

Show me the code !

A circular graphic with a dark blue background, containing several lines of JavaScript code in a light blue, monospaced font. The code is partially cut off by the edge of the circle. The visible code includes variable declarations, array manipulations, and function calls.

```
var a = 0, b = $(  
, b = q(b), b = b.replace(  
b = [], a = [], c = [], a =  
, c) && (c.push(inp_array[a]),  
, b[b.length - 1].c = r(b[b.length  
a.reverse(); b = m(a, " "); -1 < b  
&& a.splice(b, 1); b = m(a, ""); -  
tion q(a) { return a.replace(RegExp  
var c = 0, d = 0; d < b.length; d++)  
a, b) { for (var c = -1, d = 0; d <  
k; } } return c; } funct:  
(b = -1
```

Festival of Imports !

```
import javafx.application.*;  
import javafx.stage.*;  
import javafx.scene.*;  
import javafx.animation.*;  
import javafx.scene.control.*;  
import javafx.scene.input.*;  
import javafx.scene.canvas.*;  
import javafx.scene.paint.*;
```

Application with Start Method

```
GraphicsContext g;
```

```
public class MyApp extends Application {  
    public void start(Stage stage) {  
        Canvas canvas = new Canvas(300,300);  
        g = canvas.getGraphicsContext2D();  
        Group group = new Group(canvas);  
        Scene scene = new Scene(group);  
        stage.setScene(scene);  
        stage.setTitle("My Application");  
        stage.show();  
    }  
}
```

We never get to see the main method !
This is hidden away in the "Application" class

JavaFX needs to do a lot of setup
And doesn't want us interfering with it

The same goes for the constructor method

Anything we do must be in the "Start" method

Drawing Shapes

```
g.setLineWidth(4);
```

```
g.strokeLine(10, 10, 25, 25);
```

```
g.strokeRect(30, 30, 27, 27);
```

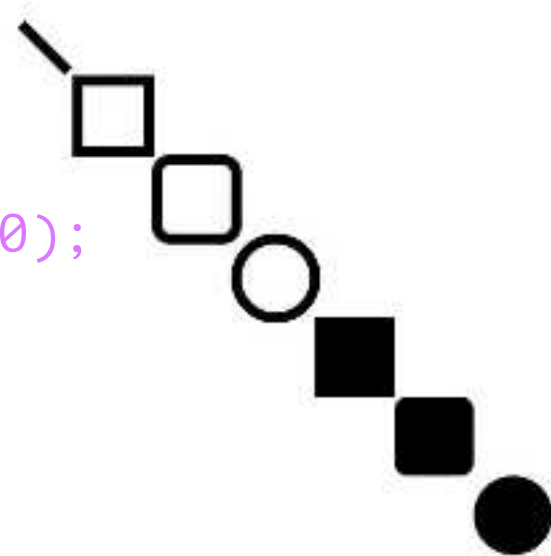
```
g.strokeRoundRect(60, 60, 30, 30, 10, 10);
```

```
g.strokeOval(90, 90, 30, 30);
```

```
g.fillRect(120, 120, 30, 30);
```

