**Python Turtle Graphics**

**INTRODUCTION :**

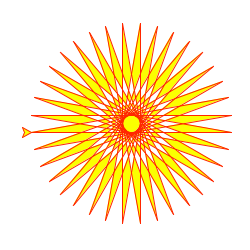
* Turtle is special feature in python used for creating various shapes , pictures and many other drawings.
* Turtle graphics is a popular way for introduced specially for creating programming interests among kids.
* Before introducing turtle in python ,it was part of the original Logo programming language developed by Wally Feurzig and Seymour Papert in 1966.

**Example**:

Imagine a robotic turtle starting at (0, 0) in the x-y plane. After an import turtle, give it the command turtle.forward(15), and it moves (on-screen!) 15 pixels in the direction it is facing, drawing a line as it moves. Give it the command turtle.right(25), and it rotates in-place 25 degrees clockwise.

**Turtle star**

Turtle can draw intricate shapes using programs that repeat simple moves.

**from** **turtle** **import** \*

color('red', 'yellow')

begin\_fill()

**while** **True**:

forward(200)

left(170)

**if** abs(pos()) < 1:

**break**

end\_fill()

done()

The [turtle](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#module-turtle) module is an extended reimplementation of the same-named module from the Python standard distribution up to version Python 2.5.

It tries to keep the merits of the old turtle module and to be (nearly) 100% compatible with it. This means in the first place to enable the learning programmer to use all the commands, classes and methods interactively when using the module from within IDLE run with the -n switch.

The turtle module provides turtle graphics primitives, in both object-oriented and procedure-oriented ways. Because it uses [tkinter](https://docs.python.org/3.3/library/tkinter.html" \l "module-tkinter" \o "tkinter: Interface to Tcl/Tk for graphical user interfaces) for the underlying graphics, it needs a version of Python installed with Tk support.

The object-oriented interface uses essentially two + two classes:

1. The [TurtleScreen](https://docs.python.org/3.3/library/turtle.html?highlight=turtle" \l "turtle.TurtleScreen" \o "turtle.TurtleScreen) class defines graphics windows as a playground for the drawing turtles. Its constructor needs a tkinter.Canvas or a [ScrolledCanvas](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#turtle.ScrolledCanvas) as argument. It should be used when [turtle](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#module-turtle) is used as part of some application.

The function [Screen()](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#turtle.Screen) returns a singleton object of a [TurtleScreen](https://docs.python.org/3.3/library/turtle.html?highlight=turtle" \l "turtle.TurtleScreen" \o "turtle.TurtleScreen) subclass. This function should be used when [turtle](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#module-turtle) is used as a standalone tool for doing graphics. As a singleton object, inheriting from its class is not possible.

All methods of TurtleScreen/Screen also exist as functions, i.e. as part of the procedure-oriented interface.

1. [RawTurtle](https://docs.python.org/3.3/library/turtle.html?highlight=turtle" \l "turtle.RawTurtle" \o "turtle.RawTurtle) (alias: [RawPen](https://docs.python.org/3.3/library/turtle.html?highlight=turtle" \l "turtle.RawPen" \o "turtle.RawPen)) defines Turtle objects which draw on a [TurtleScreen](https://docs.python.org/3.3/library/turtle.html?highlight=turtle" \l "turtle.TurtleScreen" \o "turtle.TurtleScreen). Its constructor needs a Canvas, ScrolledCanvas or TurtleScreen as argument, so the RawTurtle objects know where to draw.

Derived from RawTurtle is the subclass [Turtle](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#turtle.Turtle) (alias: Pen), which draws on “the” [Screen](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#turtle.Screen) instance which is automatically created, if not already present.

All methods of RawTurtle / Turtle also exist as functions, i.e. part of the procedure-oriented interface.

1. The procedural interface provides functions which are derived from the methods of the classes [Screen](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#turtle.Screen) and [Turtle](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#turtle.Turtle).
2. They have the same names as the corresponding methods. A screen object is automatically created whenever a function derived from a Screen method is called.
3. An (unnamed) turtle object is automatically created whenever any of the functions derived from a Turtle method is called.

To use multiple turtles on a screen one has to use the object-oriented interface