

Dibujando con OpenCV

Importar las librerías

```
In [1]: 1 # Importar las Librerías necesarias
2 from matplotlib import pyplot as plt
3 import numpy as np
4 import cv2
```

Función para mostrar imágenes en Jupyter Notebooks y Google Colab

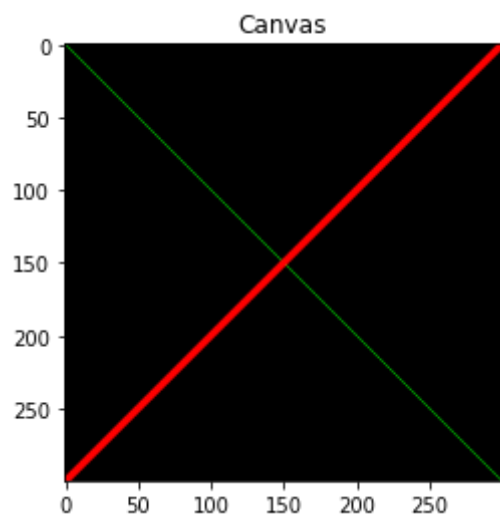
```
In [2]: 1 def mostrar_imagen(titulo,imagen):
2     imagen = cv2.cvtColor(imagen,cv2.COLOR_BGR2RGB)
3     plt.imshow(imagen)
4     plt.title(titulo)
5     plt.grid(False)
6     plt.show
```

Implementando funciones de dibujo básicas en OpenCV

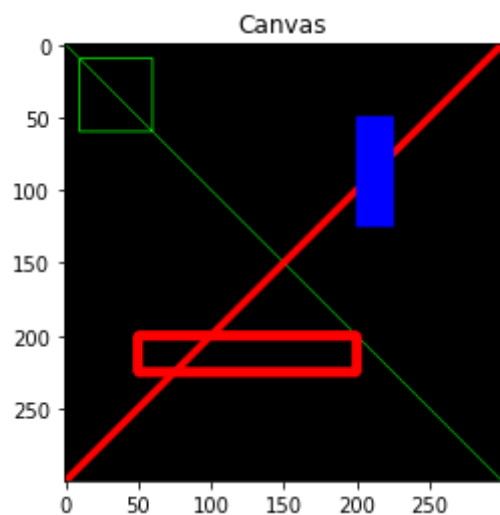
```
In [4]: 1 # inicializar el Lienzo con una imagen de 300x300 pixeles con 3
2 # canales (Rojo, Verde y Azul) con un fondo negro
3 canvas = np.zeros((300, 300, 3), dtype="uint8")
```

In [6]:

```
1 # dibujar una línea verde desde la esquina superior izquierda del
2 # lienzo hasta la esquina inferior derecha
3 green = (0, 255, 0)
4 cv2.line(canvas, (0, 0), (300, 300), green)
5 mostrar_imagen("Canvas", canvas)
6
7 # dibujar una línea roja con 3 píxeles de grosor desde la esquina
8 # superior derecha a la esquina inferior izquierda
9 red = (0, 0, 255)
10 cv2.line(canvas, (300, 0), (0, 300), red, 3)
11 mostrar_imagen("Canvas", canvas)
```

