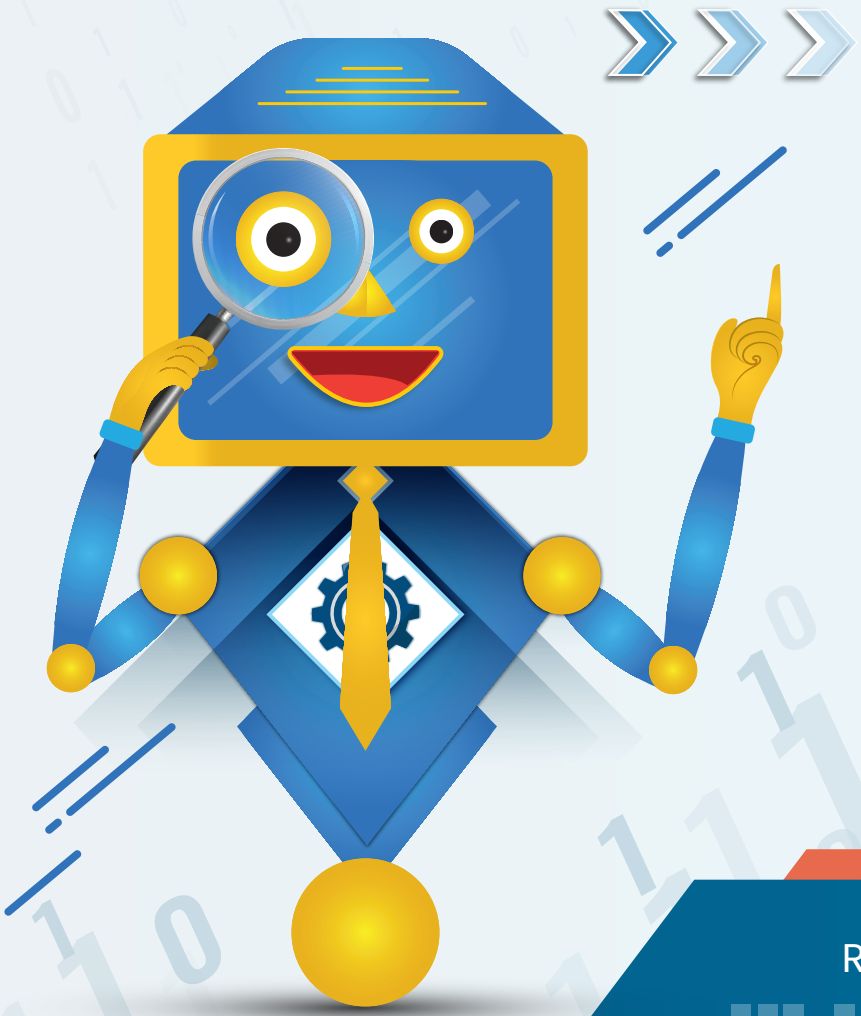
Learning Python

Advanced Level

Python Run

Student's Book

Level 3



Rana Dajani

LKD EDUCATIONAL RESOURCES

PythonRun - Advanced Level

Published by **LKD Educational Resources 2022**

Amman - Jordan
Tel: +962 6 5374141
Fax: +962 6 5516404
P.O.Box: 851346
Email: info@lkd.com.jo
Web: www.lkd.com.jo

