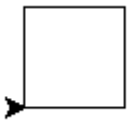
# Drawing Patterns: Turtle Graphics in Python

班級：　　座號：　　姓名：　　Date:

## 1. To draw shapes in Turtle Graphics

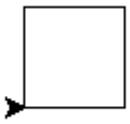
**Square1.py　(邊長 50 的正方形)**

```
from turtle import *
canvas = Screen()
canvas.setup(400,200)

sarah = Turtle()
sarah.forward(50)  # make sarah draw a square
sarah.left(90)
sarah.forward(50)
sarah.left(90)
sarah.forward(50)
sarah.left(90)
sarah.forward(50)
sarah.left(90)

canvas.exitonclick()
```



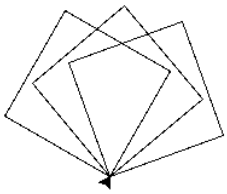
**Square2.py　(邊長 50 的正方形)**

```
from turtle import *
wn = Screen()
sarah = Turtle()

for i in range(4):      #repeat four times
    sarah.forward(50)
    sarah.left(90)

wn.exitonclick()
```



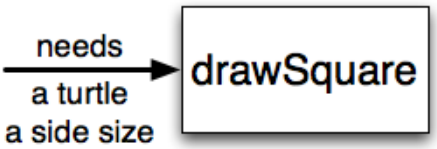
| Triangle.py (邊長 150 的三角形) | |
|---|---|
| | |

| Rectangle.py (長 150 寬 100 的長方形) | |
|---|---|
| | |

| **TiltedSqaure.py** (邊長 150 的迴旋正方形) | |
|---|---|
| Draw a tilted square. And another one, and another one. You can experiment with the angles between the individual squares.<br>  <br>The picture shows three 20 degree turns. But you could try 20, 30 and 40 degree turns, for example. | |
| | |

## 2. To make procedure in Python to enter command quicker.

def name( parameters ):
    statements



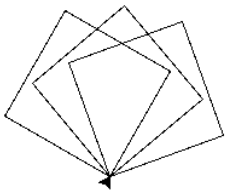

**Square1.py (邊長 150 的正方形)**

```
from turtle import *

def drawSquare(t, size):
    """Make turtle t draw a square of with side size."""
    for i in range(4):
        t.forward(size)
        t.left(90)

wn = Screen()                    # Set up the window and its attributes
alex = Turtle()                  # create alex
drawSquare(alex, 150)            # Call the function to draw the square
wn.exitonclick()
```

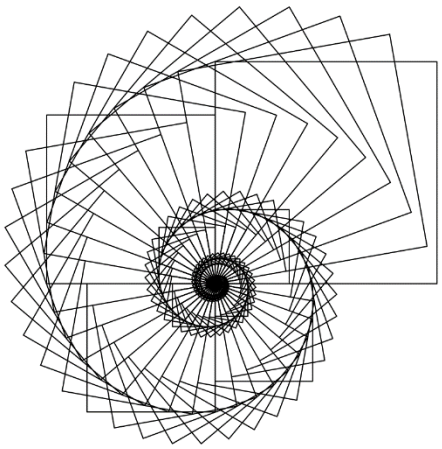
**TiltedSqaure.py (邊長 150 的迴旋正方形)**

Draw a tilted square. And another one, and another one. You can experiment with the angles between the individual squares.

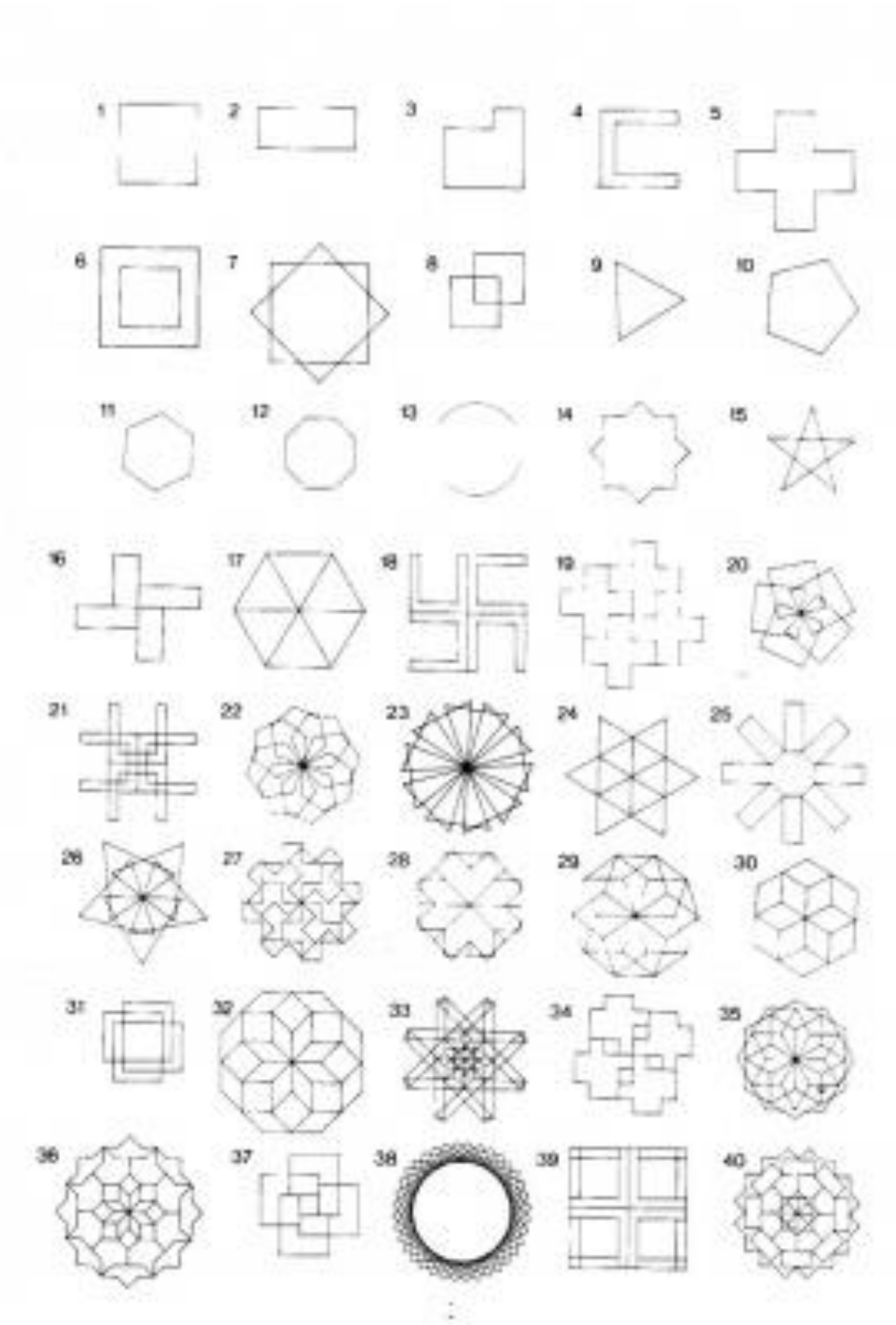
The picture shows three 20 degree turns. But you could try 20, 30 and 40 degree turns, for example.



## SpiralSqaure.py (邊長 1-250 的迴旋正方形)



【Activity】: Create your geogebra graphics

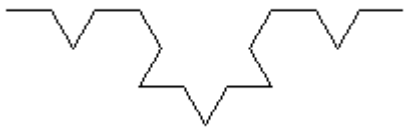


## Fractals – Using Procedure

## passing data through parameters and Recursive

### 1. Koch Curve

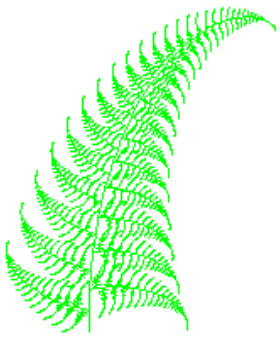
```
from turtle import *

def f(t, length, depth):
    if depth == 0:
        t.forward(length)
        return
    else:
        f(t, length/3, depth-1)
        t.right(60)
        f(t, length/3, depth-1)
        t.left(120)
        f(t, length/3, depth-1)
        t.right(60)
        f(t, length/3, depth-1)

wn = Screen()          # Set up the window and its attributes
koch = Turtle()        # create koch
koch.hideturtle()
f(koch, 200, 2)     # Call the function to draw the koch Curve
wn.exitonclick()
```



**Try:** `f(koch, 200, N)`

## 2. Fern Procedure (passing data through parameters and Recursive)

```
from turtle import *

def fern(t, size):
    if size < 5 :
        return
    t.forward(size/25)
    t.left(80)
    fern(t, size*.3)
    t.right(82)
    t.forward(size/25)
    t.right(80)
    fern(t, size*.3)
    t.left(78)
    fern(t, size*.9)
    t.left(2)
    t.back(size/25)
    t.left(2)
    t.back(size/25)

wn = Screen()          # Set up the window and its attributes
mytree = Turtle()    # create mytree
mytree.tracer(0,0)
mytree.left(90)
mytree.hideturtle()
fern(mytree,305)     # Call the function to draw the fern
wn.exitonclick()
```



【Try it】:

- **fern 600**
- **change "left" "right"**

## 3. tree Procedure (passing data through parameters and Recursive)

```
from turtle import *

def tree(t, order, len):
    if order == 1 :
        t.forward(len)
        t.left(30)
        t.forward(len)
        t.backward(len)
        t.right(30)
        t.right(30)
        t.forward(len)
        t.backward(len)
        t.left(30)
        t.backward(len)
        return
    else :
        t.forward(len)
        t.left(30)
        tree(t, order-1, len*.75)
        t.right(30)
        t.right(45)
        tree(t, order-1, len*.75)
        t.left(45)
        t.backward(len)

wn = Screen()        # Set up the window and its attributes
mt= Turtle()   # create mytree
mt.tracer(0,0)
mt.left(90)
mt.hideturtle()
tree(mt, 9, 50)    # Call the function to draw the tree
wn.exitonclick()
```

【Try it】:

- tree 9 50  tree 12 100  change "LT" "RT"
- change LT angle and RT angle

4. `fractal` **Procedure (passing data through parameters and Recursive)**

Results: