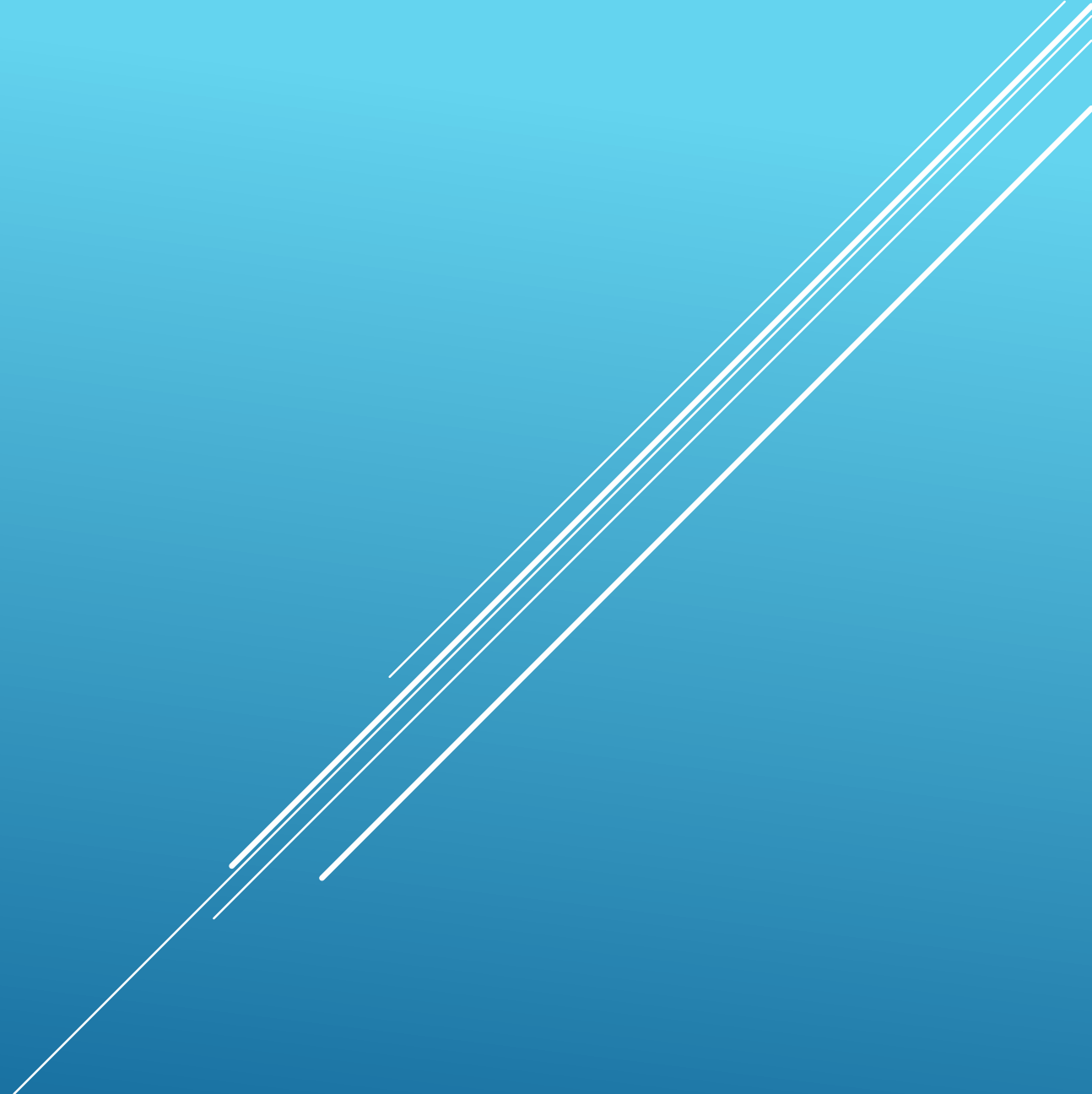


5

PYTHON繪圖

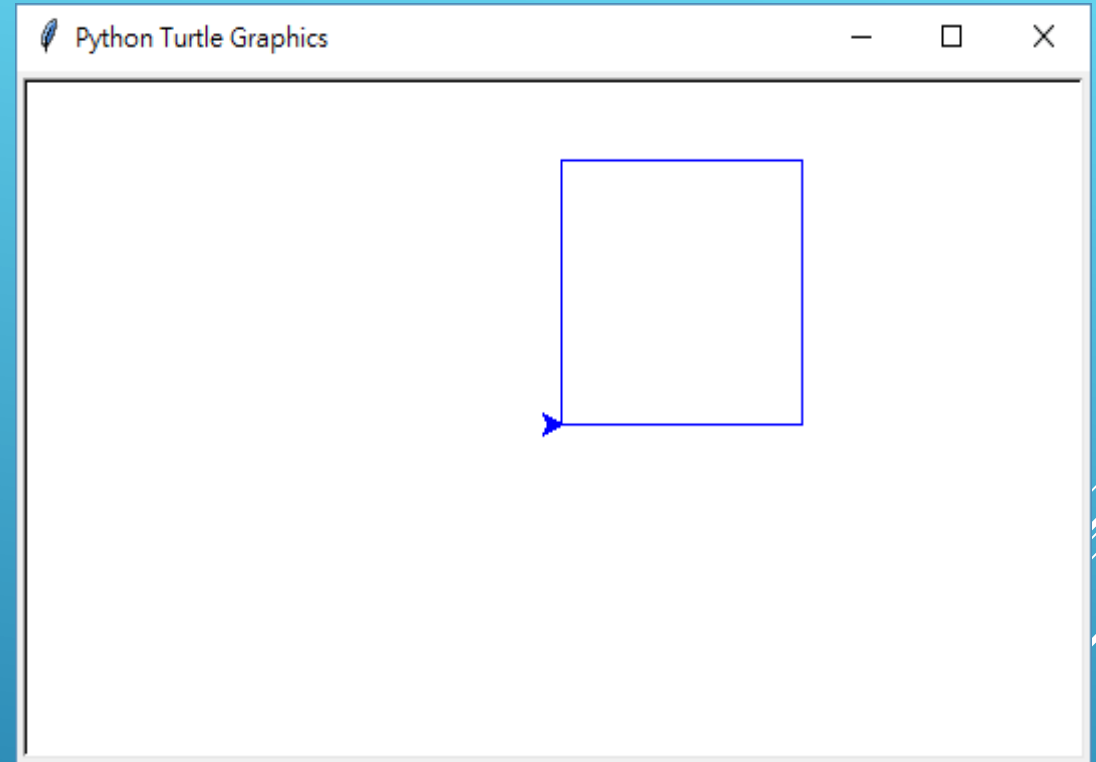
資訊社會必修的
12堂Python通識課



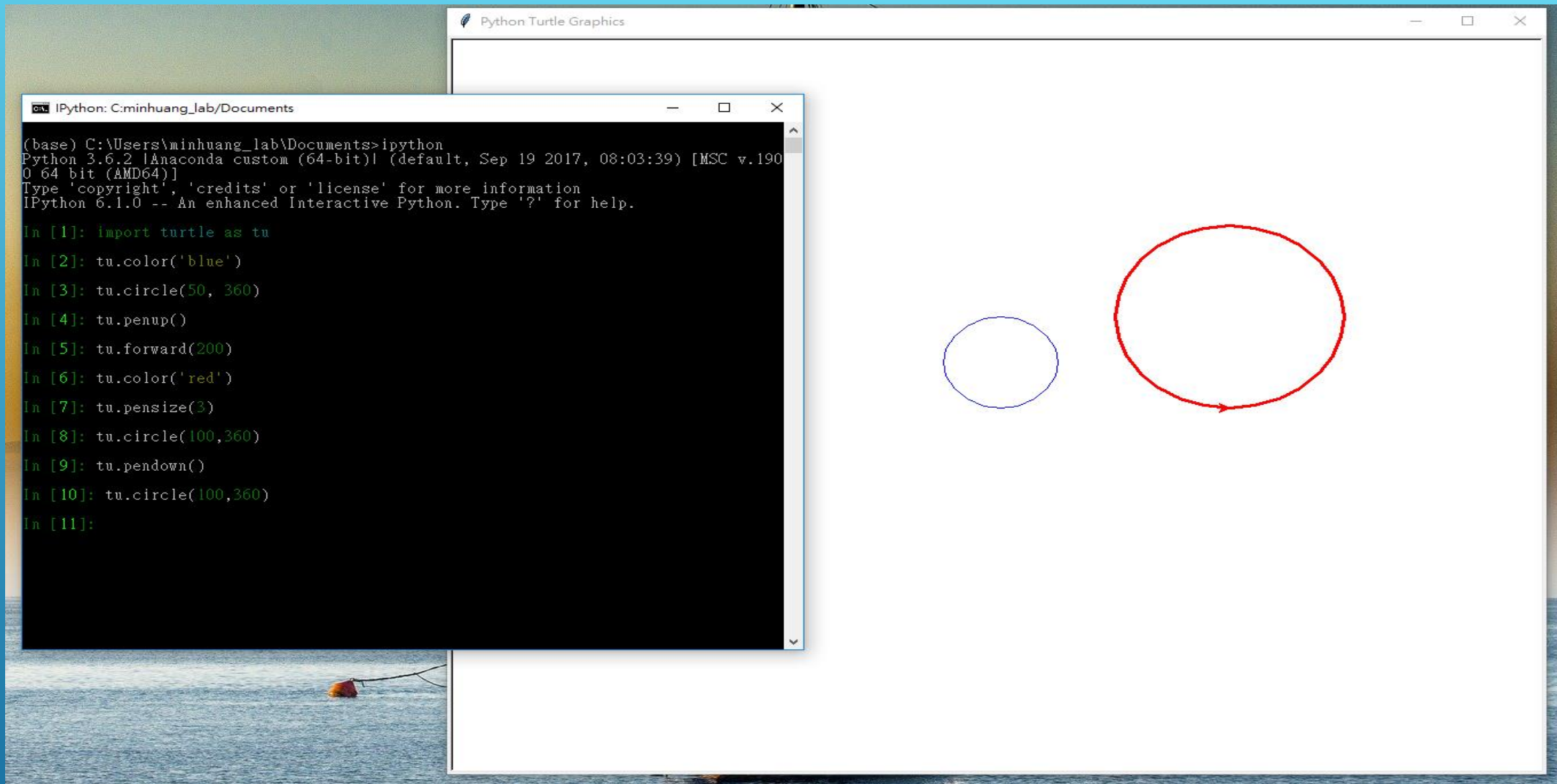
本堂課重點

- ▶ Python海龜繪圖簡介
- ▶ 繪製幾何圖形
- ▶ 繪製數學函數圖形
- ▶ 使用pygame繪製圖形

```
import turtle as tu
tu.color('blue')
for i in range(4):
    tu.forward(120)
    tu.left(90)
tu.done()
```



PYTHON海龜繪圖基本格式



利用交談介面特性來繪圖

函數或常數	說明
<code>bgcolor()</code>	設定視窗的背景顏色
<code>bgpic(picname)</code>	使用圖形檔作為視窗的背景
<code>clear()</code>	清除視窗
<code>clearscreen()</code>	
<code>reset()</code>	把所有的設定全部都重置
<code>resetscreen()</code>	
<code>screenize()</code>	查詢目前的螢幕尺寸
<code>penup()</code>	把筆拿起來，接下來的動作不會留下繪圖的痕跡
<code>pendown()</code>	把筆放下去，接下來移動的過程都會執行繪圖的工作

視窗操作相關函數

函數或常數	說明
<code>shape()</code>	如果不加參數直接呼叫，會傳回目前的游標的形狀，如需要修改，有以下幾種選擇："arrow"、"turtle"、"circle"、"square"、"triangle" "classic"
<code>shapsize(w, l, o)</code>	傳入3個參數，分別代表寬、長、以及外框
<code>pensize(w)</code>	畫筆的粗細
<code>width(w)</code>	
<code>pen()</code>	傳回目前畫筆的設定狀態
<code>pencolor(colorstring)</code> <code>pencolor((r, g, b))</code> <code>pencolor(r, g, b)</code>	設定畫筆的顏色，有幾種不同的設定方式，如果不傳入參數則會顯示目前的顏色值
<code>fillcolor(colorstring)</code> <code>fillcolor((r, g, b))</code> <code>fillcolor(r, g, b)</code>	設定想要填滿的顏色值，有幾種不同的設定方式，如果不傳入參數則會顯示目前的顏色值，和以下的 <code>begin_fill()</code> 以及 <code>end_fill()</code> 一併使用
<code>color(colorstring1, colorstring2)</code> <code>color((r1, g1, b1), (r2, g2, b2))</code>	同時設定畫筆和填滿顏色
<code>filling()</code>	傳回目前是否處於填滿狀態
<code>begin_fill()</code>	開始設定填滿用的形狀
<code>end_fill()</code>	設定結束，開始填滿顏色的作業

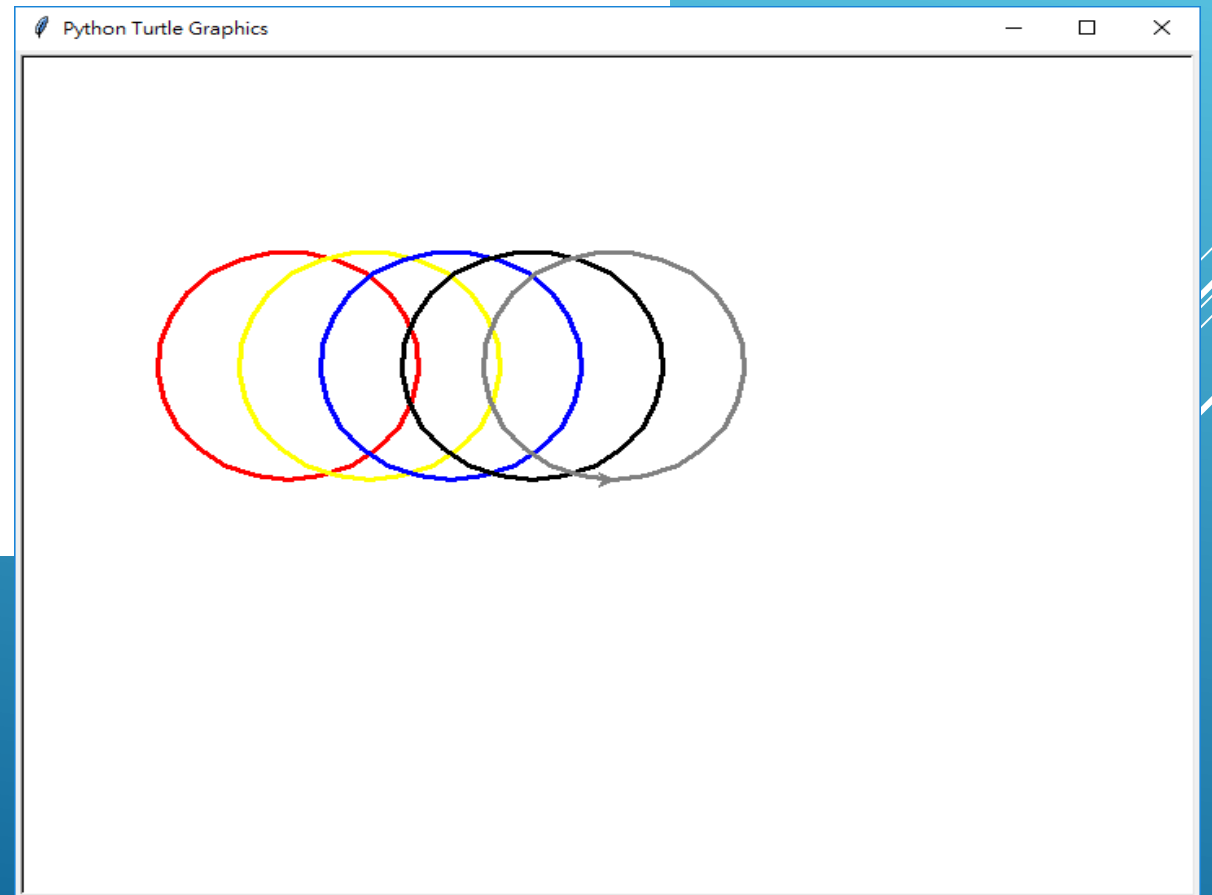
繪圖相關函數

函數或常數	說明
write()	輸出文字訊息
showturtle()	把目前的游標形狀顯示出來
hideturtle()	把目前的游標形狀隱藏起來
forward(move)	前進指定的步數
backward(move)	後退指定的步數
right(angle)	向右旋轉指定的角度
left(angle)	向左旋轉指定的角度
goto(x, y)	直接前往(x, y)座標位置
setpos(x, y)	
setposition(x, y)	
setx(x)	設定x座標位置
sety(y)	設定y座標位置
setheading(angle)	設定游標的面朝角度
home()	回到初始點
circle(r, angle)	依指定的半徑r繪製angle角度的圓
dot(size)	繪製一個指定大小的點
stamp()	把游標形狀留在目前的位置 (蓋章)
clearstamp()	清除蓋章的內容
speed(n)	設定繪製及移動的速度

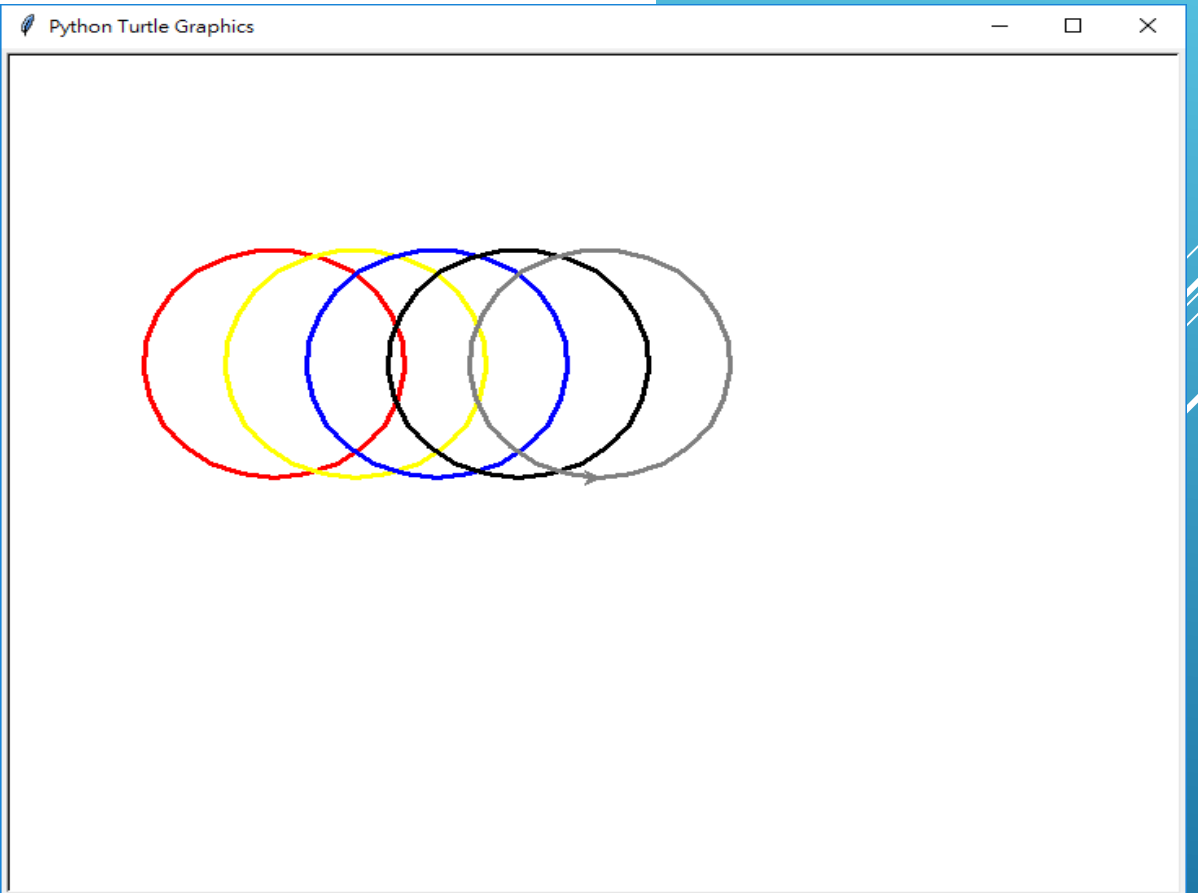
海龜動作相關函數

```
import turtle as tu
circle_colors = ['red', 'yellow', 'blue', 'black', 'gray']
tu.pensize(3)
tu.penup()
for i in range(len(circle_colors)):
    tu.goto(-200+i*50, 0)
    tu.pendown()
    tu.color(circle_colors[i])
    tu.circle(80,360)
    tu.penup()
tu.done()
```

畫5種不同顏色的圓



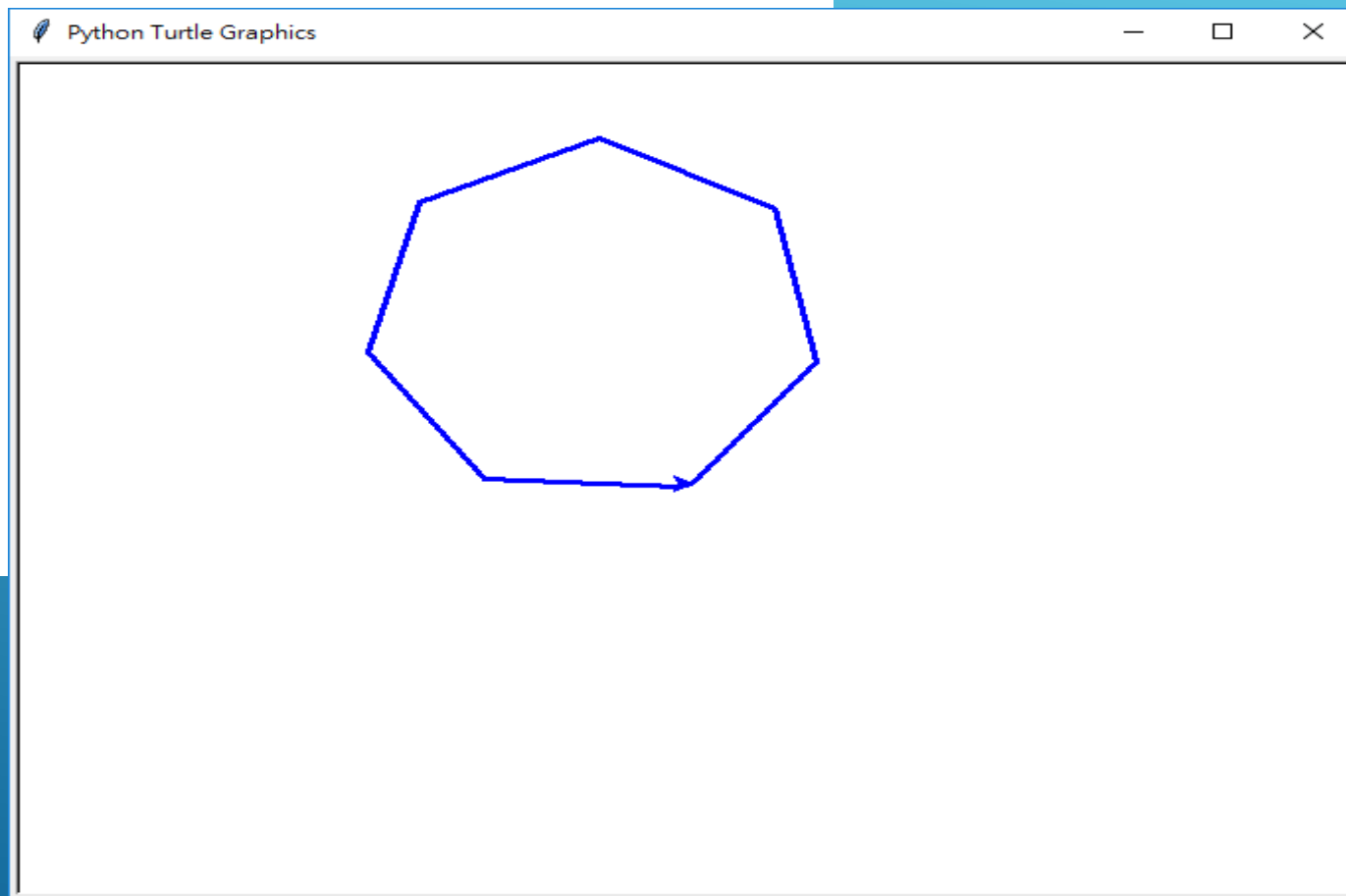

```
import turtle as tu
circle_colors = ['red', 'yellow', 'blue', 'black', 'gray']
tu.pensize(3)
tu.penup()
for i, c in enumerate(circle_colors):
    tu.goto(-200+i*50, 0)
    tu.pendown()
    tu.color(c)
    tu.circle(80,360)
    tu.penup()
tu.done()
```



畫5種不同顏色的圓
使用列舉的版本

```
import turtle as tu
n = int(input("請問要畫幾邊形？"))
tu.pensize(3)
tu.color('blue')
for i in range(n):
    tu.left(360//n)
    tu.forward(100)
tu.goto(0,0)
tu.done()
```

繪製多邊形

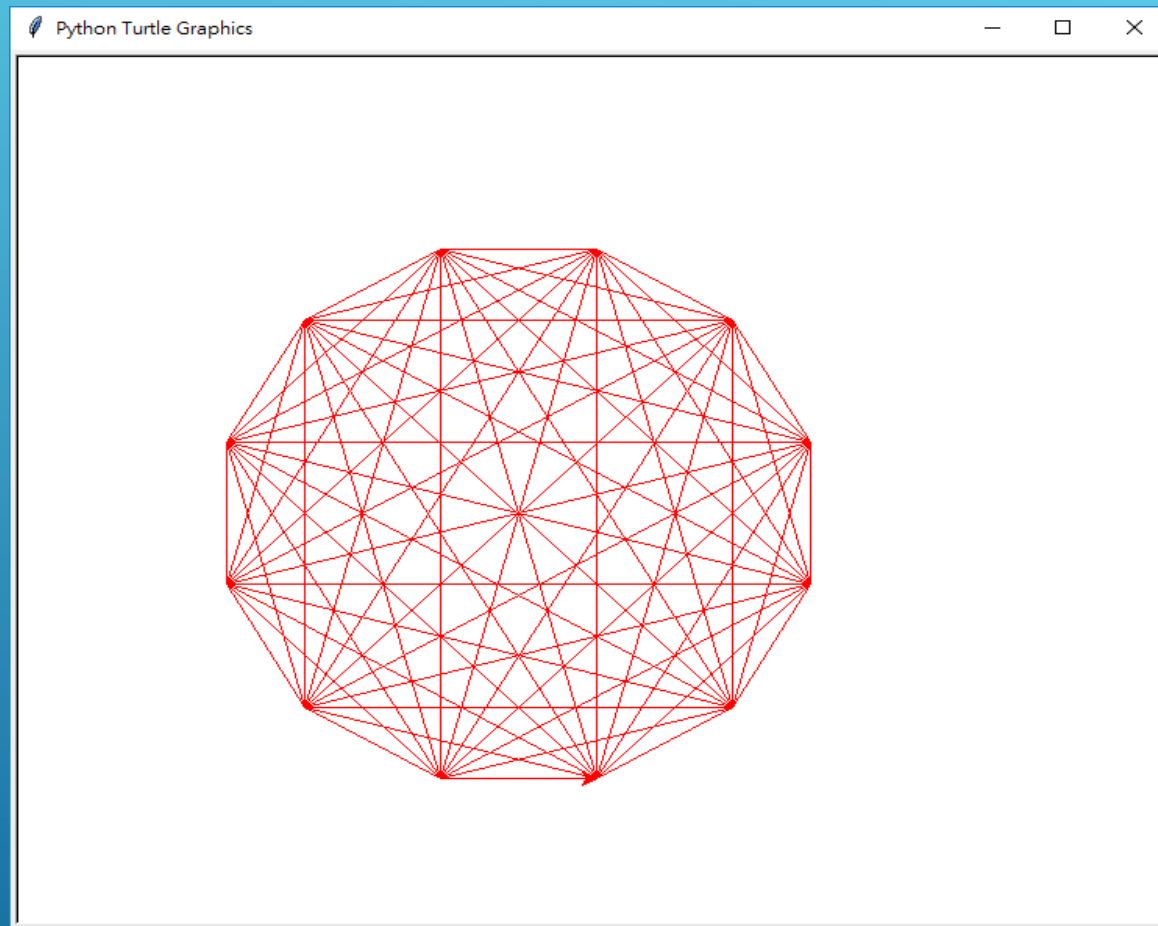


```
import turtle as tu
pos_list = list()
n = int(input("請問要畫幾邊形？"))
tu.color('red')
tu.speed(8)
tu.penup()
tu.goto(0,-200)

for i in range(n):
    tu.left(360//n)
    tu.forward(100)
    pos_list.append(tu.pos())

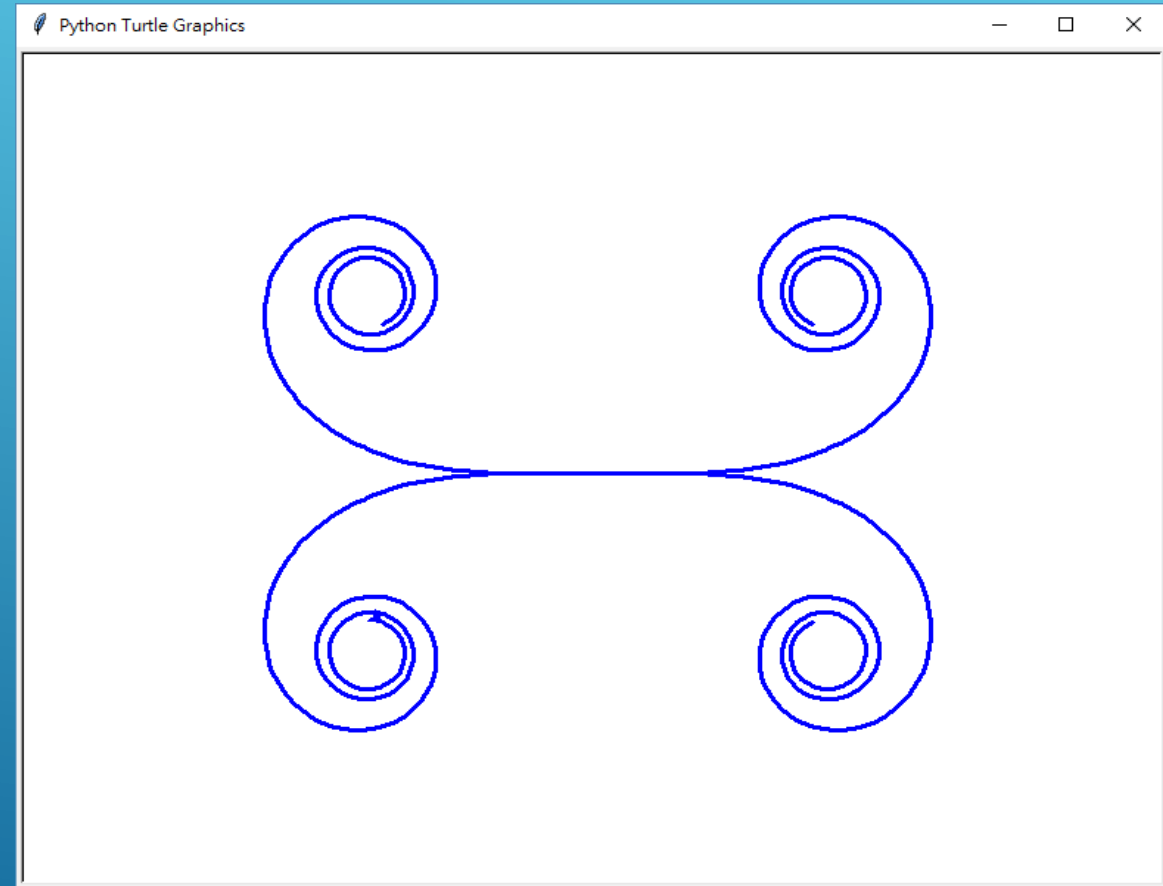
for i in range(len(pos_list)):
    for point in pos_list:
        tu.penup()
        tu.goto(pos_list[i])
        tu.pendown()
        tu.goto(point)
tu.done()
```

繪製多邊形毛線球



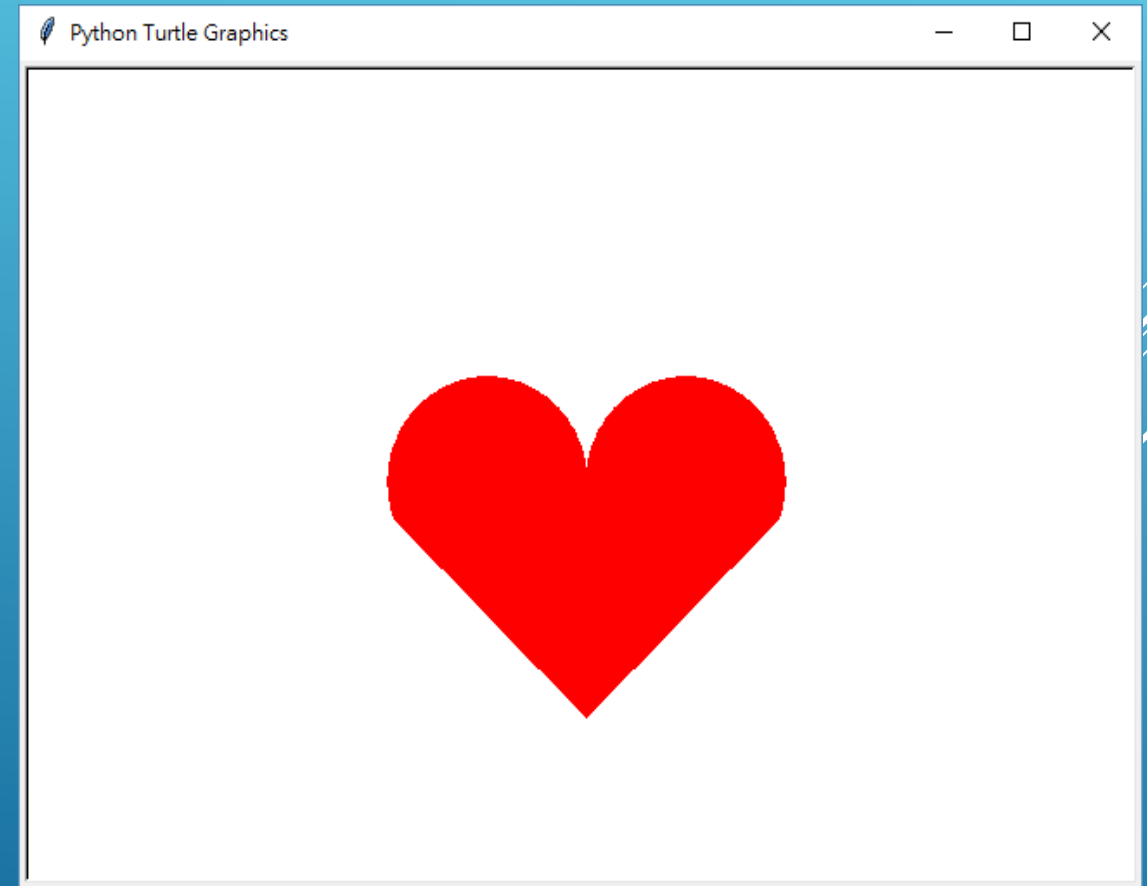
```
import turtle as tu
tu.speed(0)
tu.color('blue')
tu.pensize(3)
for i in range(150):
    tu.left(i//10)
    tu.forward(6)
tu.penup()
tu.home()
tu.pendown()
for i in range(150):
    tu.right(i//10)
    tu.forward(6)
tu.penup()
tu.home()
tu.pendown()
for i in range(150):
    tu.right(i//10)
    tu.backward(6)
tu.penup()
tu.home()
tu.pendown()
for i in range(150):
    tu.left(i//10)
    tu.backward(6)
tu.done()
```

使用漸變技巧繪製 幾何形狀

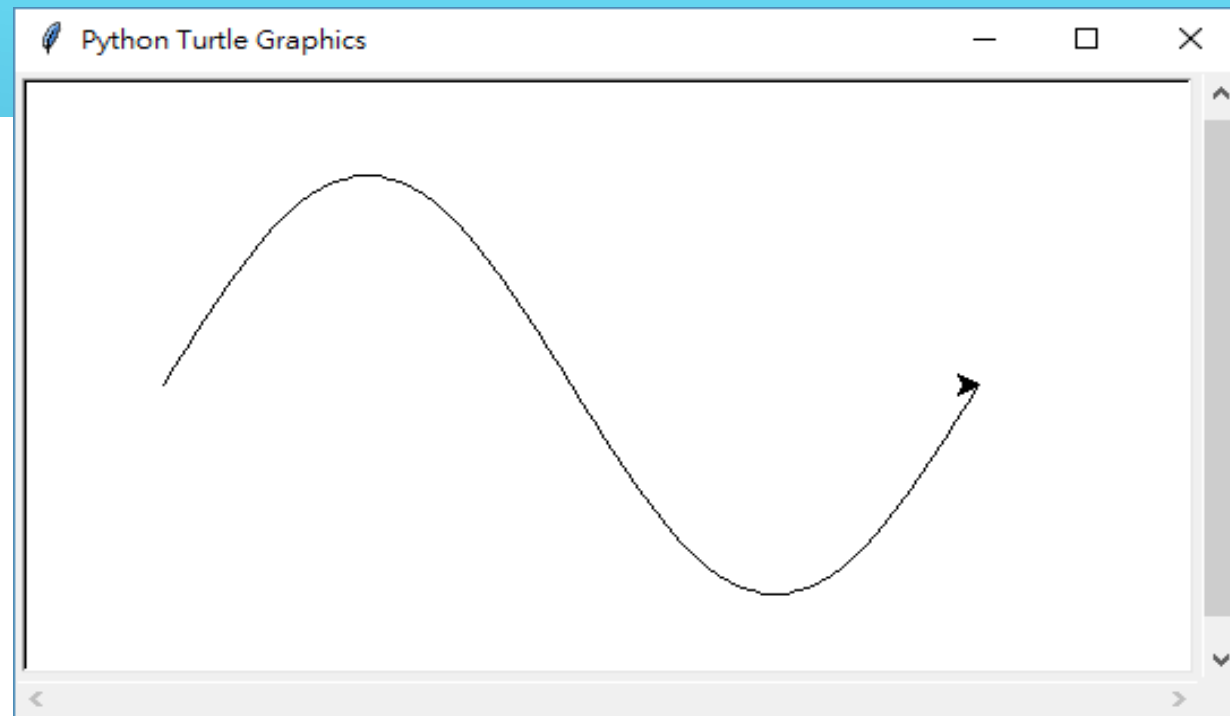


```
import turtle as tu
tu.color('red')
tu.begin_fill()
tu.left(90)
for i in range(200):
    tu.right(1)
    tu.forward(1)
tu.goto(0,-130)
tu.goto(-110.96, -20.57)
tu.penup()
tu.goto(0,0)
tu.pendown()
tu.home()
tu.left(90)
for i in range(200):
    tu.left(1)
    tu.forward(1)
tu.end_fill()
tu.ht()
tu.done()
```

使用漸變技巧繪製 愛心圖形



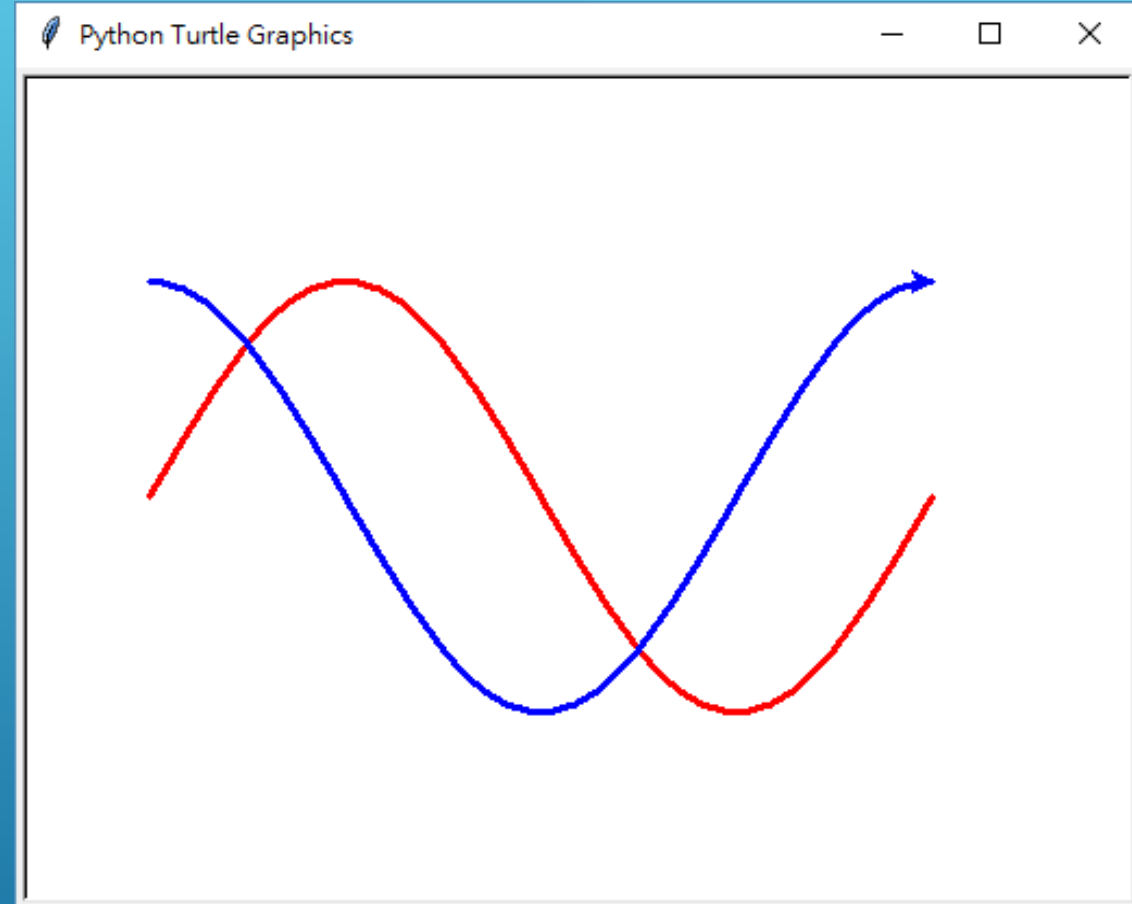
```
1: import math
2: import turtle as tu
3: tu.speed(8)
4: tu.penup()
5: tu.goto(-200,0)
6: tu.pendown()
7: for d in range(0, 361, 5):
8:     tu.goto(d-200, 100*math.sin(d*math.pi/180))
9: tu.done()
```



繪製SIN函數圖形

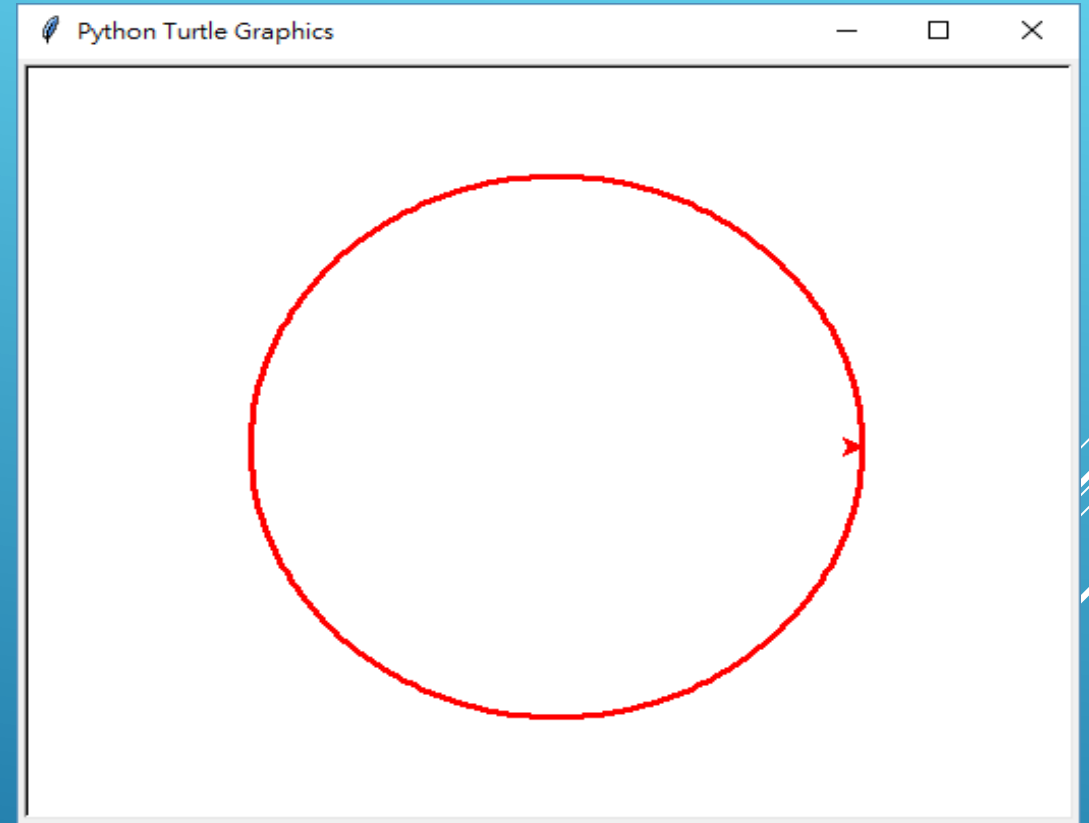
繪製SIN及COS函數圖形

```
import math
import turtle as tu
tu.speed(8)
tu.pensize(3)
tu.penup()
tu.goto(-200,0)
tu.color('red')
tu.pendown()
for d in range(0, 361, 5):
    tu.goto(d-200, 100*math.sin(d*math.pi/180))
tu.penup()
tu.goto(-200,100)
tu.color('blue')
tu.pendown()
for d in range(0, 361, 5):
    tu.goto(d-200, 100*math.cos(d*math.pi/180))
tu.done()
```



```
import math
import turtle as tu
tu.speed(8)
tu.pensize(3)
tu.penup()
tu.goto(150,0)
tu.color('red')
tu.pendown()
for d in range(0, 361, 2):
    x = 150*math.cos(d*math.pi/180)
    y = 150*math.sin(d*math.pi/180)
    tu.goto(x, y)
tu.done()
```