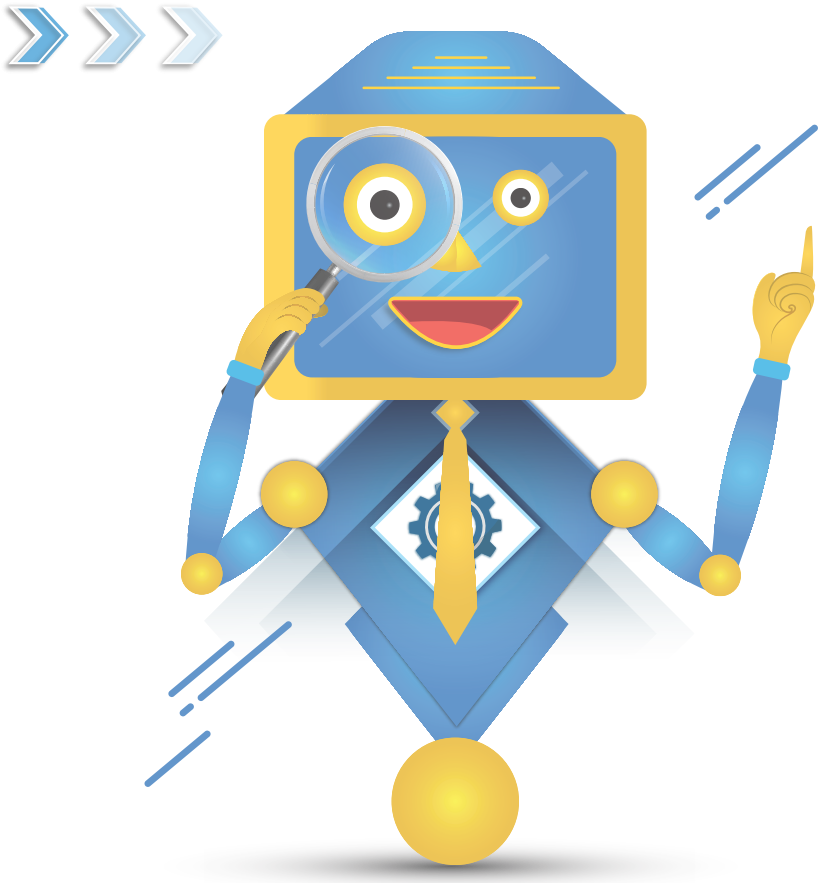
Author

Rana Dajani

ISBN: 978-9923-781-06-7

Learning Python
Advanced Level
Python Run

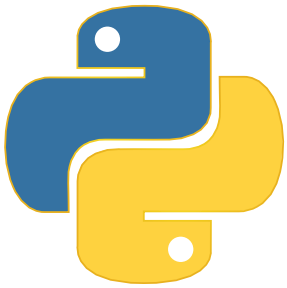
Level
3



LKD
EDUCATIONAL RESOURCES

A guide to learning Python programming language

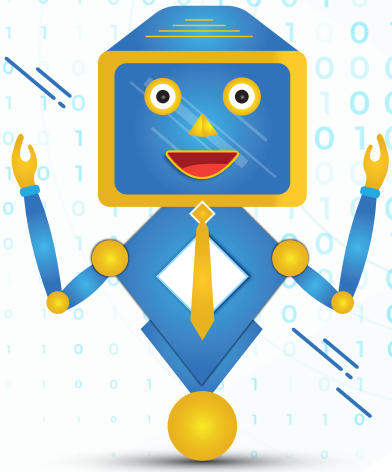
Introduction



This is the adavnced level book in the ***PythonRun*** series, that takes your coding journey to the next level. This book provides an opportunity to consolidate all the previously learnt skills, from the beginner and intermediate level books, in slightly more challenging material.

You will learn how to build graphical user interfaces (GUIs) and other fun visual projects. Futhermore, you will discover essential new programming ideas which will make you a much more accomplished coder.

These new skills may ultimately lead you to write the next big app that sweeps the world!



START CODING NOW! .....

Table of Contents

Unit 1 >> Turtles!

