

Drawing a square using turtle

```
In [ ]: '''
from turtle import Turtle, Screen

turtle = Turtle()
turtle.shape("turtle")
turtle.color("red")
def square():
    a = 4
    while a != 0:
        turtle.forward(100)
        turtle.left(90)
        a -= 1

square()

screen = Screen()
screen.exitonclick()
'''
```

Other ways of importing modules, installing packages, working with Aliases

Basic import

```
In [ ]: import turtle

pointer = turtle.Turtle()

screen = turtle.Screen()
screen.exitonclick()
```

from... import...

```
In [ ]: from turtle import Turtle, Screen
# keyword(from)
# Module name(turtle)
# keyword(import)
# Thing in the Module(Turtle)
'''

use:
from turtle import *
for importing everything
```

```
'''
pointer_1 = Turtle()
screen_1 = Screen()
screen_1.exitonclick()
```

Alias Method

```
In [ ]: import turtle as t

# just representing turtle package as t
# rest is same as it is
```

Installing Modules

```
In [ ]: '''
we can install a new package using the following code:
-----
pip install <package_name>
'''
```

```
Out[ ]: '\nwe can install a new package using the following code:\n-----
-----\npip install <package_name>\n'
```

Task: To draw a dotted line by using turtle package.

```
In [ ]: '''
from turtle import Turtle, Screen

pointer = Turtle()
a = 15
while a != 0:
    pointer.forward(10) # to draw
    pointer.penup()
    pointer.forward(10) # to move
    pointer.pendown()
    a -= 1

screen = Screen()
screen.exitonclick()
'''
```

```
Out[ ]: '\nfrom turtle import Turtle, Screen\n\npointer = Turtle()\na = 15\nwhile a\n!= 0:\n    pointer.forward(10) # to draw\n    pointer.penup()\n    pointer.\nforward(10) # to move\n    pointer.pendown()\n    a -= 1\n\nscreen = Scre\nen()\nscreen.exitonclick()\n'
```

Task:

To draw a triangle, square, pentagon, hexagon, heptagon, octagon, nanogan, decagon using turtle.

The sides of the shapes should be 100px.

```
In [ ]: '''
from turtle import Turtle, Screen
t = Turtle()
s = Screen()

# triangle
def triangle():
    t.color("red")
    triangle = 3
    sides = 3
    while sides !=0 :
        t.forward(100)
        t.left(360//triangle)
        sides -= 1
# square
def square():
    t.color("blue")
    square = 4
    sides = 4
    while sides !=0 :
        t.forward(100)
        t.left(360//square)
        sides -= 1
# pentagon
def pentagon():
    t.color("green")
    pentagon = 5
    sides = 5
    while sides !=0 :
        t.forward(100)
        t.left(360//pentagon)
        sides -= 1
# hexagon
def hexagon():
    t.color("violet")
    hexagon = 6
    sides = 6
    while sides !=0 :
        t.forward(100)
        t.left(360//hexagon)
        sides -= 1
# heptagon
def heptagon():
    t.color("orange")
    heptagon = 7
```

```

    sides = 7
    while sides !=0 :
        t.forward(100)
        t.left(360//heptagon)
        sides -= 1
# octagon
def octagon():
    t.color("red")
    octagon = 8
    sides = 8
    while sides !=0 :
        t.forward(100)
        t.left(360//octagon)
        sides -= 1
# nonagon
def nonagon():
    t.color("green")
    nonagon = 9
    sides = 9
    while sides !=0 :
        t.forward(100)
        t.left(360//nonagon)
        sides -= 1
# decagon
def decagon():
    t.color("red")
    decagon = sides = 10
    while sides !=0 :
        t.forward(100)
        t.left(360//decagon)
        sides -= 1
def draw():
    triangle()
    square()
    pentagon()
    hexagon()
    heptagon()
    octagon()
    nonagon()
    decagon()

draw()
s.exitonclick()
'''

```

Task:

Draw a Random walk using turtle.

```

In [ ]: '''
import turtle as t
import random as r

```

```

colors = ["red", "green", "blue"]

directions = [0,90,180,270]

pointer = t.Turtle()
pointer.pensize(15)
pointer.speed("fastest")
for _ in range(1000):

    pointer.color(r.choice(colors))
    pointer.forward(30)
    pointer.setheading(r.choice(directions)) # for angle

screen = t.Screen()
screen.exitonclick()
'''

```

Tuples

tuple is a data type ==> (8,9,10)

```

In [ ]: my_tuple = (8,9,10)
        print(my_tuple[0])
        print(my_tuple[1])
        print(my_tuple[2])

```

```

8
9
10

```

Difference between tuples and list

In list, we can change the values inside the list. but when we come into tuples, we can't change the values. and also we can't remove or add elements in the tuple. that's tuple is called as immutable.

```

In [ ]: my_tuple[2] = 12 # try to change the element in the tuple

```

```

-----
TypeError                                Traceback (most recent call last)
Cell In[2], line 1
----> 1 my_tuple[2] = 12 # try to change the element in the tuple

TypeError: 'tuple' object does not support item assignment

```

Task

To draw a spirograph using turtle. the radius of circle is 100.

```
In [ ]: '''
import turtle as t
import random as r

pointer = t.Turtle()
screen = t.Screen()
t.colormode(255)
pointer.speed("fastest")

def random_color():
    red = r.randint(0,255)
    green = r.randint(0,255)
    blue = r.randint(0,255)
    random_color = (red,green,blue)
    return random_color

for _ in range(90):
    pointer.color(random_color())
    pointer.circle(100)
    pointer.left(4)

screen.exitonclick()
'''
```