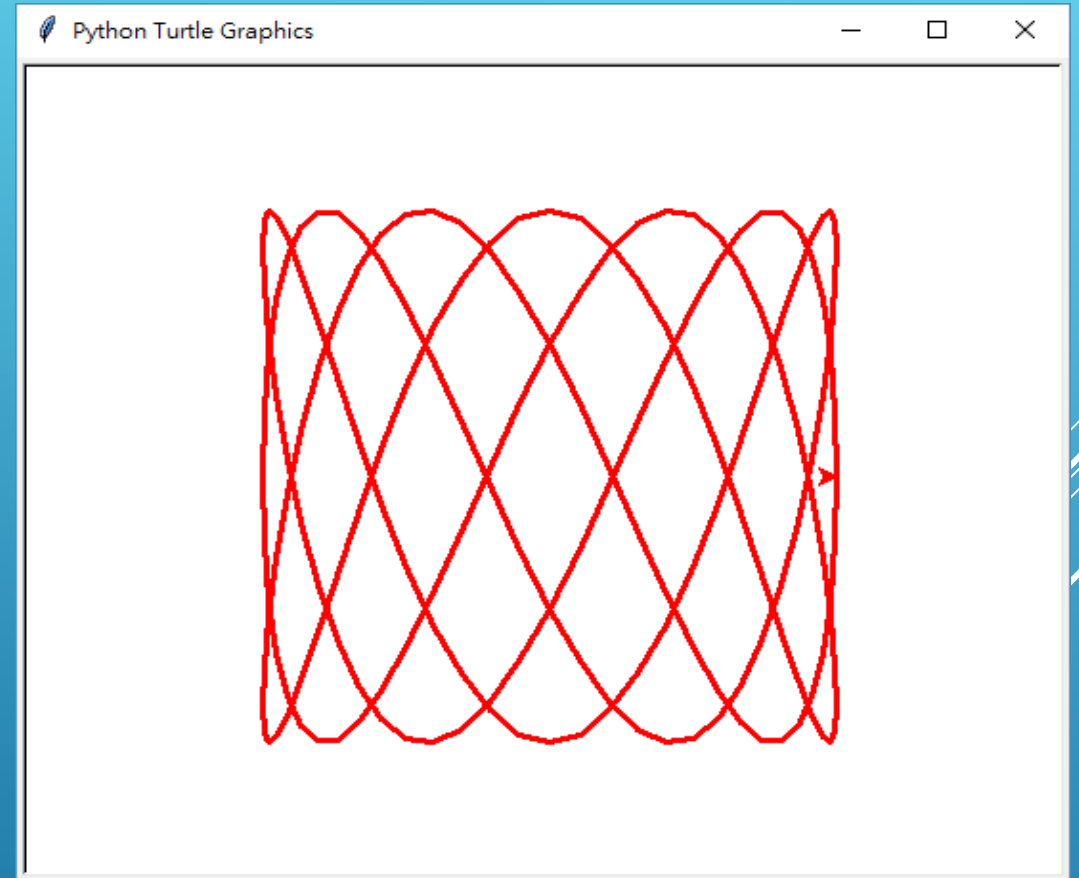
利用三角函數畫圓



$$\begin{cases} x = r * \cos \theta \\ y = r * \sin \theta \end{cases}$$


```
import math
import turtle as tu
tu.speed(8)
tu.pensize(3)
tu.penup()
tu.goto(150,0)
tu.color('red')
tu.pendown()
for d in range(0, 361, 2):
    x = 150*math.cos(3*d*math.pi/180)
    y = 150*math.sin(7*d*math.pi/180)
    tu.goto(x, y)
tu.done()
```

利用三角函數畫李沙育圖形

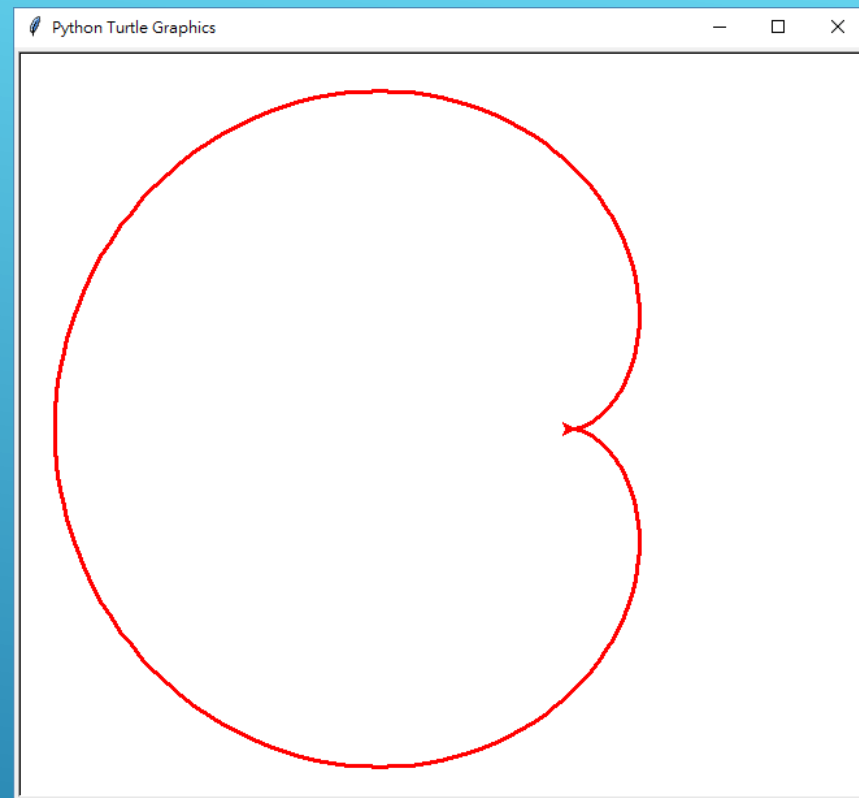


```
import math
import turtle as tu

def th(degree):
    return degree*math.pi/180

r = 100
tu.speed(8)
tu.pensize(3)
tu.penup()
tu.color('red')
for d in range(0, 361, 2):
    x = 2*r*(math.cos(th(d)) - 0.5 * math.cos(2*th(d)))
    y = 2*r*(math.sin(th(d)) - 0.5 * math.sin(2*th(d)))
    if d:
        tu.pendown()
    tu.goto(x, y)
tu.done()
```

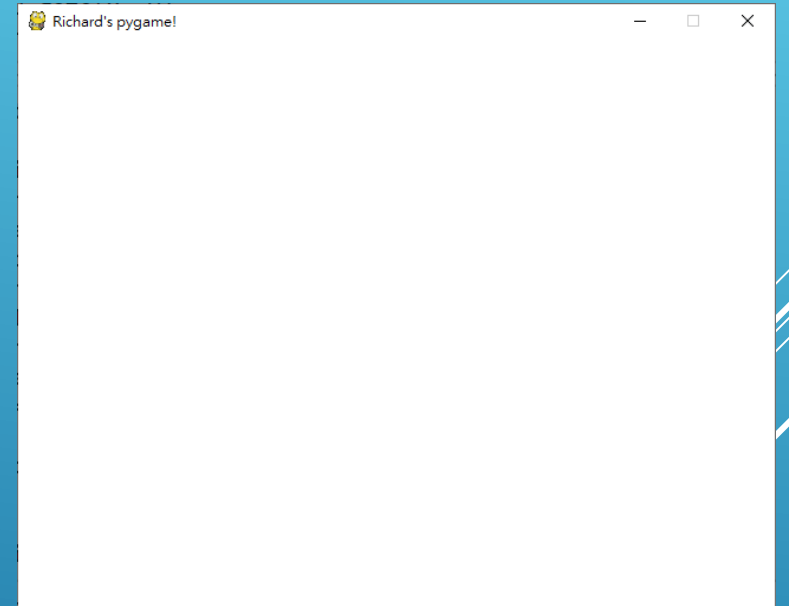
利用三角函數畫心臟線



$$\begin{cases} x(\theta) = 2r(\cos\theta - \frac{1}{2}\cos 2\theta) \\ y(\theta) = 2r(\sin\theta - \frac{1}{2}\sin 2\theta) \end{cases}$$

PYGAME繪圖基本架構

```
1: import pygame as pg
2:
3: pg.init()
4: screen = pg.display.set_mode((640, 480))
5: pg.display.set_caption("Richard's pygame!")
6: bk = pg.Surface(screen.get_size())
7: bk.fill((255,255,255))
8: screen.blit(bk, (0,0))
9: pg.display.update()
10: quit = False
11: while not quit:
12:     for event in pg.event.get():
13:         if event.type == pg.QUIT:
14:             quit = True
15: pg.quit()
```



模組名稱	用途
cdrom	對CDROM的音樂播放操作
cursors	游標的相關操作
display	用來控制顯示視窗及螢幕的模組
draw	在Surface上繪製形狀
event	管理事件處理
font	建立和顯示TrueType的字型
image	和影像相關的操作
joystick	管理搖桿相關的功能
key	處理和鍵盤相關的功能
mouse	處理和滑鼠相關的功能
sndarray	使用numpy處理音效的部份
surfarray	使用numpy處理影像的部份
time	控制計時相關功能
transform	對於影像的相關轉換功能

PYGAME模組列表

模組名稱	用途
init	對display模組的初始化作業
quit	解除display模組的作業
set_mode	用來初始化windows或screen，並傳回一個Surface，可以設定長寬以及深度（ depth ）
get_surface	取得目前設定的Surface
flip	刷新Surface到screen
update	更新螢幕中有修改到的部份
set_caption	設定pygame視窗的標題

DISPLAY模組常用函數

函數名稱	用途
blit	把一個影像畫到另一個影像
blits	把多個影像畫到另一個影像
convert	改變影像的像素格式
copy	建立一個Surface的新複本
fill	填滿一個顏色
get_size	取得影像的尺寸

SURFACE模組常用函數

函數名稱	用途
rect	繪製矩形
polygon	繪製多邊形
circle	繪製圓形
ellipse	以矩形框來繪製框內圓形
arc	繪製弧線
line	繪製線段
lines	繪製連續線段
aaline	繪製無鋸齒的線段
aalines	繪製無鋸齒的連續線段

DRAW模組支援的繪圖函數

```
import pygame as pg
```

```
pg.init()
```

```
screen = pg.display.set_mode((640, 480))
```

```
pg.display.set_caption("Richard's pygame!")
```

```
bk = pg.Surface(screen.get_size())
```

```
bk.fill((255,255,255))
```

```
pg.draw.rect(bk, (255, 0, 0), (100, 100, 300, 300), 2)
```

```
screen.blit(bk, (0,0))
```

```
pg.display.update()
```

```
quit = False
```

```
while not quit:
```

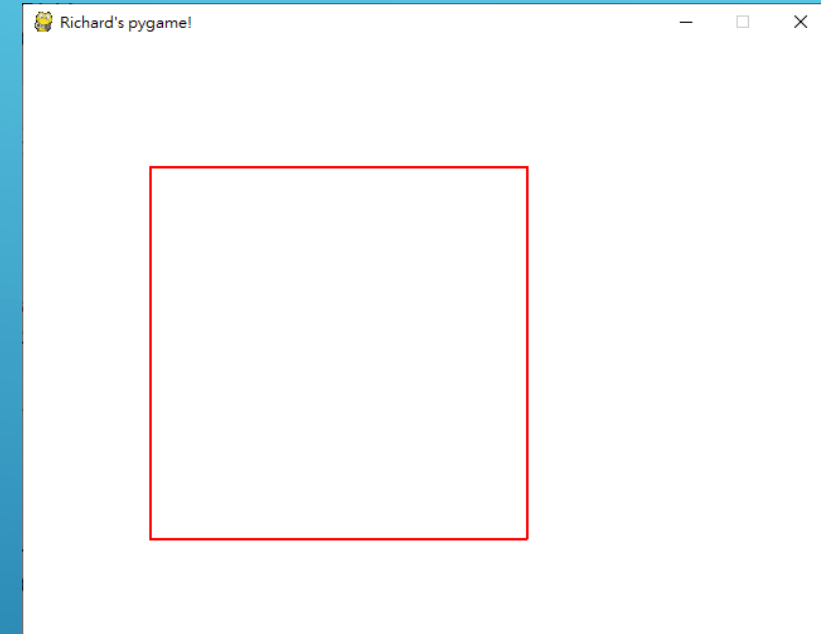
```
    for event in pg.event.get():
```

```
        if event.type == pg.QUIT:
```

```
            quit = True
```

```
pg.quit()
```

PYGAME繪製矩形




```
import random
import pygame as pg

pg.init()
screen = pg.display.set_mode((640, 480))
pg.display.set_caption("Richard's pygame!")
bk = pg.Surface(screen.get_size())
bk.fill((255,255,255))
for i in range(50):
    r = random.randint(0, 255)
    g = random.randint(0, 255)
    b = random.randint(0, 255)
    x = random.randint(0, 640)
    y = random.randint(0, 480)
    w = random.randint(10, 100)
    h = random.randint(10, 100)
    r = random.randint(10, 100)
    pg.draw.ellipse(bk, (r, g, b), (x, y, w, h), 2)
    pg.draw.circle(bk, (r, g, b), (x, y), r, 2)

screen.blit(bk, (0,0))
pg.display.update()
quit = False
while not quit:
    for event in pg.event.get():
        if event.type == pg.QUIT:
            quit = True
pg.quit()
```

PYGAME隨機圖形



```
import pygame as pg
import math

pg.init()
screen = pg.display.set_mode((640, 480))
pg.display.set_caption("Richard's pygame!")
bk = pg.Surface(screen.get_size())
bk.fill((255,255,255))
lines = list()
for th in range(0, 361):
    y = 250 - 200 * math.sin(th*math.pi/180)
    lines.append((th+140, y))
pg.draw.lines(bk, (0, 0, 255), False, lines, 2)

screen.blit(bk, (0,0))
pg.display.update()
quit = False
while not quit:
    for event in pg.event.get():
        if event.type == pg.QUIT:
            quit = True
pg.quit()
```

PYGAME繪製函數圖形