Unit 2 >> Better Turtle Graphics

Unit 3 >> Object Oriented Programming

Table of Contents

Unit 4 >> Reading and Writing Files

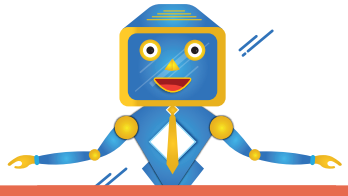


Table of Contents

Unit 5 >> Graphics Tkinter

Table of Contents

Unit 6 >> Tkinter Canvas

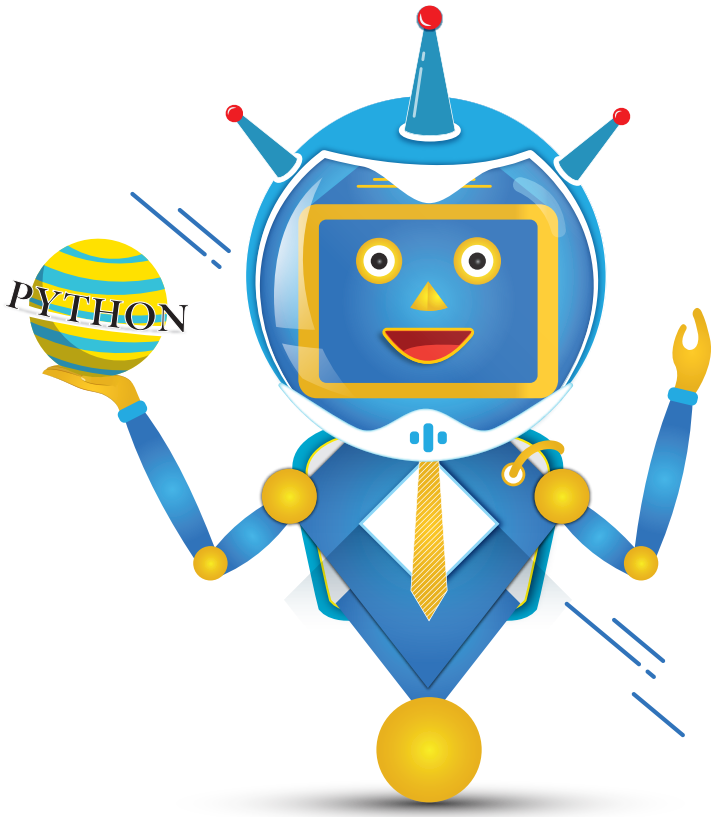
>> Project - Based Assessments

Unit 1 »

» Turtles!

- » Turtle Module
- » Creating a Canvas
- » Moving the turtle
- » Turtle Position
- » Turtle Shape and Stamps
- » Clean Canvas
- » Quick Tests

Turtles!



1.1 Turtle Module

The turtle module is a library that enables users to basically just draw with simple lines, dots, curves, and create pictures and shapes by providing them with a virtual canvas. It is a way of programming vector graphics

```
>>> import turtle as t
```

- The on screen pen that you use for drawing is called the **turtle**.
- The turtle has certain changeable characteristics, like size, colour and speed.

Note:

Make sure you do not have a file of your own lying around named turtle.py as this will interfere with loading Python's own turtle.py library.

1.2 Creating a Canvas

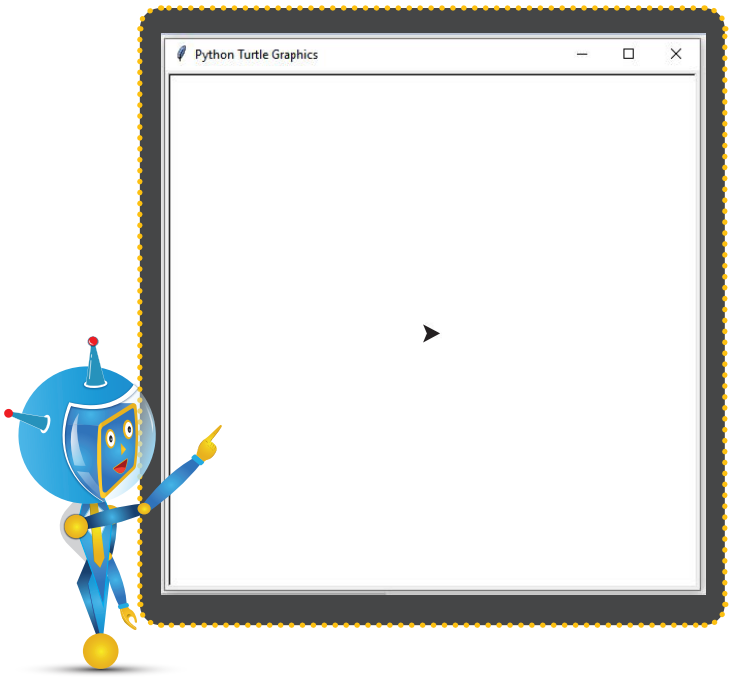
We need to create a canvas—a blank space to draw on, like an artist's canvas.

Pen() function from the turtle module, automatically creates a canvas.

