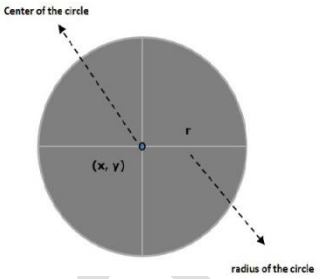
Draw Circle

A circle is the locus of all points at a fixed distance (radius of circle) from a fixed point (the center of circle). In other words, a circle is a line forming a closed loop, every point on which is a fixed distance from a center point.



A circle is defined by two parameters namely −

Centre − It is a point inside the circle. All points on the circle are equidistant (same distance) from the center point.

Radius − The radius is the distance from the center to any point on the circle. It is half the diameter.

In JavaFX, a circle is represented by a class named **Circle**. This class belongs to the package **javafx.scene.shape**. This class has 3 properties of the double datatype namely −

centerX − The x coordinate of the center of a circle.
centerY − The y coordinate of the center of a circle.
radius − The radius of the circle in pixels.

To draw a circle, you need to pass values to these properties, either by passing them to the constructor of this class, in the same order, at the time of instantiation, as follows −

```
Circle circle = new Circle(centerx, centery, radius);
```

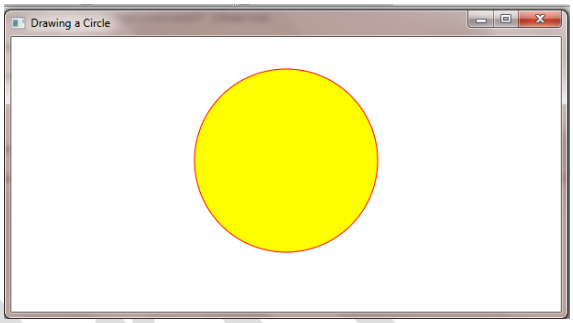
Or, by using their respective setter methods as follows −

```
setCenterX(value);
setCenterY(value);
setRadius(value);
```

Example

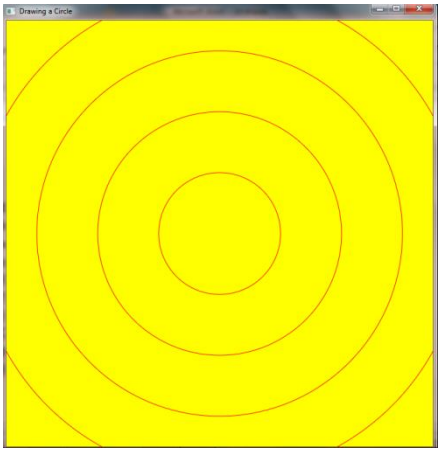
```
package javafxapplication23;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.scene.Group;
import javafx.scene.shape.Circle;
import javafx.scene.paint.Color;

public class JavaFXApplication23 extends Application {

    @Override
    public void start(Stage stage) {
      //Drawing a Circle
      Circle circle = new Circle();
```

```
        //Setting the properties of the circle
        circle.setCenterX(300);
        circle.setCenterY(135);
        circle.setRadius(100);
        circle.setStroke(Color.RED);
        circle.setFill(Color.YELLOW);

        //Creating a Group object
        Group root = new Group(circle);

        //Creating a scene object
        Scene scene = new Scene(root, 600, 300);
        //Setting title to the Stage
        stage.setTitle("Drawing a Circle");

        //Adding scene to the stage
        stage.setScene(scene);

        //Displaying the contents of the stage
        stage.show();
    }
    public static void main(String args[]){
        launch(args);
    }
}
```



Add the following code to obtained the following figure :

```
        for(double i = 700 ; i >=100; i=i-100){
        //Drawing a Circle
        Circle circle = new Circle();
        //Setting the properties of the circle
        circle.setCenterX(500);
        circle.setCenterY(500);
        circle.setRadius(i);
        circle.setStroke(Color.RED);
        circle.setFill(Color.YELLOW);
        root.getChildren().add(circle);
        }
```



