**Understanding Turtle Graphics and how to use the Turtle:**

Turtle enables us to draw graphics on the screen.

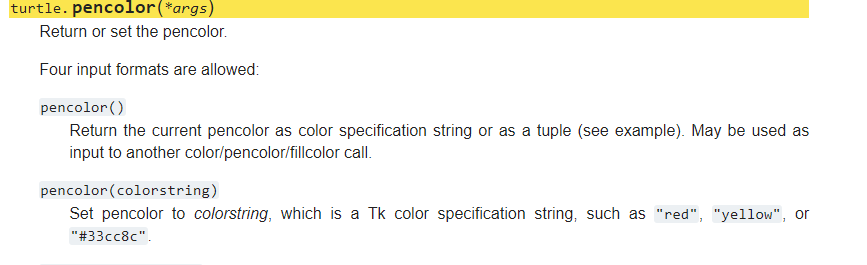
Turtle takes color from the Tk color string.

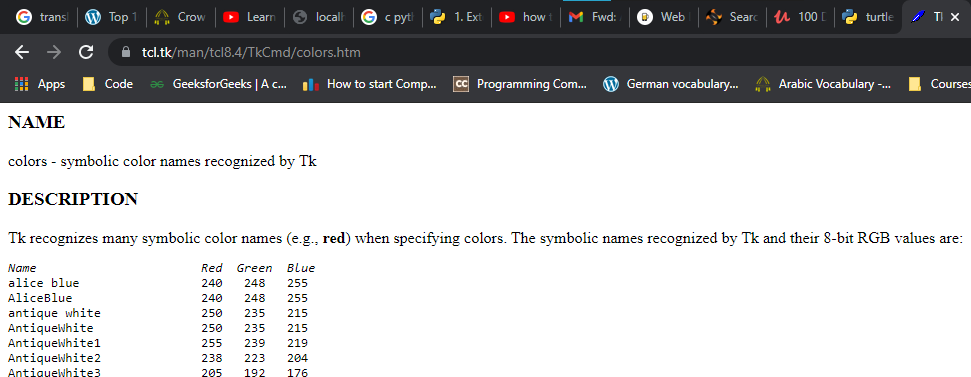
from turtle import Turtle, Screen

timmy\_the\_turtle = Turtle()

timmy\_the\_turtle.shape("turtle")

timmy\_the\_turtle.color("red")

Apart from “red”, we can use different colors 



Tk is the short for the module tkinter which is the tk interface. This is the module with the help of which we create Graphical User Interface.

The Text Interface takes a text command; like the console.

Graphical UI can show images, and allows to click, drag etc. instead of typing commands.

Turtle module relies on tkinter.

from turtle import Turtle, Screen

tim = Turtle()

tim.shape("turtle")

tim.color("coral2")

for i in range(4):

    tim.forward(100)

    tim.right(90)

**Importing Modules, Installing Packages, and working with Aliases:**

Basic Import:

Keyword import + module\_name

from…import…

from turtle import \*

However, importing everything can make our understanding to be confusing.

Aliasing Modules:

import turtle as t

Installing Modules:

There are some modules we can’t just import.

Sometimes, when the modules are not bundled with python modules, we need to install them. There are various libraries in Python world, hosted over the internet.

**Python Tuples and How to generate Random RGB colors:**

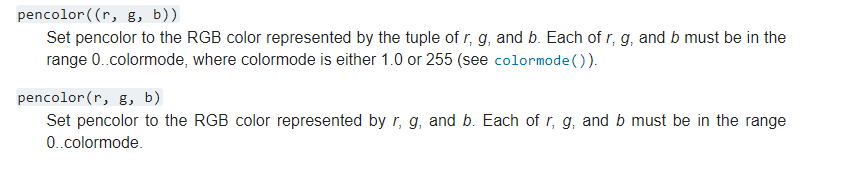
Tuples:

Tuple is a datatype in python. It is very similar to list. => Items are ordered. (1, 3, 8)

Tuples are immutable.

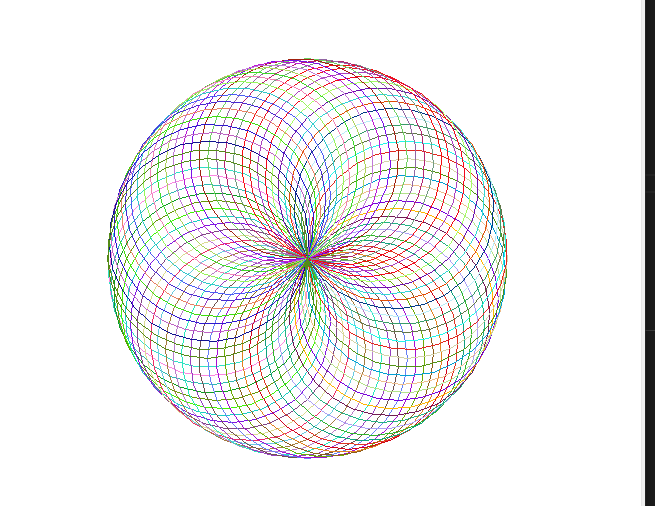
To change it, we can actually convert it into a list.

In turtle, the color is represented by a tuple so that we can’t change it. (r, G, B)



To generate the random colors, we first need to change the colormode of the **turtle** module.

**Spirograph:**



import turtle as t

import random

tim = t.Turtle()

t.colormode(255)

def random\_color():

    r = random.randint(0, 255)

    g = random.randint(0, 255)

    b = random.randint(0, 255)

    return (r,g,b)

tim.speed('fastest')

for \_ in range(90):

    tim.color(random\_color())

    tim.circle(100)

    current\_heading = tim.heading()

    tim.setheading(current\_heading + 5)

    tim.circle(100)

OR:

import turtle as t

import random

tim = t.Turtle()

t.colormode(255)

def random\_color():

    r = random.randint(0, 255)

    g = random.randint(0, 255)

    b = random.randint(0, 255)

    return (r,g,b)

tim.speed('fastest')

def draw\_spirograph(size\_of\_gap):

    for \_ in range(int(360/ size\_of\_gap)):

        tim.color(random\_color())

        tim.circle(100)

        tim.setheading(tim.heading() + size\_of\_gap)

draw\_spirograph(5)

**The Hirst Painting Project: How to extract RGB Values from images**

Colorgram is a library in python that will help us to extract color from images.

(Extracted colors end up in a palette)

import colorgram

rgb\_colors = []

colors = colorgram.extract('spot\_paint.jpg', 30)

for color in colors:

    r = color.rgb.r

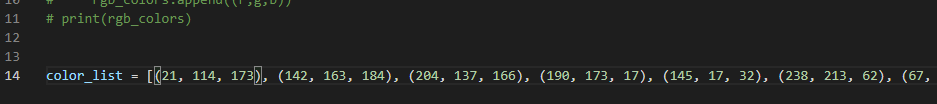
    g = color.rgb.g

    b = color.rgb.b

    rgb\_colors.append((r,g,b))

print(rgb\_colors)

We get the color list from here and we save it into a color\_list.



**Drawing the dots—turtle:**

import turtle as t

import random

t.colormode(255)

tim = t.Turtle()

tim.penup()

tim.hideturtle()

color\_list = [(21, 114, 173), (142, 163, 184), (204, 137, 166), (190, 173, 17), (145, 17, 32), (238, 213, 62), (67, 24, 31), (17, 138, 59), (219, 161, 88), (122, 71, 100), (49, 29, 26), (197, 65, 28), (7, 107, 64), (227, 169, 197), (240, 78, 29), (29, 177, 84), (21, 172, 188), (243, 214, 4), (110, 192, 140), (182, 94, 115), (35, 37, 46), (188, 182, 213), (157, 206, 215), (240, 168, 154), (147, 215, 171), (127, 32, 26)]

tim.speed('fastest')

tim.setheading(225)

tim.forward(300)

tim.setheading(0)

num\_of\_dots = 100

for dot\_count in range(1, num\_of\_dots+1):

    tim.dot(20, random.choice(color\_list))

    tim.forward(50)

    if dot\_count % 10 == 0:

        tim.setheading(90)

        tim.forward(50)

        tim.setheading(180)

        tim.forward(500)

        tim.setheading(0)

screen = t.Screen()

screen.exitonclick()