You should see a blank box (the canvas), with an arrow in the center.

```
>>> import turtle as t
```

```
>>> t.pen ()
```



To draw with the turtle pen:

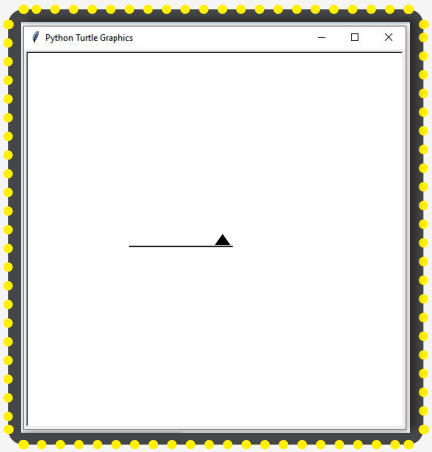
- **up():** means that no line will be drawn when it moves.
- **down():** means that a line will be drawn when it moves.

1.3 Moving the Turtle

There are four directions that a turtle can move in:

- **forward (distance) or .fd()**
- **backward (distance) or .bk()**
- **left (degree) or .lt()**
- **right (degree) or .rt()**

```
>>> import turtle as t
>>> t.pen()
>>> t.down ()
>>> t.forward(50)
>>> t.right(30)
```



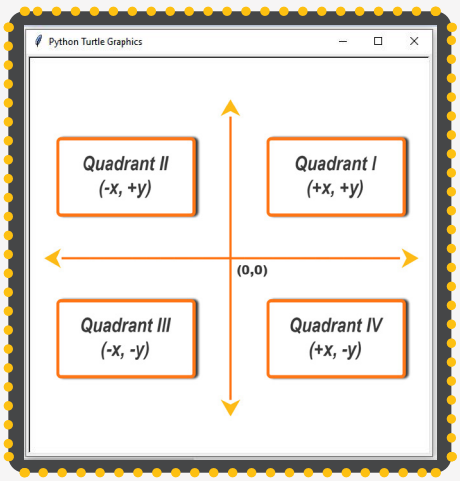
The turtle arrow moves forward 50 pixels on the screen then turns 30 degrees clockwise.

1.4 Turtle Position

The screen is divided into four quadrants. The point where the turtle is initially positioned at the beginning of your program is **t.home (0,0)** . Use **t.pos()** to know the position of the turtle on the screen at any time.

To move the turtle to any other area on the screen, you use **t.goto()** and enter the coordinates like this:

```
>>> import turtle as t
>>> t.up()
>>> t.home()
>>> t.goto(0,20)
>>> t.forward(50)
>>> print(t.pos())
(50,20)
```



- **t.setheading(dergree)**: turns the turtle to face a particular direction.

1.5 Turtle Shape and Stamps

You can change the way the turtle looks.

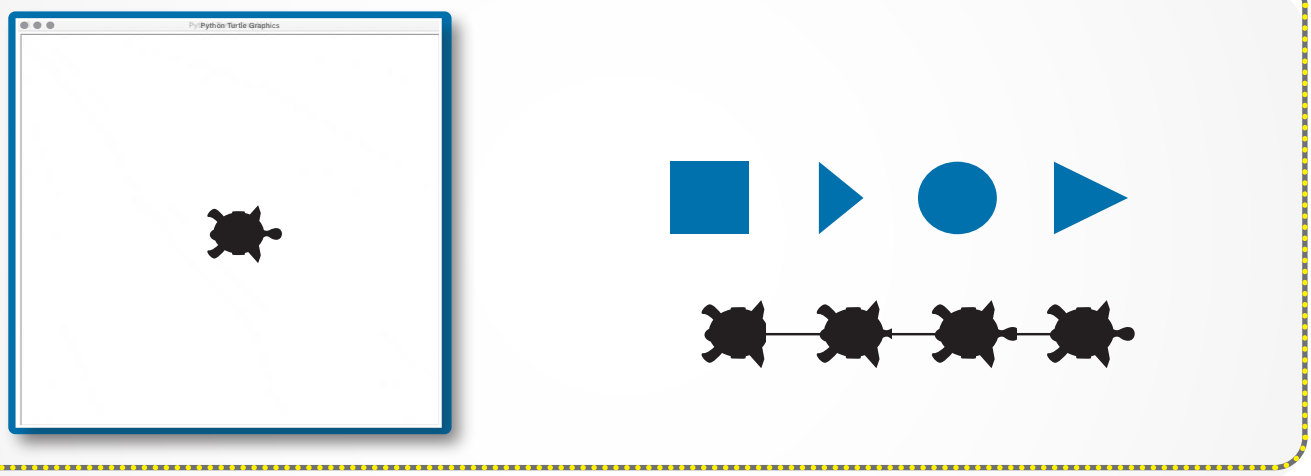
You have a couple of options that you can try :

- Square
- Arrow
- Circle
- Turtle
- Triangle

```
>>> t.shape("turtle")
```

You have the option of leaving a stamp of your turtle on the screen, which is nothing but an imprint of the turtle.

```
>>> for i in range (4):
        t.shape('turtle')
        t.stamp()
        t.fd(150)
```



1.6 Clean Canvas

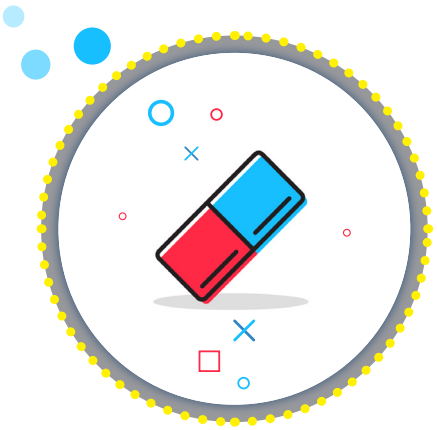
```
>>> import turtle as t
```

To erase the canvas, enter reset. This clears the canvas and puts the turtle back at its starting position.

```
>>> t.reset()
```

You can also use clear, which just clears the screen and leaves the turtle where it is.

```
>>> t.clear()
```



You can undo the last turtle action.

```
>>> t.undo()
```

If you have other turtles on your screen other than the original turtle, then their drawings will not be cleared out unless you specifically call them out in your code.

