *
my_abc=my_abc+(["m"]*3)
print(my_abc)
```

執行程式看結果。

```
['a', 'b', 'c', 'd', 'e']
['a', 'b', 'c', 'd', 'e', 'f']
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l']
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'm', 'm']
```

還可以插入一筆資料到任何位置唷

`insert(index, data)`

試試以下的程式範例並且執行

```
#insert a element to a specified location
my_abc=["a","b","c","d","e"]
my_abc.insert(0, "z")
print(my_abc)
my_abc.insert(6, "g")
print(my_abc)
my_abc.insert(6, "f")
print(my_abc)
```

```
['z', 'a', 'b', 'c', 'd', 'e']
['z', 'a', 'b', 'c', 'd', 'e', 'g']
['z', 'a', 'b', 'c', 'd', 'e', 'f', 'g']
```

程式小提醒

清單裡的每一筆資料我們說是一個  
element



# 第十六節課|清單操作：清單的基本操作

## 第十六節課|清單操作|刪除資料

最後，我們裝了好多好多資料到清單，要跟大家說如何刪除清單裡的資料囉

`del list[index]`：刪除清單某個編號的資料  
`list[index].remove`：刪除某個編號的資料

參考以下的程式範例，練習以上各種方法來刪除資料到清單中吧

```
#delete an element in the list
my_number=[1,2,3,4,5]
print(my_number)
del my_number[2]
print(my_number)
del my_number[1:3]
print(my_number)
```

看看執行結果

```
[1, 2, 3, 4, 5]
[1, 2, 4, 5]
[1, 5]
```

如果刪除整個清單呢？看看以下程式範例並執行，了解為何會有error

```
del my_number
print(my_number)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-26-19a8e947f969> in <module>
      7 print(my_number)
      8 del my_number
----> 9 print(my_number)

NameError: name 'my_number' is not defined
```

接著練習用remove來刪除資料

```
#remove an element in the list
my_number=[1,2,3,4,5]
print(my_number)
my_number.remove(1)
print(my_number)
```

```
[1, 2, 3, 4, 5]
[2, 3, 4, 5]
```





# 第十六節課|清單操作：清單的基本操作

## 第十六節課|清單操作|學生成績資料

讓我們利用清單，做一個簡單的學生成績資料工具吧，功能如下：

1. 輸入學生姓名
2. 輸入學生成績
3. 印出所有學生的成績

記得先畫流程圖，然後再寫程式，這個功能很簡單，自己試試看吧

```
student_score=[]
student_name=[]
for i in range(3):
    student_name.append(input("Please enter student name: "))
    student_score.append(input("Please enter student score: "))

for i in range(len(class_score)):
    print(student_name[i], student_score[i])
```

```
Please enter student name: Jerry
Please enter student score: 100
Please enter student name: Peter
Please enter student score: 93
Please enter student name: Penny
Please enter student score: 99
Jerry 100
Peter 93
Penny 99
```



# 作品討論與實作

討論主題

利用清單做出有用有趣的工具吧！

繼續作品實作！



tutorJr | 青少兒程式設計

# Python 課程

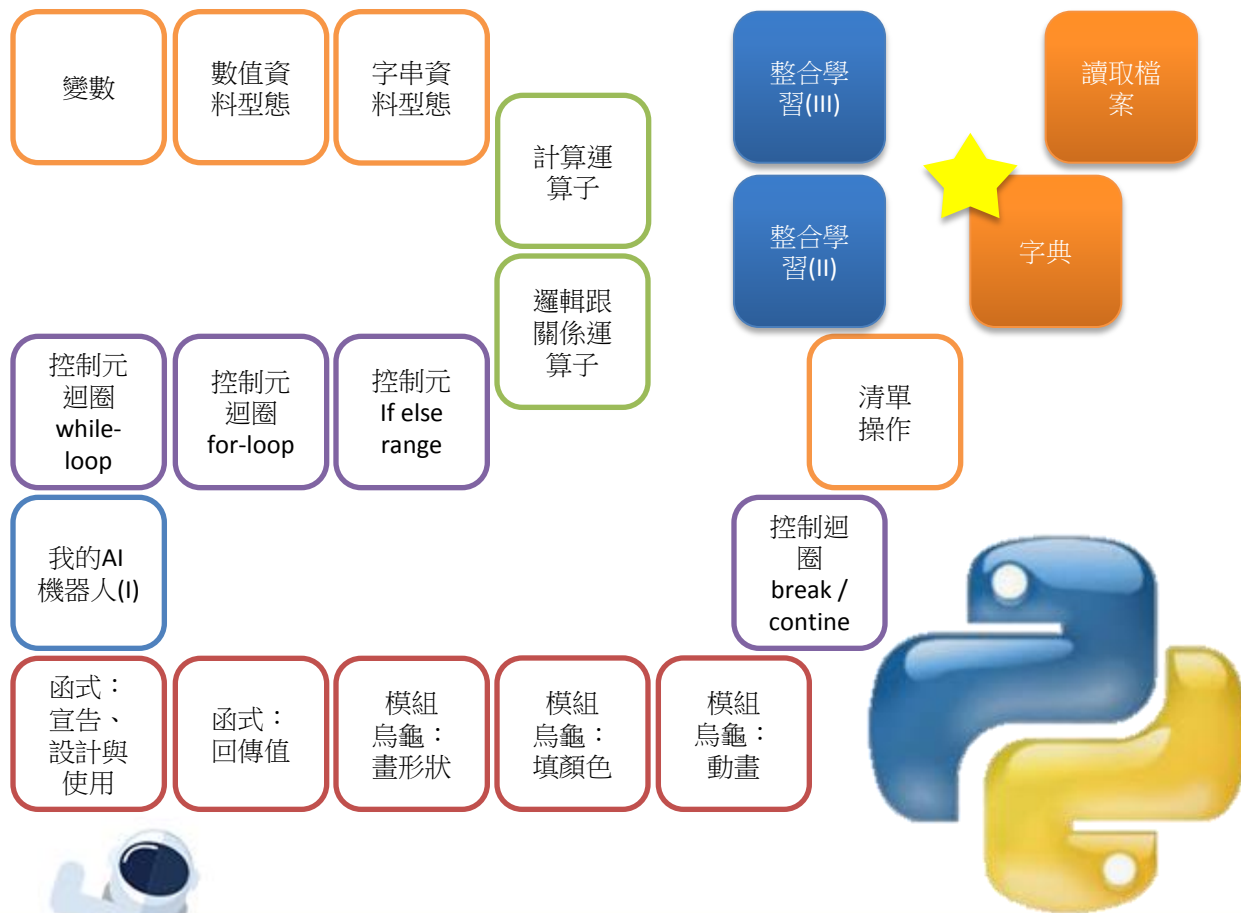
## 初級課程17

【字典操作：字典的基本使用】





# 初級Python知識點





# 第十七節課|字典：字典的基本操作

## 第十七節課|字典|什麼是字典

什麼是字典？當我們要存放的東西不只指很多，而且還很多樣化，並且，我們希望要取出這些資料時很簡單說出是哪類東西，不需要知道放的確切位置，如果有這樣的大包包該有多好，字典，就有這樣的功能。建立一個字典的方式，利用大括號：

