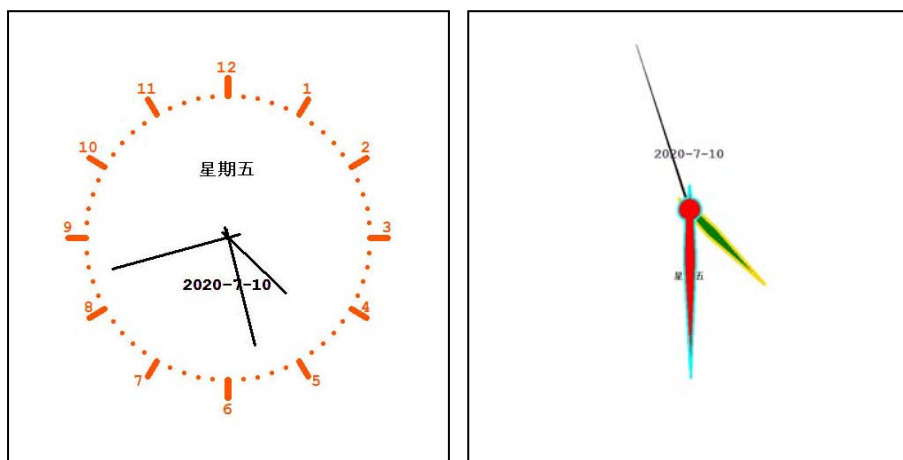


Python3 續學習與利用

一、前言：

上回初次接觸 Python 程式後，先去網上找入門教學並把基本觀念及程式碼資訊做點整理，將整理後資訊列在第 756 期電子報中，以利後續撰寫程式時較容易回顧參考。之後繼續去研究內建繪圖及其他運算之基本函式，也去網上找他人公開提供繪動態時鐘及 2D 俄羅斯方塊案例來參考，但後者太長對初學者尚有難度，就先跳回過來著手去試撰幾個用得到的小程式。以前曾用 VB 寫過簡單測量相關程式，包括繪出成果圖來，所以對 Python 之利用仍由簡易測量程式起步來做練習。



<https://kknews.cc/code/3brovng.html> 左圖是動態時鐘程式來源，右係自己改指針
<https://kknews.cc/code/jk4ja9l.html> 網路上有 2D 俄羅斯方塊程式位址。

二、此階段心得分享：

1. 內建模組

python 所提供的模組有很多，繪圖是一個比較想去先玩的，所以由海龜繪圖模組切入找到這個網上資訊 <http://yltang.net/tutorial/python/5/>



上述畫面係網上唐元亮君中文版海龜繪圖模組說明，雖是簡單說明，但初步繪圖觀念已經有了，只要去瀏覽一下，已經可以先去試繪簡易的圖了。

文章內容中在標題 (2) for 迴圈前有 **Tk 顏色名稱與數值** 之連結

<http://www.tcl.tk/man/tcl8.4/TkCmd/colors.htm>

原來是在繪圖時要標註顏色可以用 RGB 三個編號來設定，問題在於第一欄是寫顏色英文名字 接續是 RGB 編號，不熟悉這些名稱可能不知道顏色名稱跟顏色如何對應。所以就先去建一個 xls 檔，讓顏色跟顏色名稱對應起來，如此日後要挑選顏色就可以先看到是那種顏色了。其資訊總共有 753 筆，但重複或大小寫都有的非常多。xls 檔中也嘗試把中文顏色名稱寫上去，因為是利用 Google 翻譯直接翻，可能會有正確性問題，僅供參考，檔案放附錄 1 中，也許有人可以利用到。

1	Color Name	顏色名稱	Color	R	G	B	Remark
24	bisque3	濃湯3		205	183	158	
25	bisque4	濃湯4		139	125	107	
26	black	黑色		0	0	0	
27	blanched almond	杏仁白		255	235	205	
28	BlanchedAlmond	杏仁白		255	235	205	
29	blue	藍色		0	0	255	
30	blue violet	紫羅蘭色		138	43	226	
31	blue1	藍色1		0	0	255	
32	blue2	藍色2		0	0	238	
33	blue3	藍色3		0	0	205	
34	blue4	藍色4		0	0	139	
35	BlueViolet	紫羅蘭色		138	43	226	
36	brown	棕色		165	42	42	
37	brown1	棕色1		255	64	64	
38	brown2	棕色2		238	59	59	
39	brown3	棕色3		205	51	51	
40	brown4	棕色4		139	35	35	
41	burlywood	布萊塢		222	184	135	
42	burlywood1	布萊塢1		255	211	155	
43	burlywood2	布萊塢2		238	197	145	
44	burlywood3	布萊塢3		205	170	125	
45	burlywood4	布萊塢4		139	115	85	
46	cadet blue	學員藍色		95	158	160	
47	CadetBlue	學員藍色1		95	158	160	
48	CadetBlue1	學員藍色2		152	245	255	
49	CadetBlue2	學員藍色3		142	229	238	
50	CadetBlue3	學員藍色4		122	197	205	
51	CadetBlue4	學員藍色5		83	134	139	
52	chartreuse	黃綠色		127	255	0	
53	chartreuse1	黃綠色1		127	255	0	
54	chartreuse2	黃綠色2		118	238	0	

海龜繪圖模組 文章最後，有一行

[完整資料請參考； <https://docs.python.org/3/library/turtle.html>]

點進去才發現是原始 英文版的完整 Turtle graphics 說明文件。你還可以去轉換為傳統或簡體中文版，只是傳統版仍出現一堆簡體字就是。目前亦僅就需要用到的指令去了解測試，還有許多看不懂的，仍需時日去研究。

Turtle 繪圖模組最單純的運作方式，大概就是先載入模組 **import turtle**，再建立一個自訂大小的螢幕，當然可以設定螢幕底色，螢幕上座標預設以正中為原點。再設定繪筆 `turtle.Turtle()`，也就是烏龜，預設位置在原點，方向朝

右。繪筆形狀顏色也有多種選項。用移動繪筆(烏龜)的多種模式，來繪圖，當然線條寬度顏色或暫時隱藏不顯現，都隨你設定，甚至烏龜移動速度都可以任意設定，來顯示動畫效果。用烏龜繪圖，沒看到有分圖層，但可以用好幾隻烏龜來繪圖，可獨立作業來繪圖或刪除不互相干擾，跟圖層功能有點類似。看唐君中文版海龜繪圖模組簡例，最後有一行指令 `screen.exitonclick()` 是最後用滑鼠點一下螢幕來跳出的指令，沒加這一行，繪圖螢幕不會出現。

一般簡易繪圖會用到的指令只要去看 p1 所連結唐君那分資料或英文版完整說明，很快就可以上手，在此就不贅述。至於其他較細節或進階的指令，有興趣的朋友可以就本身繪圖所需部分去深入研究。

除了繪圖，最需要用到的模組應該就是數學模組了

<https://docs.python.org/3/library/math.html>

要使用數學模組，一樣是要在程式中先 `import math`，上述連結中就有完整的模組使用說明。首先最常用的是三角函數、對數、開方等工程數相學關的指令，跟 xls 上差不多，但記得指令都要加上 `math`

```
math.cos(x)          math.acos(x)
math.sin(x)          math.asin(x)
math.tan(x)          math.atan(x)
math.atan2(y,x)
math.degrees(x)      math.radians(x)
math.pi              math.e
math.log10(x)        math.log(x,[base])
math.exp(x)
math.sqrt(x)
```

上面各指令應該都可以一目了然，無需多做說明。只是要注意三角函數用到到角度都是徑度量單位不是用角度單位。其中反函數 `arctan` 有兩種指令，`math.atan(x)` 結果範圍是定在 $\pi \sim -\pi$ 間，但較精確表示的 `math.atan2(y,x)` 則是會依 參數 y,x 來呈現其所在象限。