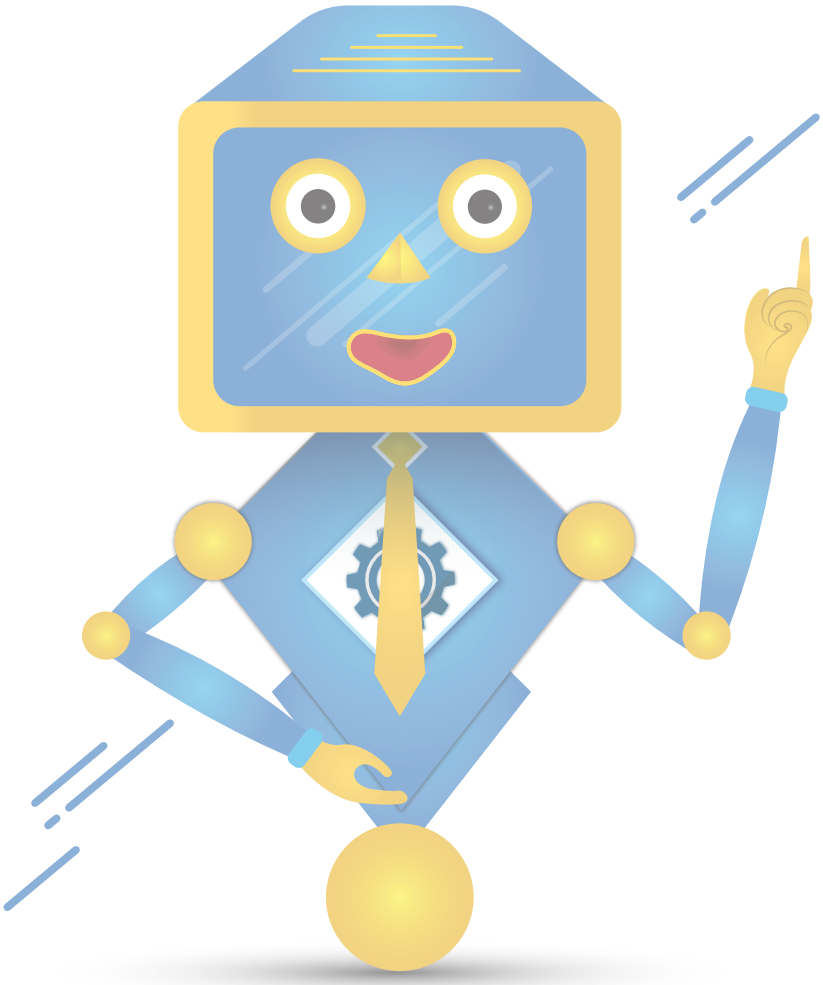
Practice: Turtle Module

Draw a square with the turtle

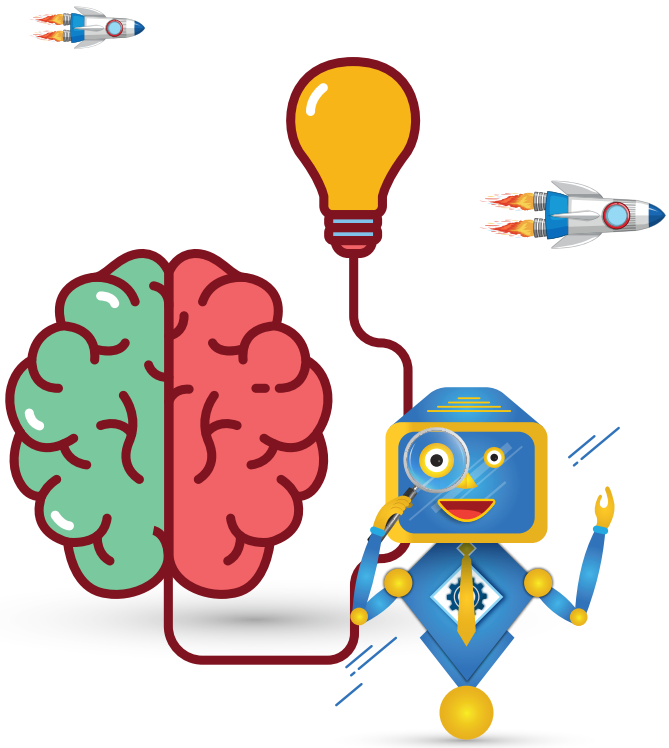
```
>>> import turtle as t
>>> for x in range(4):
        t.forward (100)
        t.left(90)
```

Or

```
>>> t.clear()
>>> t.forward(50)
>>> t.left(120)
>>> t.forward(50)
>>> t.left(120)
>>> t.forward(50)
```

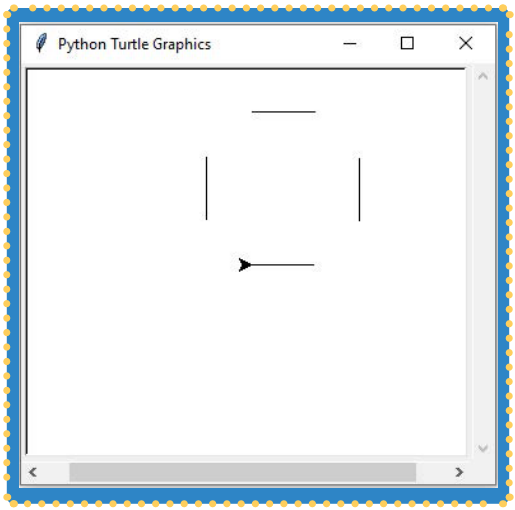


Quick Tests



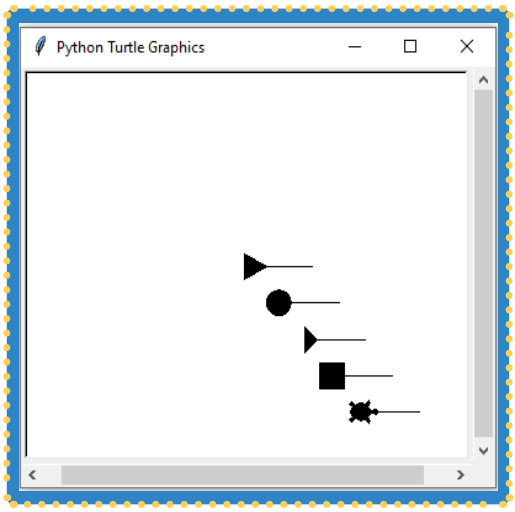
Quick Test: Box Without Corners

Write a program to draw the four lines shown here (the size is not important, just the shape):



Quick Test: Stamping Trail

Write a program to recreate this pattern:




LKD
EDUCATIONAL RESOURCES