```
mydict={key1:value1,key2:value2}
```

這裡的key需要使用引號唷

另一種方式，利用dict() 函式：

```
Mydict=dict(key1=value1, key2=value2)
```

這裡的key就不需要引號，是dict函式裡的參數。

試著建立以下清單並且印出來：兩種建立字典的方式，建立自己資料的字典，參考以下程式範例。

```
#第一次建立字典的方式，直接使用大跨號
mydict1={"name": "John", #Key要用引號，冒號後面接資料
        "Age":13,
        "height":160,
        "weight":63,
        "score":"B-"}
print(mydict1)
```

```
#第二種建立字典的方式，利用dict函式
mydict2=dict(name="Ariel", #Key不須引號，等於後接資料
            Age=11,
            height=148,
            weight=31,
            score="A+")
print(mydict2)
```

執行程式看結果。

```
{'name': 'John', 'Age': 13, 'height': 160, 'weight': 63, 'score': 'B-'}
{'name': 'Ariel', 'Age': 11, 'height': 148, 'weight': 31, 'score': 'A+'}
```





# 第十七節課|字典：字典的基本操作

## 第十七節課|字典|字典的好處

使用字典有那些好處呢？以剛剛的練習，我們把學生的資料用字典建立起來，當然一個學生的資料會更多。當我們需要某個學生的成績資料時，利用清單的話，我們需要把學生的各種資料都建立成一個個的清單，然後去找成績放在哪個清單，接著找這個學生的成績放在哪個位置...好累喔。

name	John	Ariel
Age	13	10
height	160	148
weight	63	31
score	B-	A+



如果用字典存放每個學生的資料呢？只要問某個學生的成績一個動作就可以啦。

John	
John	
13	
160	
63	
B-	1

是不是很方便呢。字典用於大量的資料數據整理上也是很常用到的唷。

英文小教室

字典的英文：directory





# 第十七節課|字典：字典的基本操作

## 第十七節課|字典|基本操作-取值

我們已經學習如何建立一個字典，要如何把字典裡的值取出來呢？

`mydict[key]`

接下來我們練習把John的姓名跟年齡資料從他的字典內取出來，看看以下程式範例

```
print(mydict1["name"])
print(mydict1["Age"])#搜尋時的key要用引號
```

執行看看是否正確

```
John
13
```

也可以利用變數來存放取出的值，看看以下的程式範例

```
myname=mydict2["name"]
myage=mydict2["Age"]
print(myname)
print(myage)
```

執行看看是否正確。

```
Ariel
11
```

要注意字典裡的資料型態取出後還是要注意相同資料型態的資料才能處理唷。試試以下的程式範例，是否有問題？

```
myname=mydict2["name"]
myage=mydict2["Age"]
myname+=myage #將姓名跟年齡相加
print(myname)
```

執行程式看結果，並且試著修復錯誤。

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-27-5f8443b16f45> in <module>
      1 myname=mydict2["name"]
      2 myage=mydict2["Age"]
----> 3 myname+=myage #將姓名跟年齡相加
      4 print(myname)

TypeError: can only concatenate str (not "int") to str
```



# 第十七節課|字典：字典的基本操作

## 第十七節課|字典|基本操作-新增資料

字典建立好了，但是我有新的資料元素要加進去字典要怎麼做呢？很簡單唷

```
mydict[new_key]=value
```

試試以下的程式範例

```
mydict1["grade"]=7  
print(mydict1)
```

執行看看，看看字典是否多了一個資料

```
{'name': 'John', 'Age': 13, 'height': 160, 'weight': 63, 'score': 'B-', 'grade': 7}
```

簡單吧，也把你的年級資料加進去字典唷

如果要把很多新的元素加進去字典裡呢？  
我們可以把這些新的元素建成一個新的字典，然後用update把更新到原來的字典

```
original_dict.update(new_dict)
```

我們新增英文成績、數學成績跟電腦成績到學生的字典裡，要怎麼做呢？自己試試看，參考以下的程式範例

```
myclass1={"math":89,  
          "english":92,  
          "computer":81}  
mydict1.update(myclass1)  
print(mydict1)
```

執行看看是否正確？

```
{'name': 'John', 'Age': 13, 'height': 160, 'weight': 63,  
 'score': 'B-', 'grade': 7, 'math': 89,  
 'english': 92, 'computer': 81}
```





# 第十七節課|字典：字典的基本操作

## 第十七節課|字典|基本操作-刪除資料

如果要把資料從字典裡刪除要怎麼做呢?其實也很簡單唷，利用del刪除某一資料的話

```
del mydict[key]
```

試試以下的程式範例

```
del mydict1["computer"]  
print(mydict1)
```

執行看看，看看computer的資料是否已經不見了？

```
{'name': 'John', 'Age': 13, 'height': 160,  
 'weight': 63, 'score': 'B-', 'grade': 7,  
 'math': 89, 'english': 92}
```

簡單吧，試看看把自己的資料一些刪除掉

還有一種方式是利用pop，把不要的資料pop出來，我們可以用變數接起來再利用

```
Data=mydict.pop(key)
```

試試以下的程式範例

```
mycomputerscore=mydict1.pop("computer")  
print(mydict1)  
print(mycomputerscore)
```

執行看看是否正確？

```
{'name': 'John', 'Age': 13, 'height': 160,  
 'weight': 63, 'score': 'B-', 'grade': 7,  
 'math': 89, 'english': 92}  
81
```







# 第十七節課|字典：字典的基本操作

## 第十七節課|字典|實作-學生查詢系統

利用字典來做一個簡單的學生資料查詢系統吧，這個系統有幾個功能

1. 輸入學生的姓名、年齡跟年級
2. 輸入要查詢的學生姓名
3. 輸入要查詢的學生資料
4. 輸出查詢的結果

流程說明：

1. 這個系統會用到清單，利用清單來建立有幾個學生，然後在這個清單內的每個學生都有自己的字典資料。
2. 依序輸入每個學生的資料完成學生的字典
3. 輸入要查詢的學生資料
4. 利用for-loop檢查每個字典裡學生姓名是否跟輸入的姓名相同
5. 相同的話輸出該字典裡要查詢的item資料

別急著寫程式，照著流程說明畫出流程圖。參考以下的程式範例

```
student=["student1","student2"]
for i in range(2):
    name=input("please enter the student's name ")
    age=input("please enter the student's age ")
    grade=input("please enter the student's grade ")
    student[i]={"name":name,
                "age":age,
                "grade":grade}
    print(student[i])
    print("=====Next")

checkname=input("please enter the student's name ")
checkitem=input("please enter the item ")
for j in range(2):
    if(student[j]["name"]==checkname):
        print("The data is: "+student[j][checkitem])
        break;
```

執行看看結果吧





# 作品討論與實作

討論主題

利用字典做出有用有趣的程式吧！

繼續作品實作！



tutorJr | 青少兒程式設計

# Python 課程

## 初級課程18

【檔案操作：讀取檔案】



# 初級Python知識點





# 第十八節課|檔案：檔案的基本操作

## 第十八節課|檔案|為何要使用檔案

在上堂課我們學習到字典，利用字典可以很有效率地在很多數據內找到想要的資料。這時候會出現另一個問題，要怎麼把大量的數據資料寫到字典裡呢？或是當我們需要的資料已經存放在一個檔案內時，要如何讀取出來？因此，python程式裡也提供了檔案處理的方法：

### 檔案開啟

```
f = open(檔名, mode)
```

利用一個變數f來存放一個檔案物件，這個物件提供了很多的函式跟方法讓我們來處理檔案，例如利用open()來開啟檔案。

**檔名：**要開啟的檔案名稱，這是一個字串，同時，要使用相對路徑，也就是跟程式存放位置相對的路徑，看看以下的檔案清單：

名稱	修改日期	類型
tutorJr_python_chap18_smamplecode_file...	2021/1/12 上午 0...	IPYNB 檔案
python_data.txt	2021/1/12 上午 0...	文字文件

python\_data.txt檔案跟範例程式在同一個路徑下時，open()函式裡的檔案參數就可以寫成

“python\_data.txt”

**模式：**檔案開啟的模式決定可以如何處理這個檔案，以下我們介紹常見的三個模式：

### 程式小提醒

檔案的名稱也是字串，記得要用“ ”或是“ ”來表示字串。





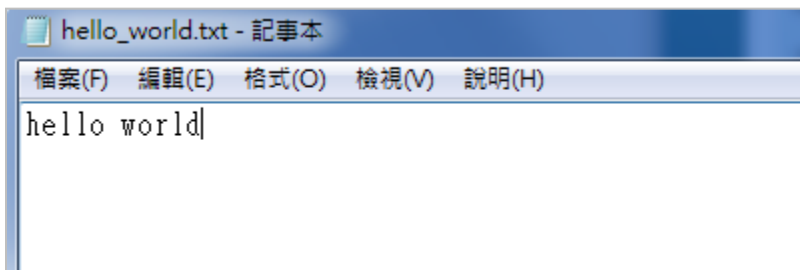
# 第十八節課|檔案：檔案的基本操作

## 第十八節課|檔案|模式

開啟檔案的模式主要介紹3種：

“r”：唯讀模式，利用這個模式讀取檔案內容，但是無法修改或是新增內容到檔案裡。如果檔案不存在會有error唷。

我們先練習利用r模式來開啟檔案吧，先建立一個檔案hello\_world.txt，檔案裡輸入hello world內容，然後儲存完成這個檔案



參考以下的程式範例，把檔案讀取出來。

```
f=open("hello_world.txt","r")  
print(f)
```

```
<_io.TextIOWrapper name='hello_world.txt' mode='r' encoding='cp950'>
```

怎麼不是hello world呢？因為f是檔案物件，而非檔案內容唷。我們先看其他兩個模式：

“w”：寫入模式(覆寫)，利用這個模式開啟檔案可以進行內容的寫入且覆寫原本的檔案，如果檔案不存在就會建立新的檔案。

“a”：寫入模式(續寫)，利用這個模式開啟檔案可以進行內容的寫入且接續在原本的內容後面。





# 第十八節課|檔案：檔案的基本操作

## 第十八節課|檔案|讀取檔案

接下來讓我們開始把檔案裡的內容讀取出來吧。檔案物件提供了read函式來讀取：

`f.read([size])`

參數size非必要的，沒有指定要讀取的大小，就會把檔案裡的內容都讀出來，否則只會讀取設定的size大小內容。參考以下的程式範例，試試看執行結果是否正確。

```
f=open("hello_world.txt","r")
words=f.read()
print(words)
f.close()
```

hello world

記得最後不再對檔案進行處理時，用f.close把檔案關閉唷，養成好習慣

接著我們試試指定要讀取的長度，看看結果如何，參考以下的程式範例。

```
f=open("hello_world.txt","r")
words=f.read(5)
print(words)
words2=f.read()
print(words2)
f.close()
```

hello  
world

思考一下執行結果，我們先讀出5個字後印出來，結果正確。

接著我們不指定size，讀出來卻不是完整的hello world? 為什麼呢?