只大致看了這兩個模組，也就開始去試撰本身想做測試的程式了，一面看上述網上的指令說明，一面在嘗試錯誤的模式下，也寫了幾個東西。

2. 試做經驗

a. 時鐘

第一個時鐘其實是直接由網上案例抄下來，只是他用的文件縮排格式不完全正確，所以還需要去一一檢討調整縮排格式。整理出的 py 檔去執行時又發現程式碼有一些小 bug，鐘面上外框的時鐘數字位置亂掉，分成幾個部分圓，不是同一個鐘面。一一去檢討後總算可以正確執行，時鐘動態畫面截圖如 p1 之左圖，整理過之程式碼放在後面附錄 4。對該檔案內容去推敲研究也學到其實各指針就各是一隻繪圖的烏龜，用各烏龜的指向控制，來顯示秒針、分針與時針的指向，也得到如何抓取當下時間控制指針方向的寫法。由

於該程式中指針全部都只用一小條線段表現，但既然可以製作一個圖來當繪圖之烏龜，所以就試繪時針跟分針造型，秒針則仍用一段直線，時鐘外框數字先忽略，顯示外觀如 p1 右圖。時針分針係用圖樣各折點座標來繪，為方便計先在 AutoCad 上繪形狀，找出座標，再回來繪製，用同一型狀利用長寬係數不同來分別產生長短胖瘦不一的時分針。此局部程式碼亦放附錄 5。