Circle Enlarge and Shrink

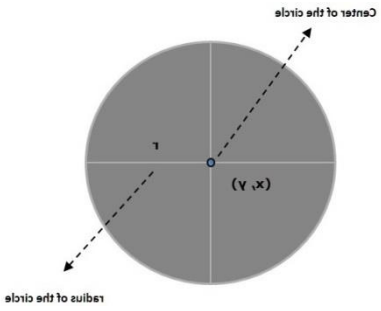
```
package javafxapplication2;
import javafx.application.Application;
```

```
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.*;
import javafx.scene.shape.*;
import javafx.scene.paint.*;
import javafx.geometry.*;
import javafx.event.*;
import javafx.scene.control.Button;

public class JavaFXApplication2 extends Application {
 private circlepane cp = new circlepane();

@Override // Override the start method in the Application class
public void start(Stage primaryStage) {


   HBox hBox = new HBox();
   hBox.setSpacing(5);
   hBox.setAlignment(Pos.CENTER);
   Button btEnlarge = new Button("Enlarge");
   Button btShrink = new Button("Shrink");
   hBox.getChildren().addAll(btEnlarge, btShrink);

   btEnlarge.setOnAction(new EnlargeHandler());
   btShrink.setOnAction(new shrinkHandler());

   BorderPane borderPane = new BorderPane();
   borderPane.setCenter(cp);
   borderPane.setBottom(hBox);
   BorderPane.setAlignment(hBox, Pos.CENTER);

   Scene scene = new Scene(borderPane, 300, 300);
   primaryStage.setTitle("ShowImage"); // Set the stage title
   primaryStage.setScene(scene); // Place the scene in the stage
   primaryStage.show(); // Display the stage
   }

class EnlargeHandler implements EventHandler<ActionEvent>{
  @Override
  public void handle(ActionEvent e){
     cp.enlarge();
  }
}

class shrinkHandler implements EventHandler<ActionEvent>{
  @Override
```

```
  public void handle(ActionEvent e){
     cp.shrink();
  }

}

class circlepane extends StackPane{
  private Circle circle = new Circle(50);

  public circlepane(){
      circle.setStroke(Color.BLACK);
      circle.setFill(Color.BLUE);
      getChildren().add(circle);
}
public void enlarge(){
   circle.setRadius(circle.getRadius()+5);
}

public void shrink(){
   circle.setRadius(circle.getRadius()> 5 ? circle.getRadius()- 5 :circle.getRadius());
}
}
}
```

Draw An Ellipse

An Ellipse is defined by two points, each called a focus. If any point on the Ellipse is taken, the sum of the distances to the focus points is constant. The size of the Ellipse is determined by the sum of these two distances. The sum of these distances is equal to the length of the major axis (the longest diameter of the ellipse). A circle is, in fact, a special case of an Ellipse.



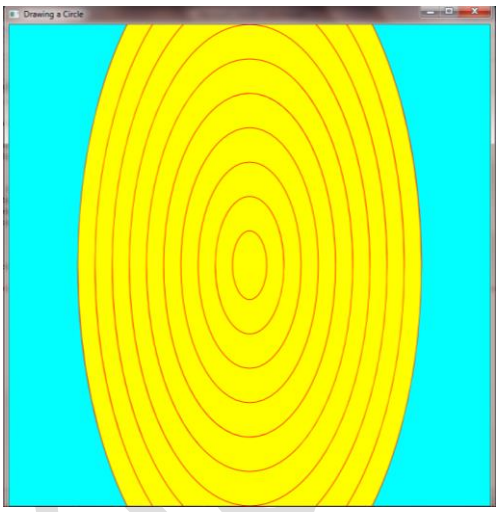
In JavaFX, an Ellipse is represented by a class named **Ellipse**. This class belongs to the package **javafx.scene.shape**.

This class has 4 properties of the double datatype namely −

centerX − The x coordinate of the center of the ellipse in pixels.
centerY − The y coordinate of the center of the ellipse in pixels.
radiusX − The width of the ellipse pixels.
radiusY − The height of the ellipse pixels.