## 第十八節課|檔案：檔案的基本操作

### 第十八節課|檔案|讀取一行一行讀

檔案物件也提供了讀取一行內容的函式：

#### f.readline()

新增些內容到我們的檔案內，然後試試看用readline讀取檔案，看看有沒有不同

```
hello_world.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
hello world
Today is a sunny day
我想去公園玩

f=open("hello_world.txt","r")
words=f.readline()
print(words)
f.close()

hello world
```

為何只有第一行被印出來呢，因為readline就是一次印出一行唷。Readline會去偵測內容裡的換行符號 `\n` 然後就停止。

想看看，要如何利用readline印出所有內容？

檔案物件也提供了一次讀出所有行的函式：

#### f.readlines()

利用這個函式可以把所有行讀出來，同時以清單的方式存到指定的變數內唷。試試看以下的程式範例並執行看看結果：

```
f=open("hello_world.txt","r")
words=f.readlines()
print(words)
f.close()

['hello world\n', 'Today is a sunny day\n', '我想去公園玩\n']
```







# 第十八節課|檔案：檔案的基本操作

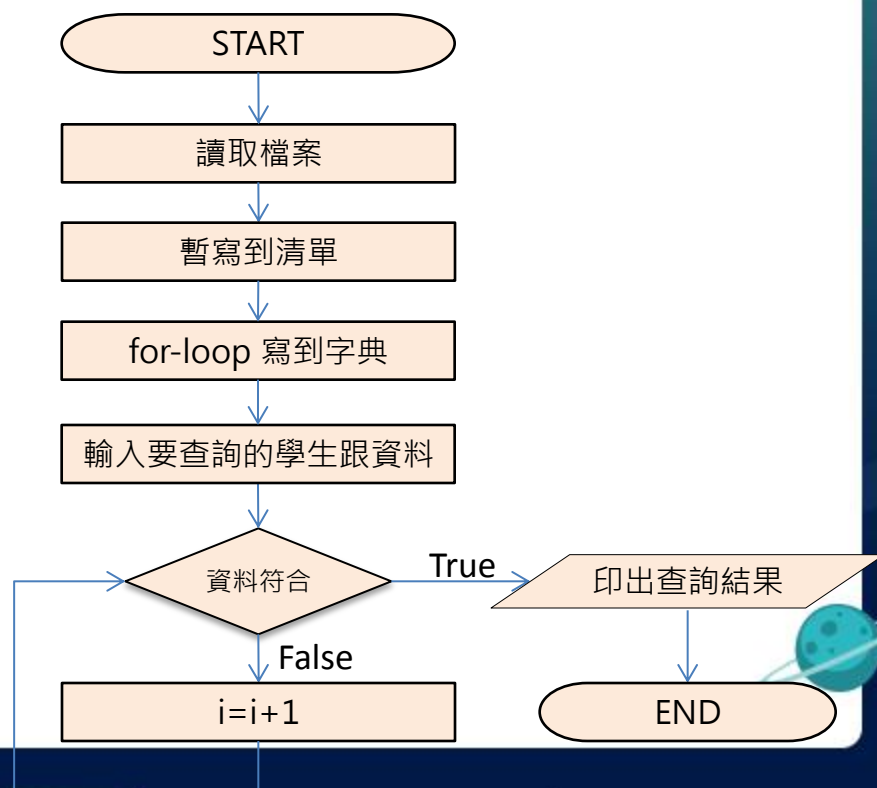
## 第十八節課|檔案|學生查詢系統

還記得上堂課我們做的學生查詢系統嗎？  
在學習到檔案後，讓我們利用學生的資料  
檔案來開發這個學生查詢系統吧。

我們先建立一個文字檔案如以下範例：

```
python_data.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Peter
15
9
Sue
10
4
Kevin
12
6
Ken
9
3
Joe
11
5
Richard
14
8
|
```

接下來，思考一下如何把檔案讀出來，同時放到字典裡呢？先劃出流程圖吧





# 第十八節課|檔案：檔案的基本操作

## 第十八節課|檔案|學生查詢系統

參考以下的程式範例，試試看

```
f=open("python_data.txt","r")
wordslst=f.read().splitlines()
#print(wordslst)
f.close()

student=[]
for i in range(6):
    student.append(wordslst[i*3+0])

print(student) #確認清單裡有所有學生

for i in range(6):
    student[i]={"name":wordslst[i*3+0],
               "age":wordslst[i*3+1],
               "grade":wordslst[i*3+2]}
    #print(student[i])

student_found=False

print("all data is ready for query")
checkname=input("please enter the student's name ")
checkitem=input("please enter the item ")
for j in range(6):
    if(student[j]["name"]==checkname):
        print("The data "+ checkitem + " is: "+student[j][checkitem])
        student_found=True
        break;
if(student_found==False):
    print("Can't find the data")
```





# 第十八節課|檔案：檔案的基本操作

## 第十八節課|檔案|學生查詢系統

執行程式，看看結果是否正確

```
['Peter', 'Sue', 'Kevin', 'Ken', 'Joe', 'Richard']  
all data is ready for query  
please enter the student's name Joe  
please enter the item age  
The data age is: 11
```

```
['Peter', 'Sue', 'Kevin', 'Ken', 'Joe', 'Richard']  
all data is ready for query  
please enter the student's name Lin  
please enter the item age  
Can't find the data
```



# 作品討論與實作

討論主題

利用檔案做出有用有趣的程式吧！

繼續作品實作！



tutorJr | 青少兒程式設計

# Python 課程

## 初級課程19

【檔案操作：修改檔案】



# 初級Python知識點







# 第十九節課|檔案：檔案的修改

## 第十九節課|檔案|開啟與讀取-複習

在上堂課我們學習到檔案的開啟與讀取，我們先來複習一些相關的知識吧：

檔案物件：用來開啟檔案的物件，提供很多函式操作檔案，利用檔案開啟建立物件：

```
f = open(檔名, mode)
```

讀取檔案：將檔案裡的內容讀出來，利用read來讀取內容

```
f.read([size])
```

讀行：也可以利用readline讀出一行內容

```
f.readline()
```

讀行同時忽略換行符號：

```
f.read().splitline()
```

試試以下的程式範例複習檔案的讀取吧

```
f=open("hello_world.txt","r")
words=f.readlines()
print(words)
f.seek(0)
words_new=f.read().splitlines()
print(words_new)
f.close()
```

執行看看有何不同？

```
['hello world\n', 'Today is a sunny day\n', '我想去公園玩\n']
['hello world', 'Today is a sunny day', '我想去公園玩']
```

利用splitline會把換行符號 `\n` 給忽略掉，但是readlines會把換行符號當成內容讀出來唷



# 第十九節課|檔案：檔案的修改

## 第十九節課|檔案|寫入檔案

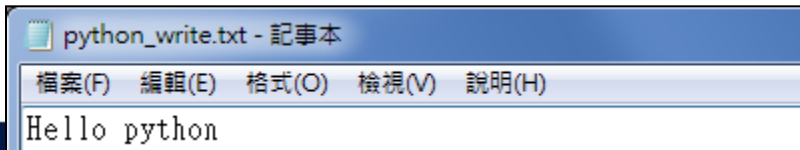
我們知道如何開啟檔案跟讀取檔案後，接下來學習如何把資料寫到檔案內，最基本的方式就是利用檔案物件提供的write：

`f.write("資料")`

跟讀取檔案相同，需要先開啟檔案後才能進行資料寫入，同時檔案開啟的模式要設置為“w”。試試以下的程式範例，試著建立一個檔案同時寫入一些資料

```
f=open("python_write.txt","w")
f.write("Hello python")
f.close()
```

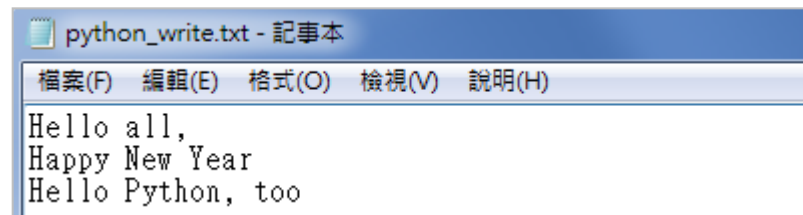
看看執行結果，要到跟程式同目錄下找這個檔案開啟看看



接著我們寫入更多資料，也試著加入換行符號，看看以下的程式範例

```
f=open("python_write.txt","w")
f.write("Hello all,\nHappy New Year\nHello Python, too")
f.close()
```

執行看看有何不同？



同樣的檔案，為何之前寫的資料不見了呢？

### 程式小提醒

“w”：寫入模式(覆寫)，利用這個模式開啟檔案可以進行內容的寫入且覆寫原本的檔案，如果檔案不存在就會建立新的檔案

# 第十九節課|檔案：檔案的修改

## 第十九節課|檔案|寫入檔案

讀取檔案內容有提供一次讀取多行，同樣在寫入資料時檔案物件也提供了一次寫入多行的函式：

`f.writelines(data)`

這裡的data指的是像list或是tuple之類的資料型態，透過f.writelines()會把data裡的資料全部寫入到檔案內，試試以下的程式範例：

```
f=open("python_write.txt","w")
data=["Hello eveyone, \nToday is a good day\n"]
f.writelines(data)
f.close()
```

看看執行結果，要到跟程式同目錄下找這個檔案開啟看看

```
python_write.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Hello eveyone,
Today is a good day
```

還有另一種把資料寫入檔案的方式，就是我們熟悉的print()，可以利用print把資料印到檔案內唷。

`Print(data, file=f)`

試試以下的程式範例並執行看看

```
f=open("python_file_print.txt","w")
student_data=['Peter', '15', '9',
              'Sue', '10', '4',
              'Kevin', '12', '6',
              'Ken', '9', '3',
              'Joe', '11', '5',
              'Richard', '14', '8',]
print(student_data, file=f)
f.close()
```

```
python_file_print.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
['Peter', '15', '9', 'Sue', '10', '4', 'Kevin', '12', '6', 'Ken', '9', '3', 'Joe', '11', '5', 'Richard', '14', '8']
```

# 第十九節課|檔案：檔案的修改

## 第十九節課|檔案|檔案內容裡的位置

還記得上堂課裡的一個範例問題嗎？當我們讀取檔案內容後再次讀取，發現讀不到資料了，還記得原因嗎？看看當時的範例

```
f=open("hello_world.txt","r")
words=f.read(5)
print(words)
words2=f.read()
print(words2)
f.close()
```

```
hello
world
```

第二次讀取內容時沒有指定size，應該要從頭把內容都印出來，但是卻只有印出後面的資料，原因是在處理檔案的時候，有個指標在指著內容的位置，每次處理內容時都會從這個指標開始唷。

看看以下的示意圖

↓  
h e l l o w o r l d

執行f.read(5)之後，指標移動到空格的位置

↓  
h e l l o w o r l d

要如何可以從頭印呢？檔案物件提供了找尋位置的功能：

**f.seek(location)**

試著修改剛剛的程式，參考以下範例

```
f=open("hello_world.txt","r")
words=f.read(5)
print(words)
f.seek(0)
words2=f.read()
print(words2)
f.close()
```

```
hello
hello world
```

## 第十九節課|檔案：檔案的修改

### 第十九節課|檔案|學生查詢系統-新增

還記得上堂課我們做的學生查詢系統嗎？讓我們把這個系統加上新的功能，可以新增學生資料進去吧。

先思考這個新增功能要如何做？

1. 詢問是否要新增資料
2. 輸入學生姓名
3. 輸入學生年齡
4. 輸入學生年級
5. 輸出結果

接下來，思考一下如何把資料寫進去檔案內呢？先劃出流程圖吧





## 第十九節課|檔案：檔案的修改

### 第十九節課|檔案|學生查詢系統-新增

參考以下的程式範例，試試看

```
need_modify=int(input(print("如果需要增加學生，請按1；不需要增加，請按0結束")))
if(need_modify==1):
    f=open("python_data.txt","a")
    f.write(input("請輸入學生姓名"))
    f.write("\n")
    f.write(input("請輸入學生年齡"))
    f.write("\n")
    f.write(input("請輸入學生年級"))
    f.write("\n")
f.close()
f=open("python_data.txt","r")
wordslis=f.read().splitlines()
print(wordslis)
f.close()
```

如果需要增加學生，請按1；不需要增加，請按0結束

None1

請輸入學生姓名Penny

請輸入學生年齡12

請輸入學生年級6

['Peter', '15', '9', 'Sue', '10', '4', 'Kevin', '12', '6', 'Ken', '9', '3', 'Joe', '11', '5', 'Richard', '14', '8', 'Penny', '12', '6']