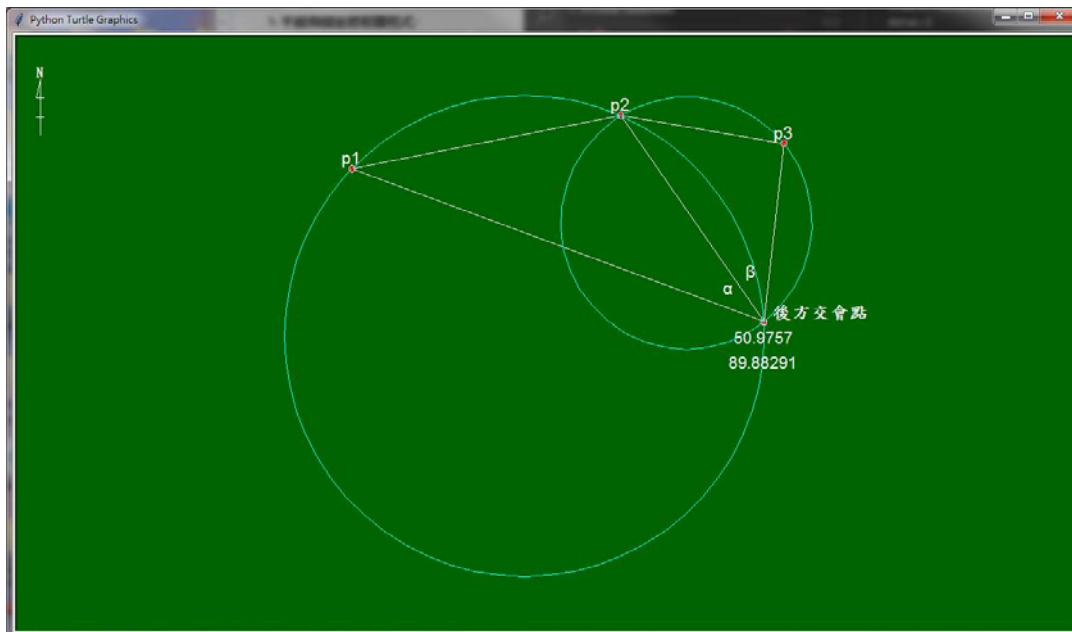
b. 平面曲線座標相關程式

以前就曾用 VB 寫過類似程式，利用原計算模式來練習改寫為 python 程式碼。沒去利用現有模組功能，所有需要的副程式都自己寫較單純，包括：

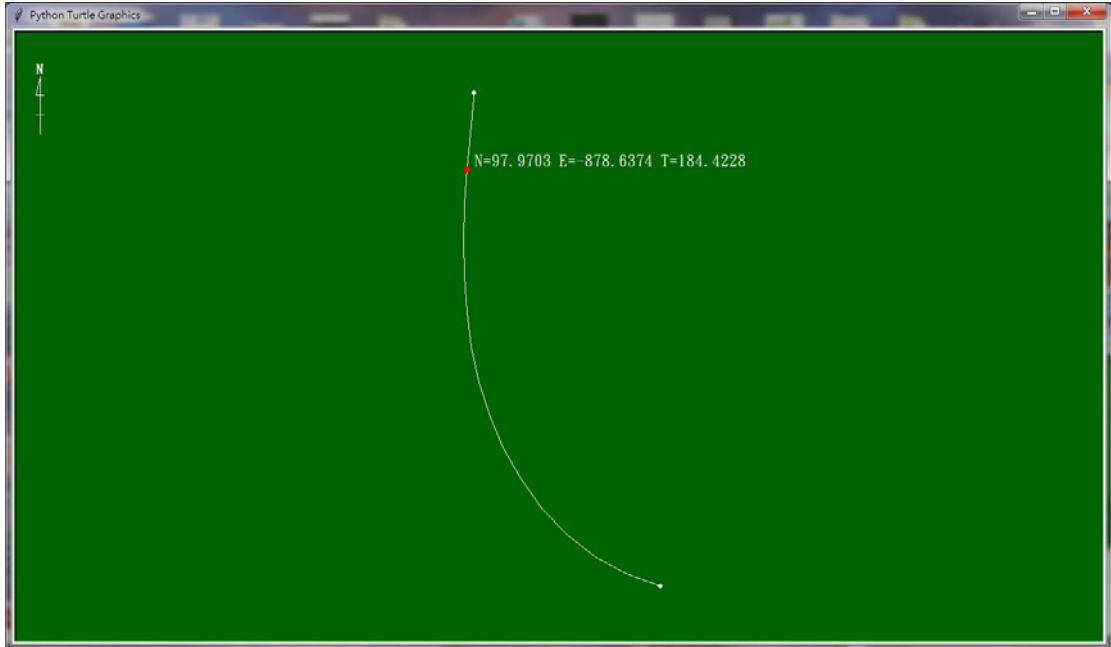
p00(n2,e2,n1,e1)	計算兩點座標距離與方位
p01(n,e,d,t)	由第一點座標距離與方位計算第二點座標
p02(n,e,t,a,l,er)	計算克羅梭曲線座標
p03(n,e,t,r,l)	計算圓曲線座標
p04(n1,e1,q1,n2,e2,q2)	計算前方交會點座標
p05(n1,e1,n2,e2,n3,e3,a,b)	計算後方交會點座標
p20(n1,e1,t1,r1,sta1,a,er,n2,e2,t2,r2,sta2)	推算蛋型曲線起點
p21(n,e,t,a,l,er,na,ea,er1)	求任一點與克羅梭曲線之關係
p22(n,e,t,a,l,er,na,ea,ta,er1)	計算直線與克羅梭曲線交點
p31(n,e,t,r,na,ea)	計算任何一點與圓曲線之關係
p32(n,e,t,r,na,ea,ta)	求直線與圓曲線交點座標

此模組中還附有一些必備的小副程式就不贅述。各副程式後幾列已把他加 # 成綠色字當成標註列之資訊，則是該程式編寫時當測試資料用的資訊。此一做到目前之平面曲線副程式模組，檔名 survey.py 以檔案格式置附錄 2 中供連結參考，有興趣的朋友可以參考利用，發現有 bug 歡迎提供寶貴意見。