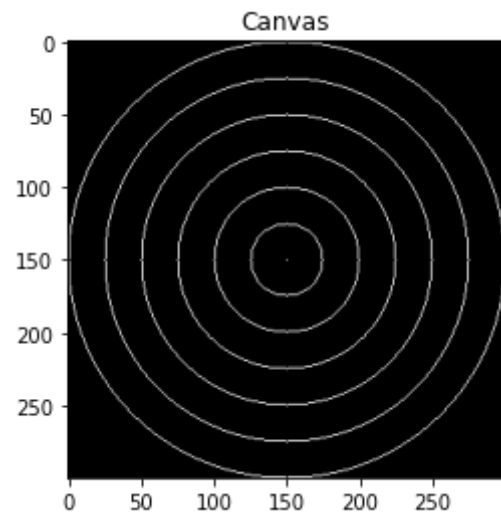


```
In [7]: 1 # dibujar un cuadrado verde de 50x50 pixeles, comenzando en 10,10 y
2 # terminando en 60,60
3 cv2.rectangle(canvas, (10, 10), (60, 60), green)
4 mostrar_imagen("Canvas", canvas)
5
6 # dibujar otro rectángulo, uno rojo con 5 pixeles de grosor
7 cv2.rectangle(canvas, (50, 200), (200, 225), red, 5)
8 mostrar_imagen("Canvas", canvas)
9
10 # dibujar un rectángulo y rellenarlo (azúl y rellenado )
11 blue = (255, 0, 0)
12 cv2.rectangle(canvas, (200, 50), (225, 125), blue, -1)
13 mostrar_imagen("Canvas", canvas)
```



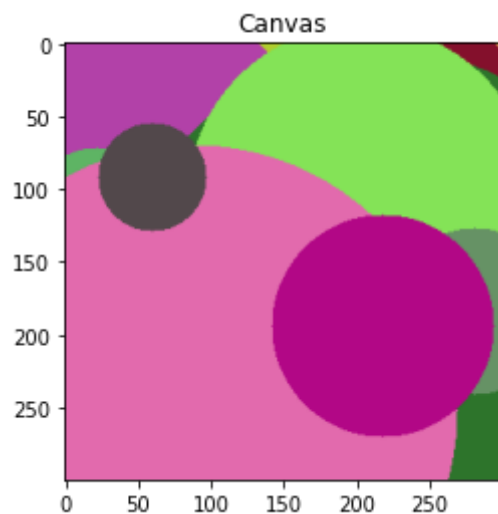
In [8]:

```
1 # reinicializar el canvas como un arreglo vacío, calcular el
2 # centro (x,y) del canvas
3 canvas = np.zeros((300, 300, 3), dtype="uint8")
4 (centerX, centerY) = (canvas.shape[1] // 2, canvas.shape[0] // 2)
5 white = (255, 255, 255)
6
7 # bucle para incrementar el radio, desde 25 pixeles hasta 150 pixeles
8 # en incrementos de 25
9 for r in range(0, 175, 25):
10     # draw a white circle with the current radius size
11     cv2.circle(canvas, (centerX, centerY), r, white)
12
13 # mostrar el dibujo
14 mostrar_imagen("Canvas", canvas)
```



In [9]:

```
1  # reinicializar el canvas nuevamente
2  canvas = np.zeros((300, 300, 3), dtype="uint8")
3
4  # dibujemos 25 circulos
5  for i in range(0, 25):
6      # generar un radio aleatorio entre 5 y 200, generar un color
7      # aleatorio, y luego escoger aleatoriamente un punto en el
8      # lienzo donde sera dibujado el círculo
9      radius = np.random.randint(5, high=200)
10     color = np.random.randint(0, high=256, size=(3,)).tolist()
11     pt = np.random.randint(0, high=300, size=(2,))
12
13     # dibujar el circulo aleatorio
14     cv2.circle(canvas, tuple(pt), radius, color, -1)
15
16 # mostrar el dibujo
17 mostrar_imagen("Canvas", canvas)
```



Dibujando en las imágenes con OpenCV

```
In [13]: 1 args = {  
2         "image": "golden_retriever_4meses.jpg"  
3     }
```

```
In [14]: 1 # cargar la imagen  
2 image = cv2.imread(args["image"])  
3  
4 # dibujar un circulo, dos circulos rellenos y un rectángulo  
5 cv2.circle(image, (168, 188), 90, (0, 0, 255), 2)  
6 cv2.circle(image, (150, 164), 10, (0, 0, 255), -1)  
7 cv2.circle(image, (192, 174), 10, (0, 0, 255), -1)  
8 cv2.rectangle(image, (134, 200), (186, 218), (0, 0, 255), -1)  
9  
10 # mostrar la imagen  
11 mostrar_imagen("Output", image)
```