To draw an ellipse, you need to pass values to these properties, either by passing them to the constructor of this class, in the same order, at the time of instantiation, as shown below −

Circle circle = new Circle(centerX, centerY, radiusX, radiusY);



Or, by using their respective setter methods as follows −
setCenterX(value);
setCenterY(value);
setRadiusX(value);
setRadiusY(value);

Example

```
package javafxapplication23;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.scene.Group;
import javafx.scene.shape.*;
import javafx.scene.paint.Color;

public class JavaFXApplication23 extends Application {

   @Override
   public void start(Stage stage) {
    //Creating a Group object
    Group root = new Group();

    for(double i = 250 ; i >=25; i=i-25){
    //Drawing a Circle
     Ellipse e = new Ellipse();
    //Setting the properties of the circle
    e.setCenterX(350);
    e.setCenterY(350);
    e.setRadiusX(i);
    e.setRadiusY(i*2);
    e.setStroke(Color.RED);
    e.setFill(Color.YELLOW);
    root.getChildren().add(e);
    }

    //Creating a scene object
    Scene scene = new Scene(root, 700, 700, Color.CYAN);
    //Setting title to the Stage
    stage.setTitle("Drawing a Circle");

    //Adding scene to the stage
    stage.setScene(scene);
```

```
      //Displaying the contents of the stage
      stage.show();
   }
   public static void main(String args[]){
      launch(args);
   }
}
```

Draw Polygon

A closed shape formed by a number of coplanar line segments connected end to end. A polygon is described by two parameters, namely, the length of its sides and the measures of its interior angles.

In JavaFX, a polygon is represented by a class named **Polygon**. This class belongs to the package **javafx.scene.shape.**

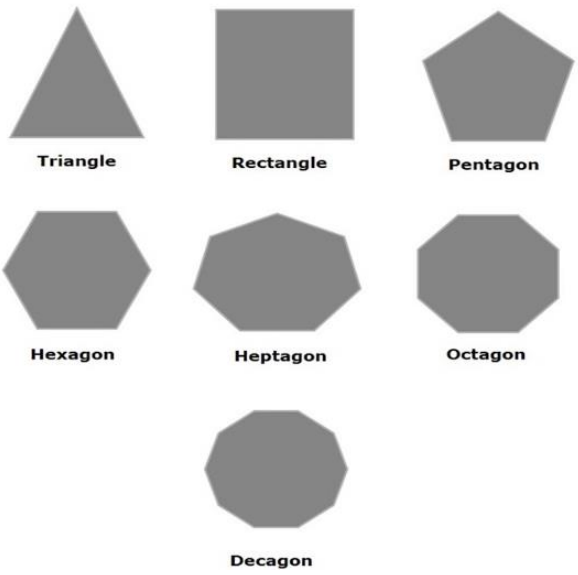
By instantiating this class, you can create a polygon node in JavaFX. You need to pass the x, y coordinates of the points by which the polygon should be defined in the form of a double array.

You can pass the double array as a parameter of the constructor of this class as shown below −

```
Polygon polygon = new Polygon(doubleArray);
```

Or, by using the **getPoints()** method as follows −

```
polygon.getPoints().addAll(new Double[]{ List of XY
coordinates separated by commas });
```




```
package javafxapplication27;

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.shape.Polygon;
import javafx.scene.paint.Color;

public class JavaFXApplication27 extends Application {

   @Override
   public void start(Stage stage) {
     //Creating a Polygon
     Polygon polygon = new Polygon();
```

```
        //Adding coordinates to the polygon
        polygon.getPoints().addAll(new Double[]{
           200.0, 50.0,
           400.0, 50.0,
           450.0, 150.0,
           400.0, 250.0,
           200.0, 250.0,
           150.0, 150.0,       });

        polygon.setStroke(Color.BLACK);
        polygon.setFill(Color.color(Math.random(),Math.random(),Math.random()));

        //Creating a Group object
        Group root = new Group(polygon);

        //Creating a scene object
        Scene scene = new Scene(root, 600, 300);

        //Setting title to the Stage
        stage.setTitle("Drawing a Polygon");

        //Adding scene to the stage
        stage.setScene(scene);

        //Displaying the contents of the stage
        stage.show();
    }
    public static void main(String args[]){
        launch(args);
    }
}
```