現在測量儀器已相當先進，所以這些資訊實用性不高了，主要目的是用來學習試寫 py 程式碼罷了。例如後方交會也是古早以前的模式，但在輸入資料、計算及繪圖細節上還是花了一點時間去揣摩。



要利用附錄中之副程式，仍需自己去寫主程式。要先掛載常用 `sys`、`pandas`、`turtle` 等模組，上述副程式放在已設好路徑的檔案夾中。要使用的副程式參數，可以直接在主程式中寫入，或寫些指令來由螢幕上輸入。若要繪圖出來，則要先設定繪圖螢幕，並將要繪圖資訊來利用副程式跟螢幕大小比對設定縮放比例才繪上去。克羅梭曲線 計算之簡易主程式例 如附錄 3。



三、結語：

1. 撰寫程式有無限的空間去開拓，雖然如今網路上資源已經非常充沛，但要找到完整的學習資訊，可能還是要就所需資訊一處處去尋找。覺得終究要一步一腳印，直接針對本身需要寫程式的目標，下去撰寫去嘗試，經驗應該就可以慢慢累積。
2. 這期間也沒很積極去撰寫去嘗試，僅就走過的經驗與已經撰寫的東西分享，讓有興趣的朋友多一點參考案例。本篇非以教學方式來敘述，僅提供現有程式碼供參考。本身還是初學者，所以內容不會是最佳寫法，但仍可得到所需資訊。也就是僅就所需要的有限指令去稍做了解，已經可以利用 python 來計算繪製所需資訊。
3. 目前關於平面曲線還只寫到各類曲線之副程式，但一條道路含有各類曲線，鍵入各曲線起點樁號及相關資訊，計算出整條路線各整樁座標方位才有實用價值，是後續要做的。

四、附錄：

1. 顏色與 RGB 對照檔案 `first.xlsm`
2. `survey.py` 平面曲線相關簡易程式模組
3. `vbspiral.py` 克羅梭曲線 簡易主程式
4. `clock.py` 程式碼 文字中 `。`是空白位置

```

*****
# coding=utf-8
import turtle
from datetime import *

# 抬起畫筆，向前運動一段距離放下
def Skip(step):
    turtle.penup()
    turtle.forward(step)
    turtle.pendown()
def mkHand(name, length):
    # 註冊 Turtle 形狀，建立錶針 Turtle
    turtle.reset()
    Skip(-length * 0.1)
    # 開始記錄多邊形的頂點。當前的烏龜位置是多邊形的第一個頂點。
    turtle.begin_poly()
    turtle.forward(length * 1.1)
    # 停止記錄多邊形的頂點。當前的烏龜位置是多邊形的最後一個頂點。將與第一個頂
    # 點相連。
    turtle.end_poly()
    # 返回最後記錄的多邊形。
    handForm = turtle.get_poly()
    turtle.register_shape(name, handForm)

def Init():
    global secHand, minHand, hurHand, printer
    # 重置 Turtle 指向北
    turtle.mode("logo")
    # 建立三個錶針 Turtle 並初始化
    mkHand("secHand", 135)
    mkHand("minHand", 125)
    mkHand("hurHand", 90)
    secHand = turtle.Turtle()
    secHand.shape("secHand")
    minHand = turtle.Turtle()
    minHand.shape("minHand")
    hurHand = turtle.Turtle()
    hurHand.shape("hurHand")

