Drawing with Opencv

Here are the Steps

1) Initialize our Canvas as a 300 x 300 Pixel image with 3 Channels Red, Blue, Green with a **Black** background.

```
Canvas = np.zeros( (300,300,3), dtype = "uint8")
```

2) Now lets draw a diagonal line from Top left to Bottom Right of Green

```
              B  G  R
green     = ( 0, 255, 0)
cv2. line =     ( Canvas, (0,0), (300,300), green)
cv2. inshow ( "Canvas", Canvas)
cv2. waitkey (0)
```

The line which we draw is by default 1 pixel thick.

Now lets say that we want to draw 3 pixel thickness

```
             B  G  R           Position
red      = ( 0, 0, 255 )  ⏞⏞⏞⏞⏞⏞⏞⏞⏞⏞⏞⏞⏞⏞⏞
cv2. line =     ( Canvas, (300,0), (0,300) ,red, 3)
cv2. inshow ( "Canvas", Canvas)              ↑   ↑
cv2. waitkey (0)                          Color  Thickness
```

The above one will draw a line from top right to bottom left corner

3) Now lets say we want to draw a rectangle on the same image. But starting position → 10X10

Ending Position → 60 X 60. Color → Green.

```
CV2. rectangle( Canvas, (10, 10), (60, 60), green)
CV2. imshow( "Canvas", Canvas)
CV2. waitKey(0)
```

Therefore → 50X50 is a Square

Similarly, we can draw with thickness and fill the Rectangle/space.

Thickness

```
CV2. rectangle( Canvas, (10, 10), (60, 60), green, 3, -1)
CV2. imshow( "Canvas", Canvas)
CV2. waitKey(0)
```

Fill

4) Now lets see how to draw circles

For that first we need to reinitialize our Canvas as an empty array, then compute the Center (x,y) Co-ordinate of the Canvas.

```
Canvas = np.zeros((300, 300, 3), dtype = "uint8")
(CenterX, CenterY) = (Canvas.Shape[1] // 2,
                      Canvas.Shape[0] // 2)
White = (255, 255, 255)
```

→ This is nothing but we are trying to compute the Center of Co-ordinates

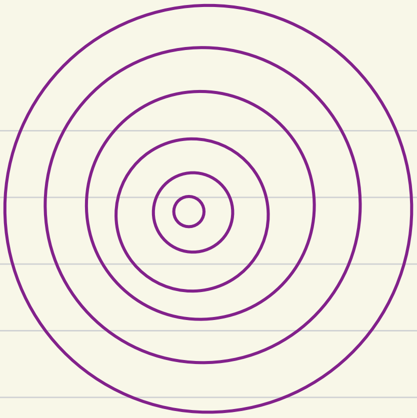
Now we have the Center and by using the radius Starting with 25 pixels to 150 pixels by incrementing 25 pixels, we will be able to draw multiple circles.

```
for r in range(0, 175, 25):
    Cv2.circle(Canvas, (Center X, Center Y), r,
                                              White)

Cv2.imShow("Canvas", Canvas)
Cv2.waitKey(0)
```

When we draw like this → we will get it as Bulls eye affect / Dart Board

Like this we will get the output



5) Now lets draw random circles

```
Canvas = np.zeros((300,300,3), dtype="uint8")
for i in range(0,25):
```

We will generate a radius size between 5 and 200, generate a random color and pick a random point on our Canvas to draw the circle.

```
radius = np.random.randint(5, high=200)
Color = np.random.randint(0, high=256,
                          size=(3,)).tolist()
Pt = np.random.randint(0, high=300, size=(2,))
```

→ It means we need 3 values ranging between 0 and 255 and we are converting that from numpy array to list

If we don't convert to list → openCV throws an error

→ Similarly, we need two points of Canvas (300, 300)

```
cv2. circle (canvas, tuple(pt), radius, color, -1)

cv2. imshow("canvas", canvas)
cv2. waitKey(0)
```

6) Till now what we seen was, we were creating our own Canvas and then we were drawing it.

What if we want to draw on a specific image?

Instead of Canvas, just pass the image Variable.

Rest everything will be remaining the same.