

Turtle Graphics

Try the following commands in Python:

```
import turtle  
turtle.shape('turtle')
```

Try making the window size a little smaller so that it's easier to work with.

```
turtle.setup(500,500)
```

Try changing the background color of the screen to blue:

```
turtle.Screen().bgcolor("blue")
```

The **bgcolor()** function is a prewritten block of code that changes the background color of the turtle's screen to one you decide.

Here is what is happening:

1. We tell the computer which object we want to interact with. In this case, it's the **Screen**. Because the **Screen** object belongs to the turtle module, we make this connection known using the dot notation. So, we are telling the computer that we want to specifically use the **Screen** object that belongs to the turtle module, we use a dot (.) in between them. That's how we get the first part:

```
turtle.Screen()
```

2. But we are not finished yet! We still have to tell the computer to use a specific function that belongs to the Screen object to change the color. In our case, it's the **bgcolor()** function. Just as before, we put a dot in between the **Screen** object and the name of the function we want to use:

```
turtle.Screen().bgcolor()
```

Finally, we give the **bgcolor()** function a color:

```
turtle.Screen().bgcolor("blue")
```

3. So altogether, the computer understands our code to mean:
 1. Please find the turtle module's **Screen** object.
 2. When you do, find the **bgcolor()** function that belongs to it.
 3. Finally, do what the **bgcolor()** function says to do with the color we've given it. In this case, it is blue.

Remember, we didn't write the code for this; it's already written for us in the turtle module. That's why we needed to import the turtle module first before using it. Now, the computer can

go through the turtle module's code, find the objects and functions we are asking it to use, and run the code that is already written for us.

COLORS AND SIZES

You can change the color of the turtle by specifying the name of the color or the color code. Here are the color codes for the common colors used:

blue, green, red, yellow, orange, white, black, brown, gray, pink, purple, aqua, maroon, crimson, violet, magenta.

Try the following commands:

```
turtle.color("yellow")
```

Try changing the color of the turtle to different colors.

```
turtle.turtlesize(10, 10, 2) ← turtlesize changes the size of the turtle (too big!)
```

The **turtlesize()** function uses three numbers as its input: The first and second numbers are used to stretch the turtle lengthwise (up and down) and widthwise (left and right) a certain amount. The third number sets the size of the turtle's outline (to make the line thinner or thicker).

```
turtle.resizemode('auto') ← resets the turtle back to its original size
```

```
turtle.turtlesize(3, 2, 2)
```

What if you only wanted to change the turtle's outline without changing its height or its width?

```
turtle.turtlesize(outline=30) (Too thick!)
```

```
turtle.turtlesize(outline=3) (Better!)
```

MOVING THE TURTLE

Try the following commands:

```
turtle.forward(200)
```

```
turtle.back(350)
```

```
turtle.left(90)
```

```
turtle.forward(300)
```

CREATING A PEN

```
pen = turtle.Turtle()
```

 ← creating an instance of the turtle object and calling it
"pen"

```
pen.color("blue")  
pen.pensize(5)  
pen.forward(100)  
pen.color("orange")
```

DRAWING A SQUARE

```
pen.forward(100)  
pen.left(90)  
pen.forward(100)  
pen.left(90)  
pen.forward(100)  
pen.left(90)  
pen.forward(100)  
pen.hideturtle()
```

 ← hides the arrow shape

FILLING SHAPES WITH COLOR

```
pen.fillcolor('blue')
```

 ← telling the computer we want to fill the shape we are
about to draw with a blue color

```
pen.begin_fill()
```

```
pen.circle(50)
```

 ← draws a circle of radius 50

```
pen.end_fill()
```

```
pen.circle(100)
```

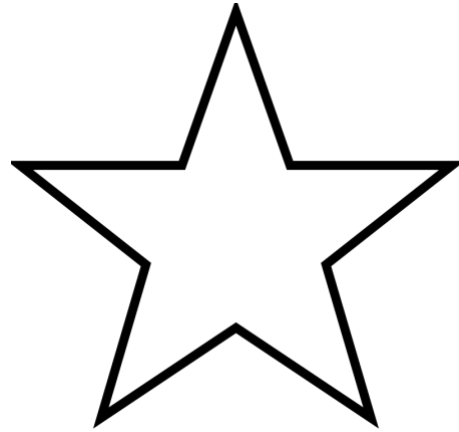
 ← draws a circle of radius 100

```
pen.circle(100, 180)
```

 ← draws a circle for radius 100 but only for 180 (a semicircle)

DRAWING STARS

```
# Drawing a star
import turtle as t
for x in range (1,9):
    t.forward(100)
    t.left(225)
t.hideturtle()
```



```
#Drawing another star
import turtle as t
for x in range (1,38):
    t.forward(100)
    t.left(175)
t.hideturtle()
```

STAMP FUNCTION

The **stamp()** function “stamps” a copy of the shape you select each time you use it. Let’s try the **stamp()** function using a turtle shape.

```
turtle_stamp = turtle.Turtle()
turtle_stamp.shape('turtle')
turtle_stamp.color('green')
turtle_stamp.penup() ← hides the line that is drawn when you move the turtle
turtle_stamp.forward(100)
turtle_stamp.stamp()
turtle_stamp.left(90)
turtle_stamp.forward(100)
turtle_stamp.stamp()
turtle_stamp.left(90)
turtle_stamp.forward(100)
turtle_stamp.stamp()
turtle_stamp.left(90)
turtle_stamp.forward(100)
turtle_stamp.stamp()
```



Let’s try the **stamp()** function again, but this time using a loop.

```
import turtle
import random
#creates a stamp
stamp=turtle.Turtle()
#make it a turtle shape
stamp.shape('turtle')
#lift the color of the stamp so we don't draw a continuous line
stamp.penup
#set RGB color mode to allow random colors in RGB
```

```
turtle.colormode(255)
#set some variables
#one for the initial distance to move (paces)
# and three more to hold the starting color values
paces = 20
random_red=50
random_green=50
random_blue = 50
#start a for loop to repeat the stamping code
#repeat for 50 times
for i in range(50):
    #use the random function to pick a random number for the red value
    random_red=random.randint(0,255)
    #repeat random function for green
    random_green=random.randint(0,255)
    #repeat random function for blue
    random_blue = random.randint(0,255)
    #set the stamp color with the randomly chosen RGB values
    stamp.color(random_red, random_green, random_blue)
    #stamp a turtle with the colors from the last step
    stamp.stamp()
    #add more paces
    paces += 3
    #move forward by the new number of paces
    stamp.forward(paces)
    #slightly turn direction as we move to start spiraling
    stamp.right(25)
```

WRITE()

The **write()** function enables you to write on the screen.

```
pen = turtle.Turtle()  
pen.write("Turtles rock!")
```

If you wish to change the font (typestyle) and size of the text, we can give the write() function a second parameter. Here, we are using an "Open Sans" font with size 60 and a normal type (as opposed to bold).

```
pen.write("Turtles rock", font=("Open Sans", 60, "normal"))
```

Practice Questions

1. Write a program to create a red color-filled rectangle of size 400.
2. Write a program to create a yellow color-filled triangle of size 300 with orange lines
3. Write a program to create an aqua color-filled hexagon (a six-sided shape with equal sides).
4. Write a program to write your name on the screen in green letters of size 50.