# 作品討論與實作

討論主題

利用檔案做出有用有趣的程式吧！

繼續作品實作！



tutorJr | 青少兒程式設計

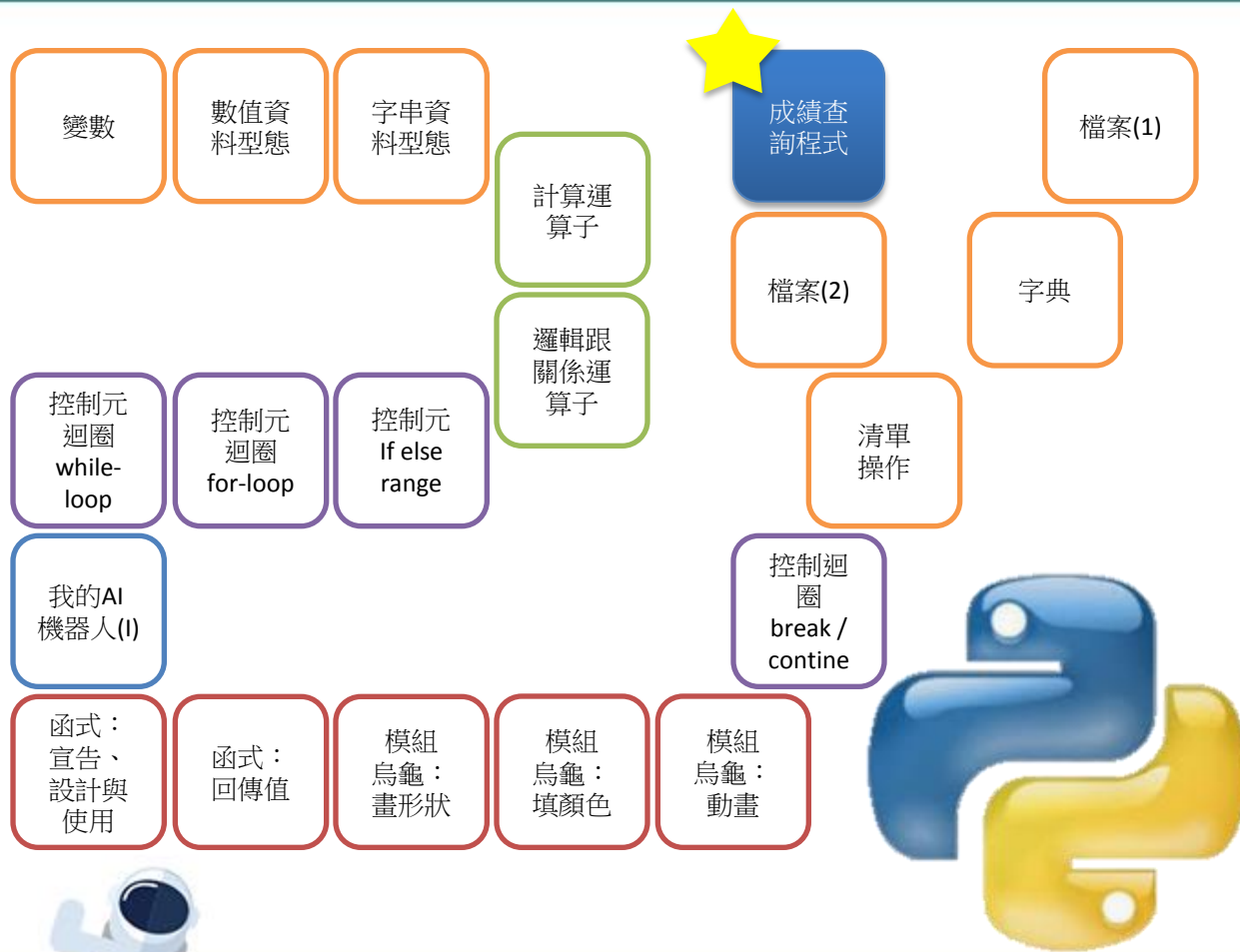
# Python 課程

## 初級課程20

【綜合練習：班級成績查詢程式】



# 初級Python知識點





## 第二十節課|綜合練習：成績查詢程式

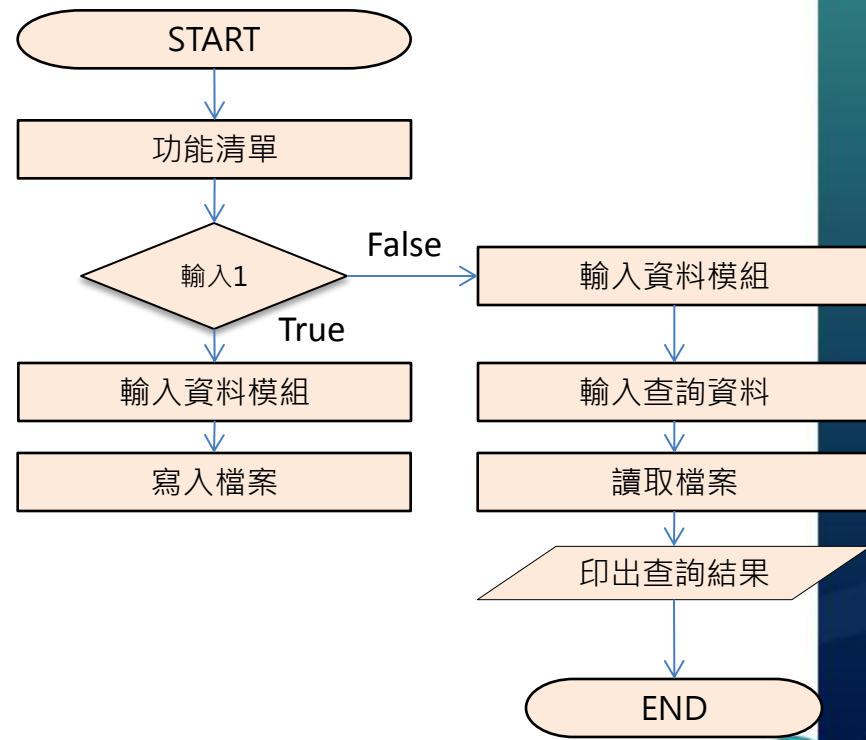
### 第二十節課|綜合練習|成績查詢程式

最後一堂Python基礎課程，讓我們利用所學的程序，來完成一個成績查詢系統吧。

首先我們先定義好這個成績查詢系統會有那些功能

1. 功能清單：輸入資料(1)跟查詢資料(2)
2. 輸入學生資料：姓名、班級跟座號。
3. 輸入學生成績：數學、國語跟自然。
4. 資料寫入檔案：score.txt
5. 輸入要查詢的學生姓名
6. 印出查詢的成績資料

我們會應用到學過的各種程式知識，首先，先試著劃出模組的流程圖吧。想看看，會需要那些模組呢？





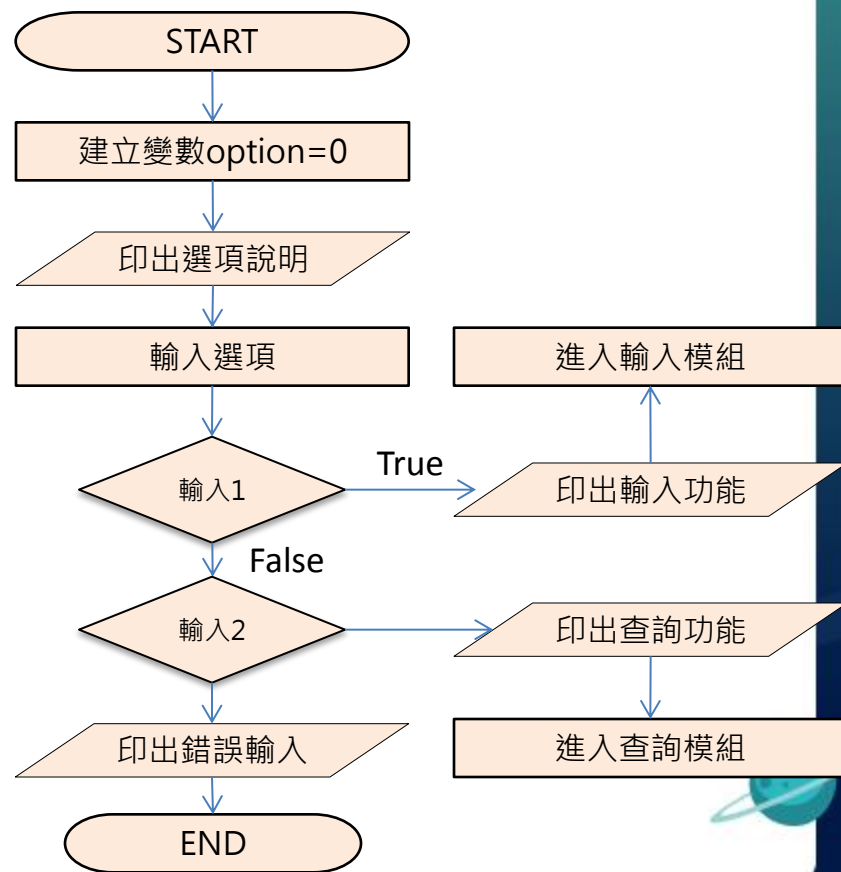
## 第二十節課|綜合練習：成績查詢程式

### 第二十節課|綜合練習|功能清單建立

功能清單會應用到的程式知識包含：變數、輸入、字串處理、if/elif 控制。我們利用變數來記錄功能，利用輸入跟字串來顯示介面跟進行輸入，記得資料形態要轉換唷。最後利用if控制判斷輸入的結果。流程圖很簡單試試看：

#### 程式小提醒

Input輸入後的資料型態預設是文字唷





## 第二十節課|綜合練習：成績查詢程式

### 第二十節課|綜合練習|功能清單建立

接著試著寫出功能清單程式，參考以下的程式範例：

```
option=0
print("學生查詢系統")
print("請選擇功能")
print("輸入學生成績請按 1 ")
print("查詢學生成績請按 2 \n請輸入")
option=int(input())
if(option==1):
    print("開始輸入學生成績")
elif(option==2):
    print("開始查詢學生成績")
else:
    print("你輸入的功能編號有誤")
```

測試結果，記得每種狀況都要測試，確定都能正常執行。

```

學生查詢系統
請選擇功能
輸入學生成績請按 1
查詢學生成績請按 2
請輸入
```

```

請輸入
1
開始輸入學生成績
```

```

請輸入
5
你輸入的功能編號有誤
```

```

請輸入
2
開始查詢學生成績
```



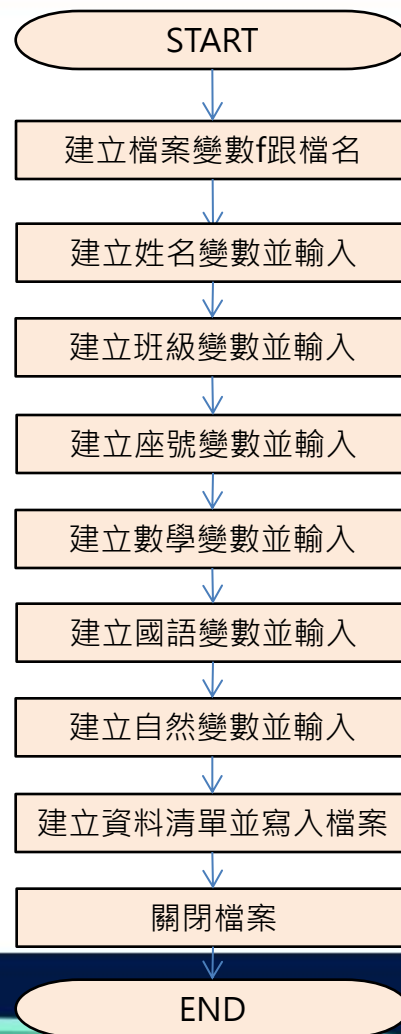




## 第二十節課|綜合練習：成績查詢程式

### 第二十節課|綜合練習|輸入成績資料

接著，要來做輸入學生資料的模組囉！我們先把功能做出來，要輸入的資料包含：姓名、班級、座號、數學、國語跟自然成績。這裡會需要用到變數、清單、輸入跟檔案，先把流程圖畫出來吧。





