قسم الانظمة الطبية الذكية

المرحلة الثانية

Lecture: (8)

Subject: Object oriented programming II
Class: Second
Lecturers: Dr. Dunia H. Hameed , Dr. Maytham N. Meqdad

Object Oriented Programming (II) – Eighth Lecture

1. Graphics Module in Python

Graphics is the most interesting topic to start with while learning any programming language, especially for kids and Graphics-based game developers. The applications of Graphics are endless in the field of programming and computer science. it is also essential for students to have a thorough understanding of Computer Graphics as a separate subject. It is because each pixel and screen resolution, aspect ratios are some essential ingredients that computer graphics cater to in every system.

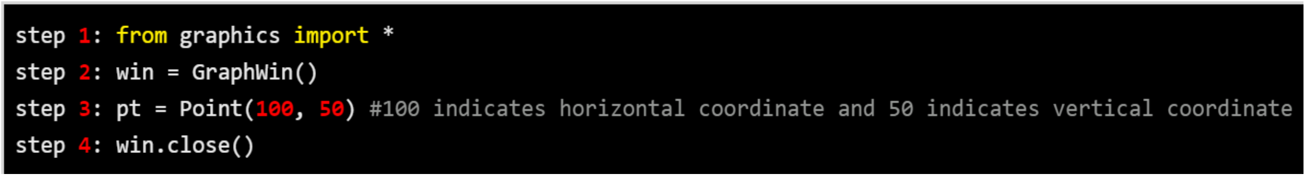
You will learn how the Graphic module in Python allows drawing simple objects using functions such as the GraphWin(), Point(), Circle() and many more. You will be creating a real graphic using these objects as well.

Steps to create graphics

Creating graphics in Python contains a few basic and fixed steps:

1. Import all libraries from the graphics Class
2. Create a Window object
3. Draw the objects in the created window.
4. Close the window.

Example:

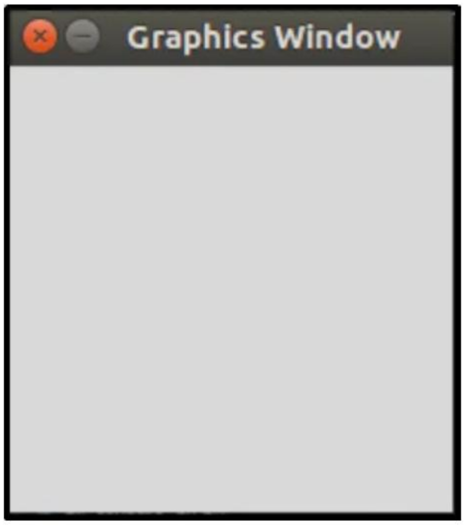


```
step 1: from graphics import *
step 2: win = GraphWin()
step 3: pt = Point(100, 50) #100 indicates horizontal coordinate and 50 indicates vertical coordinate
step 4: win.close()
```

2. Graphic window

You can create a graphics within a window using the **GraphWin()** method which creates a new window somewhere beneath another window. To view it, you can drag it at the top. This window by default is labeled as “Graphics Window” and creates a default **200 X 200** pixels window.

A pixel **(Picture Element)** is nothing but a tiny square displayed on your screen, usually, there are 1000 pixels on the screen and more the number of pixels per row more is the screen resolution and better the image quality will become. It is because each pixel can store more detailed element of the image.

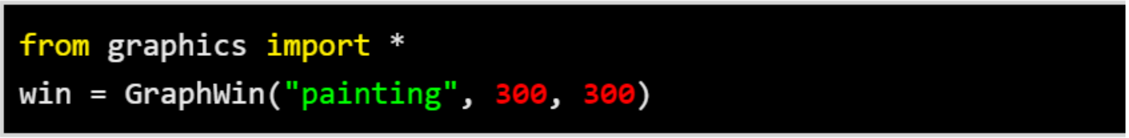


Default graphic window

The GraphWin() method can take three arguments:

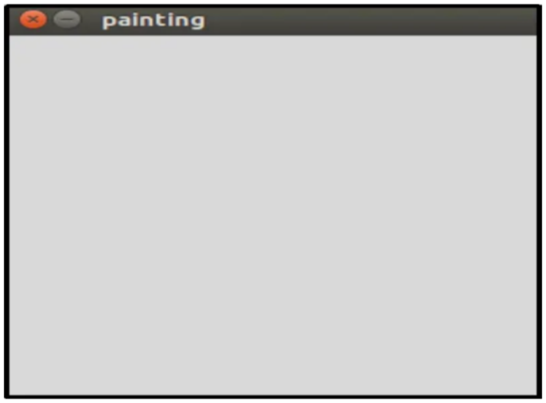
1. Title for the graphic window.
2. X-coordinate (the length of the graphic window).
3. Y-coordinate (the height of the graphic window).

Example: Graphic window with label painting and having specific coordinates



```
from graphics import *
win = GraphWin("painting", 300, 300)
```

Output:

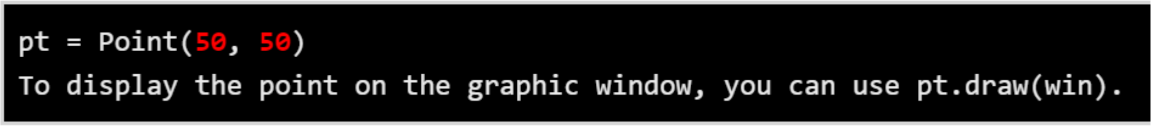


3. Graphic objects

A. Point object

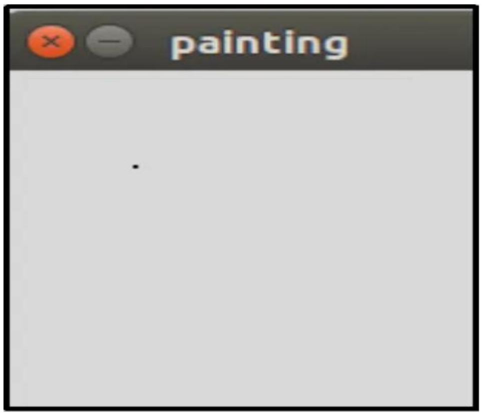
Point: To create a point object you can use the Point() method with X and Y coordinates.

Example:



```
pt = Point(50, 50)
To display the point on the graphic window, you can use pt.draw(win).
```

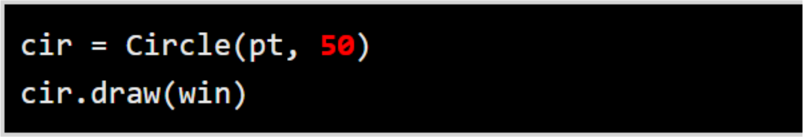
Output:



B. **Circle object**

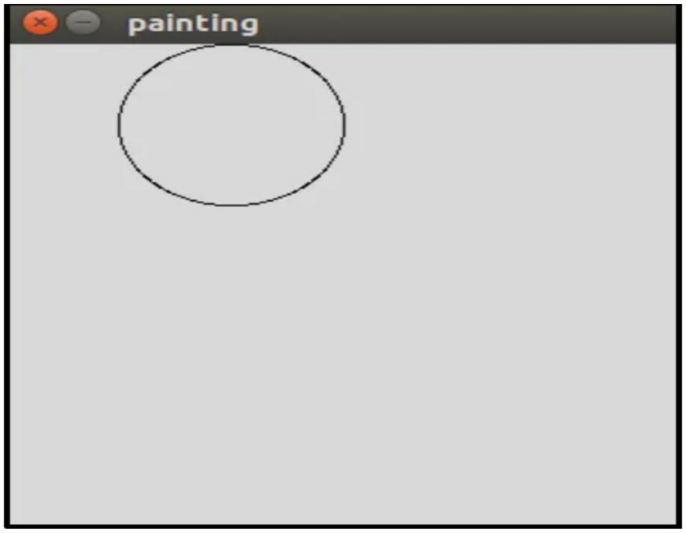
Circle: To create a circle you can use the Circle() method and have a centre at a specific point, pt = Point(50, 50), and a radius of 50. To display the circle on the graphic window use cir.draw(win).

Example:



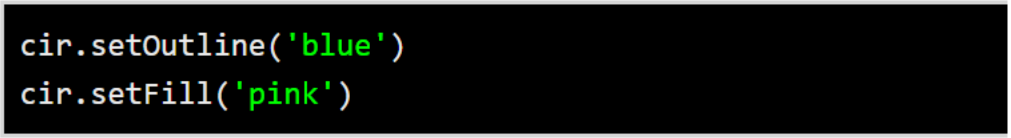
```
cir = Circle(pt, 50)
cir.draw(win)
```

Output:



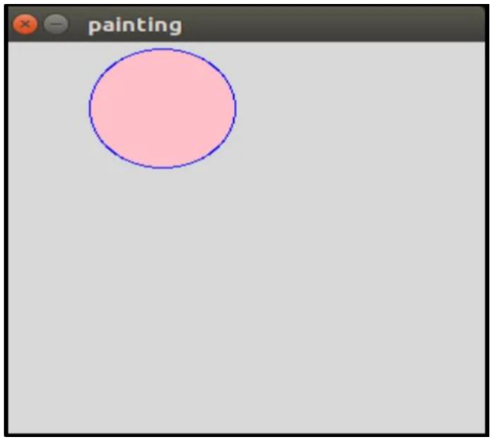
The color of the outline circle is by default black. You can have colors to the outline of the circle as well as within the circle.

Example:



```
cir.setOutline('blue')
cir.setFill('pink')
```

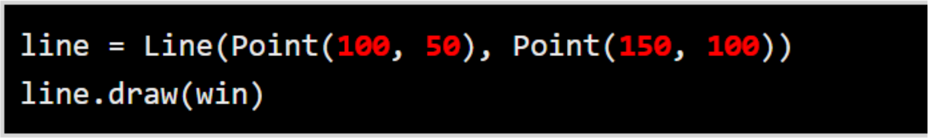
Output:



C. **Line Object**

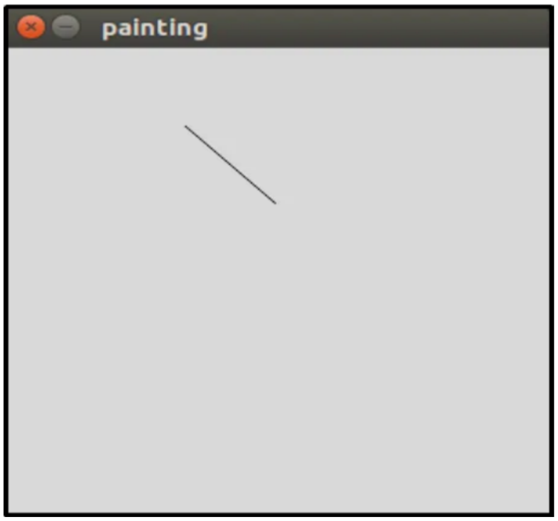
Line: A line is a segment between two points, hence to draw a line you will have to define two points.

Example:



```
line = Line(Point(100, 50), Point(150, 100))
line.draw(win)
```

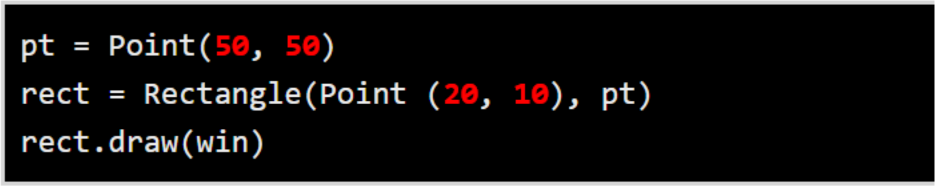
Output:



D. Rectangle Object

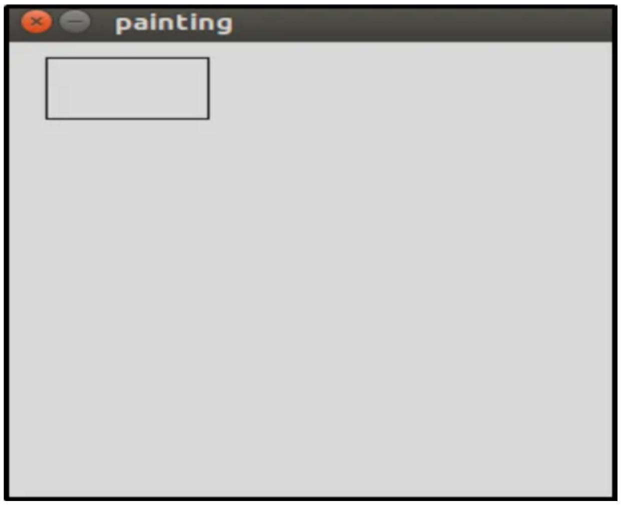
Rectangle: A rectangle is defined using two points and these two points must be diagonally opposite to each other. To display the rectangle on the graphic window use rect.draw(win).

Example:



```
pt = Point(50, 50)
rect = Rectangle(Point (20, 10), pt)
rect.draw(win)
```

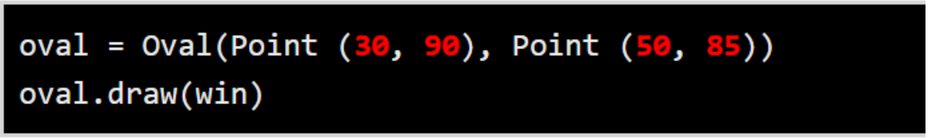
Output:



E. Oval Object

Oval: You can define an oval shape object by providing two points that determine the bounding box.

Example:



```
oval = Oval(Point (30, 90), Point (50, 85))
oval.draw(win)
```

Output: