Python Turtle

Python Turtle is a fun way to play with Python. It allows you to type in some basic commands and see the turtle move on screen.

Imagine a robotic turtle, where each command you type, the robot turtle responds. Sounds like fun? Well, let's have a go.

Drawing a circle

Create a new python script file called **turtle-circle.py**. Type:

```
from turtle import *      # import the turtle module

color('green') # turn turtle green
up() # raise pen
goto(0, -50) #move turtle to centre of screen and -50 on the y axis
down() #pen down
circle(50) #draw a circle 50 pixels in diameter
up()#raise pen and stop drawing
```

Save and press F5.

Drawing a square

Create a new python script file called **turtle-square.py**. Type:

```
from turtle import *      # import the turtle module

color('blue')
forward(100)
right(90)
forward(100)
right(90)
forward(100)
right(90)
forward(100)
```

Save and press F5.

Another way to make a square:

```
count=0
while count <=3:
        forward (100)
        right(90)
        count += 1
```

Drawing diagonal lines

Create a new python script called **turtle-diagonal.py**. Import the turtle module and type:

```
color('magenta')

count=0
while count <=7:
        left(45)
        forward(40)
```

What shape does this create?

Changing the Turtle's Dynamics

Changing the Colour

Most of the webpage colours can be used in turtle. Colours like 'dodgerblue' and 'limegreen' can be used. See Appendix 1 for the colour list.
Type, **pencolor('color')** to change the pen colour.
Type **bgcolor('color')** to change the background colour.
Type your chosen colour in the parenthesis.

Changing the Speed

To make the turtle move faster, type **speed(0)**. You can change the speed by changing the number between the parenthesis, zero [0] is the fastest speed while 10 is really slow.

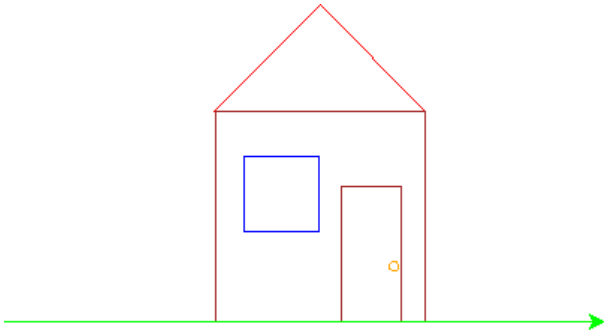
Changing the Turtle

To hide the turtle type **hideturtle()**. To show the turtle, type **showturtle()**. You can also type, **shape('turtle')** to change the arrow into a turtle shape.

Changing the Pen Thickness

The pen is automatically set at 1 pixel thick. To change the thickness, type **pensize(3)**. The number in the parenthesis is the thickness in pixels.

Task 1: Building a House



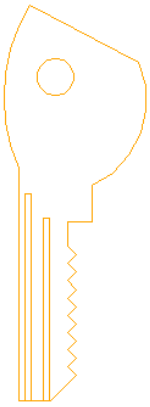
Get turtle to draw a house with a window and door. Save it at **turtle_house.py**.

Don't forget to put comments into your script. Your script should be no more than 50 lines in length.

Task 2: Concentric Circles

Get turtle to draw two five different coloured circles, one inside the other. Save as **turtle_concentricCircles.py**. Don't forget to put comments into your script. Your script should be no more than 30 lines in length.

Task 3: 8 Pointed Star

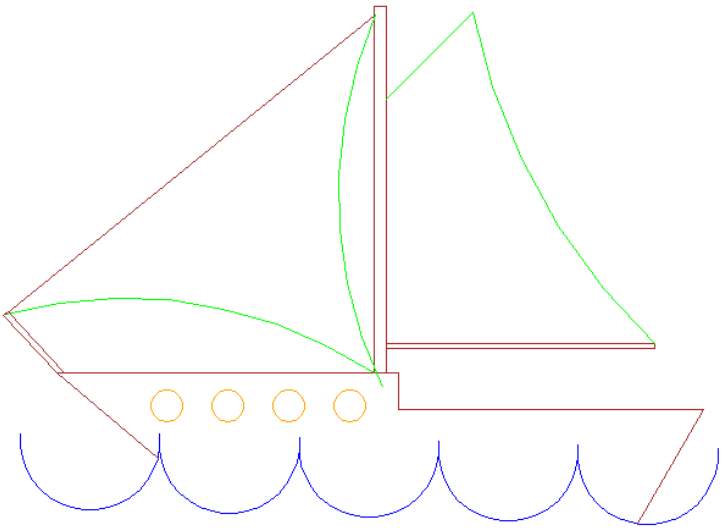


Get turtle to draw an eight pointed star, save as **turtle_8PointedStar.py**. Don't forget to put comments into your script. Your script should be no more than 10 lines in length.

Task 4: School Key

Get turtle to draw a key, save as **turtle_key.py**. Don't forget to put comments in the script. Your script should be no more than 80 lines in length.

Task 5: Yacht

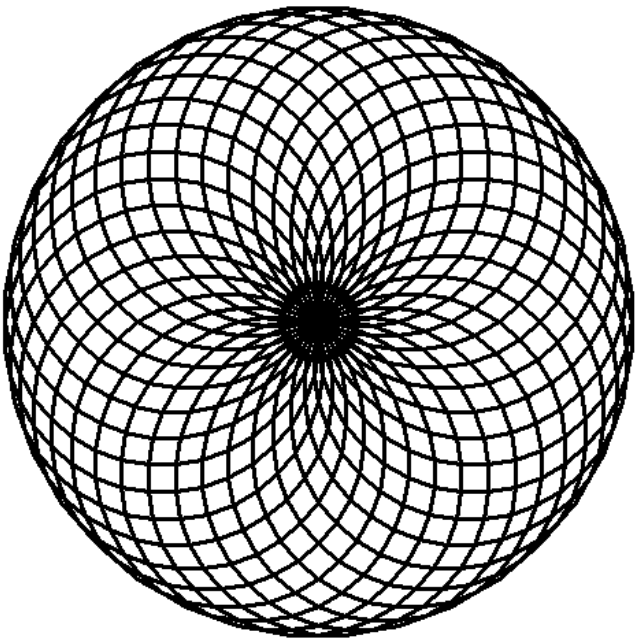


Get turtle to draw a yacht, save it as **turtle_yacht.py**. Don't forget to put comments in the script. Make your script as short as possible. Try to use functions.

Task 6: Grid

Get turtle to draw a grid that covers the Turtle screen, save it as **turtle_grid.py**. The grid must be 20px X 20px. Don't forget to put comments in the script. Your script should be no more than 30 lines in length. Try to use functions.

Making Patterns



Turtle can also be used to make patterns, like spirograms, tangrams, fractals and kaleidoscopes.

Spirograms

Create a new python script, call it **turtle_spirogram1.py**.

```
from turtle import *
def spinout(n,s):
        for spin in range(n):
                right(360./n)
                forward(s)
def spinin(n, s):
        for spin in range(n):
                right(360./n)
                spinout(n, s)
def main():
        speed(0)
        hideturtle()
        bgcolor('black')
        pencolor('purple')
        pensize(3)

        tracer(40,0) #draws entire pattern, if this is hidden, will draw separate circles
        spinin(40,20) #40 is width of pattern, 20 is height

main()
```

By changing the integers, see what you can make.