```

```

for hand in secHand, minHand, hurHand:
    hand.shapesize(1, 1, 3)
    hand.speed(0)
    # 建立輸出文字 Turtle
    printer = turtle.Turtle()
    # 隱藏畫筆的 turtle 形狀
    printer.hideturtle()
    printer.penup()

def SetupClock(radius):
    # 建立表的外框
    turtle.reset()
    turtle.pensize(7)
    turtle.pencolor("#ff5500")
    turtle.fillcolor("green")

    turtle.shape("blank")
    #turtle.shapesize(0.3,0.6)

    for i in range(60):
        Skip(radius)
        if i % 5 == 0:
            turtle.forward(20)

            Skip(-radius - 20)
            Skip(radius + 20)

            if i == 0:
                turtle.write(int(12), align="center", font=("Courier",
14, "bold"))

            Skip(-radius - 20)
            turtle.right(6)

        elif i == 30:
            Skip(25)

```

```

        turtle.write(int(i / 5), align="center", font=("Courier
", 14, "bold"))
        Skip(-25)

        Skip(-radius-20)
        turtle.right(6)

    elif (i == 25 or i == 35):
        Skip(20)
        turtle.write(int(i / 5), align="center", font=("Courier
", 14, "bold"))
        Skip(-20)

        Skip(-radius-20)
        turtle.right(6)
        turtle.stamp()

    else:
        turtle.write(int(i / 5), align="center", font=("Courier
", 14, "bold"))
        Skip(-radius - 20)

        turtle.right(6)

    else:
        turtle.dot(5)
        Skip(-radius)
        turtle.right(6)

def Week(t):
    week = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星
期日"]
    return week[t.weekday()]

def Date(t):
    y = t.year
    m = t.month
    d = t.day

```



```

return "%s-%d-%d" % (y, m, d)

def Tick():
    # 繪製錶針的動態顯示
    t = datetime.today()
    second = t.second + t.microsecond * 0.000001
    minute = t.minute + second / 60.0
    hour = t.hour + minute / 60.0
    secHand.setheading(6 * second)
    minHand.setheading(6 * minute)
    hurHand.setheading(30 * hour)

    #turtle.tracer(False)
    printer.forward(65)
    printer.write(Week(t), align="center", font=("Courier", 14, "bold"))
)
    printer.back(130)
    printer.write(Date(t), align="center", font=("Courier", 14, "bold"))
)
    printer.home()
    #turtle.tracer(True)

    # 100ms 後繼續調用 tick
    turtle.ontimer(Tick, 100)

def main():
    # 打開/關閉龜動畫，並為更新圖紙設置延遲。
    turtle.tracer(False)
    Init()
    SetupClock(160)
    turtle.tracer(True)
    Tick()
    turtle.mainloop()

if __name__ == "__main__":
    main()

```

原文網址：<https://kknews.cc/code/3brovng.html>

5.繪製時針分針秒針自撰部分之程式碼

```
def Hand():
    turtle.hideturtle()
    turtle.tracer(False)
    # turtle.speed(1)
    d=125.0608
    turtle.goto(0,-1*d)
    turtle.begin_poly()
    #turtle.setheading(97)    # 原 mode="normal"
    turtle.setheading(-7)
    turtle.fd(66.0967)
    #turtle.setheading(172.28460)
    turtle.setheading(277.7154)
    turtle.circle(-60,150.5706)
    turtle.goto(-30.71111,250.1216)
    turtle.goto(0,1000-d)
    turtle.goto(30.71111,250.1216)
    turtle.goto(22.1998,180.8028-d)
    #turtle.setheading(338.2846)
    turtle.setheading(111.7154)
    turtle.circle(-60,150.5706)
    turtle.goto(0,-1*d)
    turtle.end_poly()
    handForm = turtle.get_poly()
    turtle.register_shape("hand",handForm)    # hadForm 為 poly 名字
    turtle.tracer(True)
    turtle.clear()
    hurHand.shape("hand")
    hurHand.shapesize(0.4,0.25,6)
    hurHand.color("gold","green")
    minHand.shape("hand")
    minHand.shapesize(0.4,0.4,3.5)
    minHand.color("cyan","red")
    turtle.register_shape("myfab",((-1,1.5),(0,20),(1,1.5)))
    secHand.shape("myfab")
    secHand.shapesize(2,18)
```