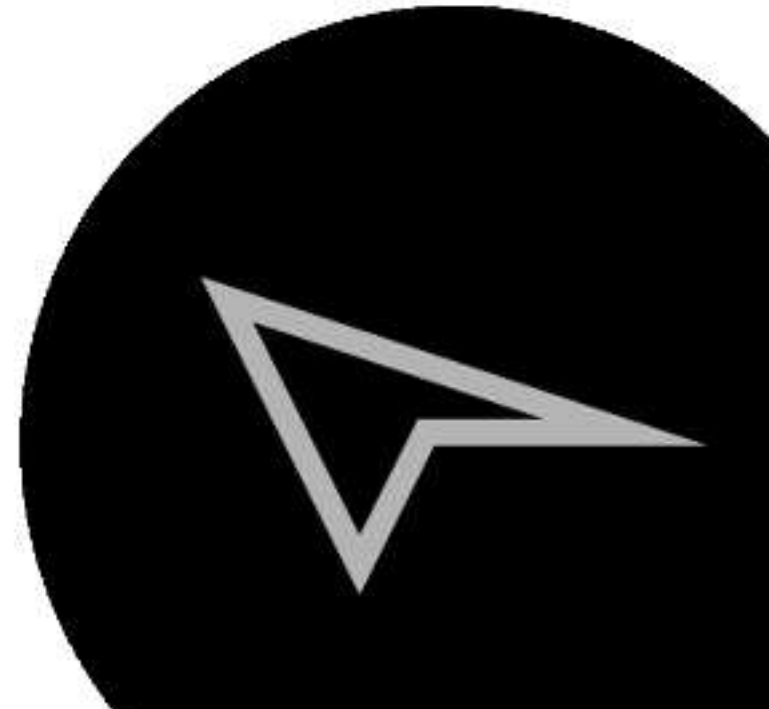
```
g.fillRoundRect(150, 150, 30, 30, 10, 10);
```

```
g.fillOval(180, 180, 30, 30);
```



Drawing Polygons

```
g.setLineWidth(3);  
g.beginPath();  
g.moveTo(30, 20);  
g.lineTo(15, 20);  
g.lineTo(10, 30);  
g.lineTo(00, 10);  
g.lineTo(30, 20);  
g.closePath();  
g.stroke();
```

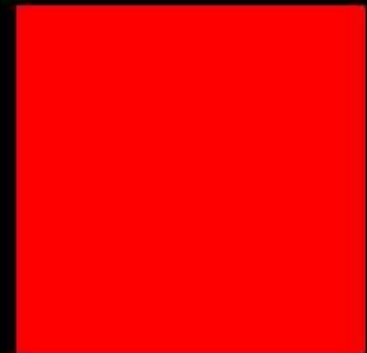
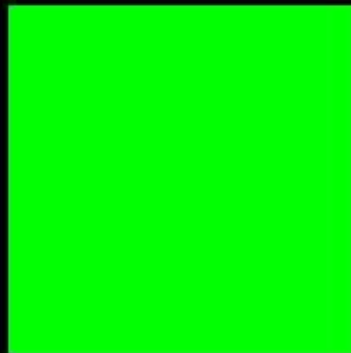
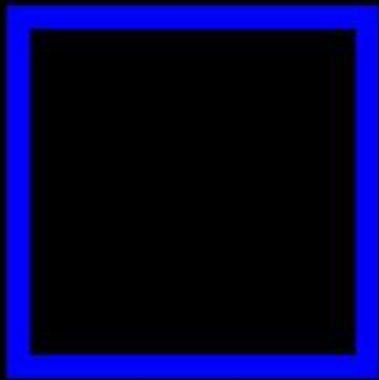


Colours

```
g.setStroke(Color.rgb(0, 0, 255));
```

```
g.setFill(Color.rgb(0, 255, 0));
```

```
g.setFill(Color.hsb(360, 1.0, 1.0));
```



Text

As well as basic shapes JavaFX also renders text:

```
g.strokeText("Hello", 20, 20);  
g.fillText("Hello", 20, 20);
```

We set the colour of text by changing stroke & fill:

```
g.setStroke(Color.rgb(255, 0, 0));  
g.setFill(Color.rgb(0, 255, 0));
```

To change the font, pass in a new font descriptor:

```
Font headingFont = new Font("Arial", 20);  
g.setFont(headingFont);
```


Drawing Images

Various image types can be drawn with JavaFX

```
FileInputStream imageData;  
imageData = new FileInputStream("photo.jpg")  
Image img = new Image(imageData);
```

```
int x = 100;  
int y = 100;  
g.drawImage(img, x, y);
```

```
int width = 20;  
int height = 30;  
g.drawImage(img, x, y, width, height);
```