## 第二十節課|綜合練習：成績查詢程式

### 第二十節課|綜合練習|輸入成績資料

接著試著寫出功能清單程式，參考以下的程式範例：

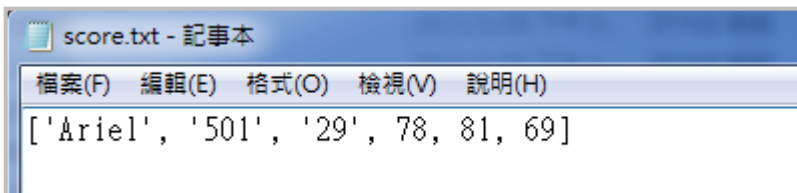
```
f=open("score.txt","w")
print("請輸入學生姓名:")
name=input()
print("請輸入學生班級:")
class_no=input()
print("請輸入學生座號:")
seat_no=input()
print("請輸入學生數學成績:")
math_score=int(input())
print("請輸入學生國語成績:")
chinese_score=int(input())
print("請輸入學生自然成績:")
science_score=int(input())

student_data=[name,class_no,seat_no,math_score,
               chinese_score,science_score]

print(student_data, file=f)
f.close()
```

測試結果，同時開啟檔案確認資料是否都有寫進去。

```
請輸入學生姓名:
Ariel
請輸入學生班級:
501
請輸入學生座號:
29
請輸入學生數學成績:
78
請輸入學生國語成績:
81
請輸入學生自然成績:
69
```





## 第二十節課|綜合練習：成績查詢程式

### 第二十節課|綜合練習|輸入成績資料

完成輸入成績資料的功能後，我們把這個功能做成模組使用吧

利用def定義模組後，將程式碼搬到模組下，然後在清單選擇輸入後執行輸入模組。試試看，參考以下的程式範例

#### 程式小提醒

檔案開啟時的模式要注意，如果要從後面接續寫檔案的話要使用a



```
def score_input():
    f=open("score.txt","a")
    print("請輸入學生姓名:")
    name=input()
    print("請輸入學生班級:")
    class_no=input()
    print("請輸入學生座號:")
    seat_no=input()
    print("請輸入學生數學成績:")
    math_score=int(input())
    print("請輸入學生國語成績:")
    chinese_score=int(input())
    print("請輸入學生自然成績:")
    science_score=int(input())
    student_data=[name,class_no,seat_no,math_score,
                  chinese_score,science_score]
    print(student_data, file=f)
    f.close()

option=0
print("學生查詢系統")
print("請選擇功能")
print("輸入學生成績請按 1 ")
print("查詢學生成績請按 2 \n請輸入")
option=int(input())
if(option==1):
    print("開始輸入學生成績")
    score_input()
elif(option==2):
    print("開始查詢學生成績")
else:
    print("你輸入的功能編號有誤")
```

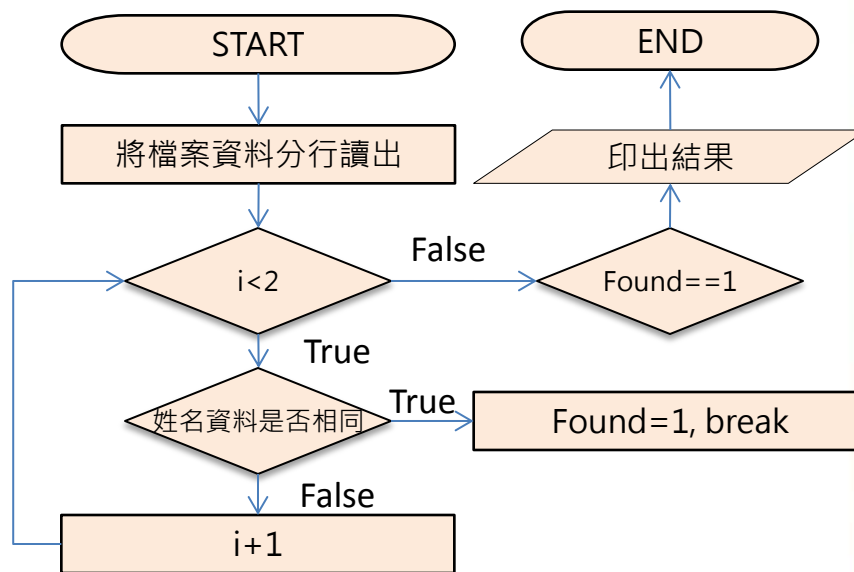


## 第二十節課|綜合練習：成績查詢程式

### 第二十節課|綜合練習|查詢成績資料

完成學生成績資料的輸入功能後，我們來做查詢的功能吧。查詢功能提供輸入學生姓名後，比對是否存在，然後印出學生的成績的功能。

我們先劃出流程圖，使用基礎的運算子去比對資料在清單裡的位置，以及資料是否相同。





## 第二十節課|綜合練習：成績查詢程式

### 第二十節課|綜合練習|查詢成績資料

接著試著寫出功能清單程式，參考以下的程式範例：

```
def score_query():  
    found=0;  
    f=open("score.txt","r")  
    students_data=f.read().splitlines()  
    query_name=input("請輸入要查詢的學生姓名:")  
    for i in range(3):  
        if(students_data[(i*6)]==query_name):  
            found=1;  
            break;  
    if(found==1):  
        print("你要查詢的學生成績資料如下:")  
        print("學生姓名："+query_name)  
        print("學生班級："+students_data[(i*6)+1])  
        print("學生座號："+students_data[(i*6)+2])  
        print("數學分數："+students_data[(i*6)+3])  
        print("國語分數："+students_data[(i*6)+4])  
        print("自然分數："+students_data[(i*6)+5])  
    else:  
        print("not found")  
    #print(students_data)  
    f.close()
```

測試結果，完成查詢系統囉

```
學生查詢系統  
請選擇功能  
輸入學生成績請按 1  
查詢學生成績請按 2  
請輸入  
2  
開始查詢學生成績  
請輸入要查詢的學生姓名:Ariel  
你要查詢的學生成績資料如下:  
學生姓名:Ariel  
學生班級:501  
學生座號:77  
數學分數:89  
國語分數:86  
自然分數:47
```



# 作品討論與實作

## 討論主題

要如何加上修改成績功能呢？

繼續作品實作！

