02-Drawing-on-Images

August 13, 2025

1 Drawing on Images

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import cv2
```

[2]:
```
blank_img = np.zeros(shape=(512,512,3),dtype=np.int16)
```

[3]:
```
B = blank_img
C = blank_img
```

[4]:
```
blank_img.shape
```

[4]:
```
(512, 512, 3)
```

[5]:
```
a = np.array(blank_img)
```

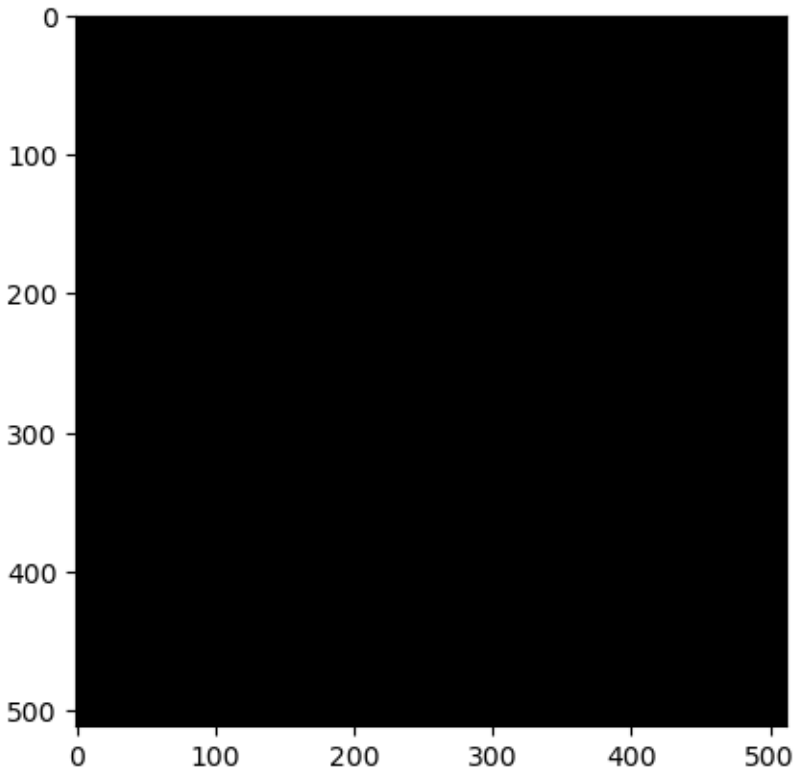
[6]:
```
a
```

[6]:
```
array([[[0, 0, 0],
        [0, 0, 0],
        [0, 0, 0],
        …,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]],

       [[0, 0, 0],
        [0, 0, 0],
        [0, 0, 0],
        …,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]],

       [[0, 0, 0],
```

```
        [0, 0, 0],
        [0, 0, 0],
        …,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]],

       …,

       [[0, 0, 0],
        [0, 0, 0],
        [0, 0, 0],
        …,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]],

       [[0, 0, 0],
        [0, 0, 0],
        [0, 0, 0],
        …,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]],

       [[0, 0, 0],
        [0, 0, 0],
        [0, 0, 0],
        …,
        [0, 0, 0],
        [0, 0, 0],
        [0, 0, 0]]], dtype=int16)
```

[7]:
```
plt.imshow(blank_img)
```

[7]:
```
<matplotlib.image.AxesImage at 0x72baa2b83790>
```



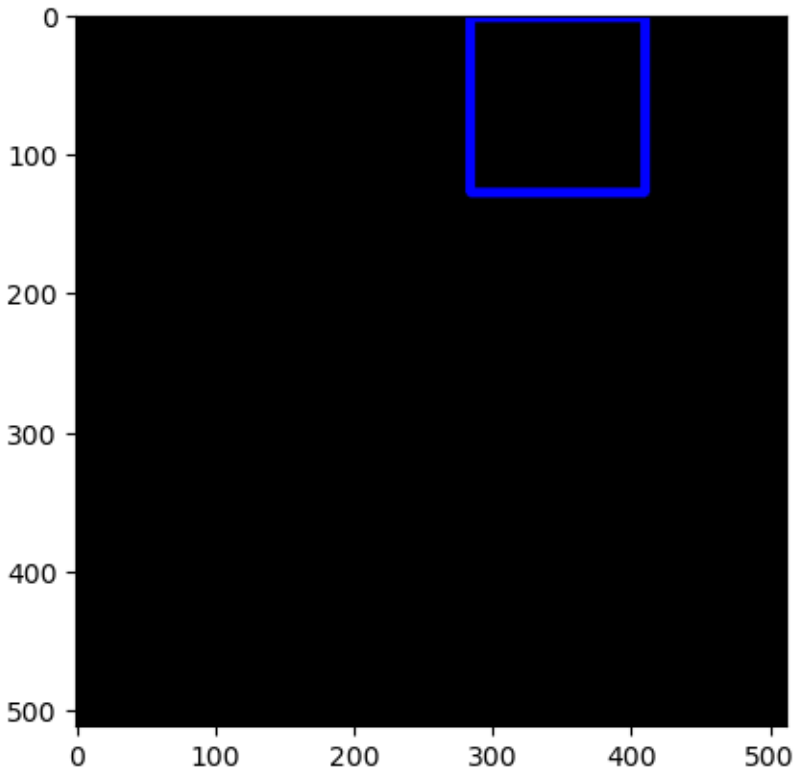
Shapes

1.0.1 Rectangles

- img Image.
- pt1 Vertex of the rectangle.
- pt2 Vertex of the rectangle opposite to pt1 .
- color Rectangle color or brightness (grayscale image).
- thickness Thickness of lines that make up the rectangle. Negative values, like #FILLED,mean that the function has to draw a filled rectangle.
- lineType Type of the line. See #LineTypes
- shift Number of fractional bits in the point coordinates.

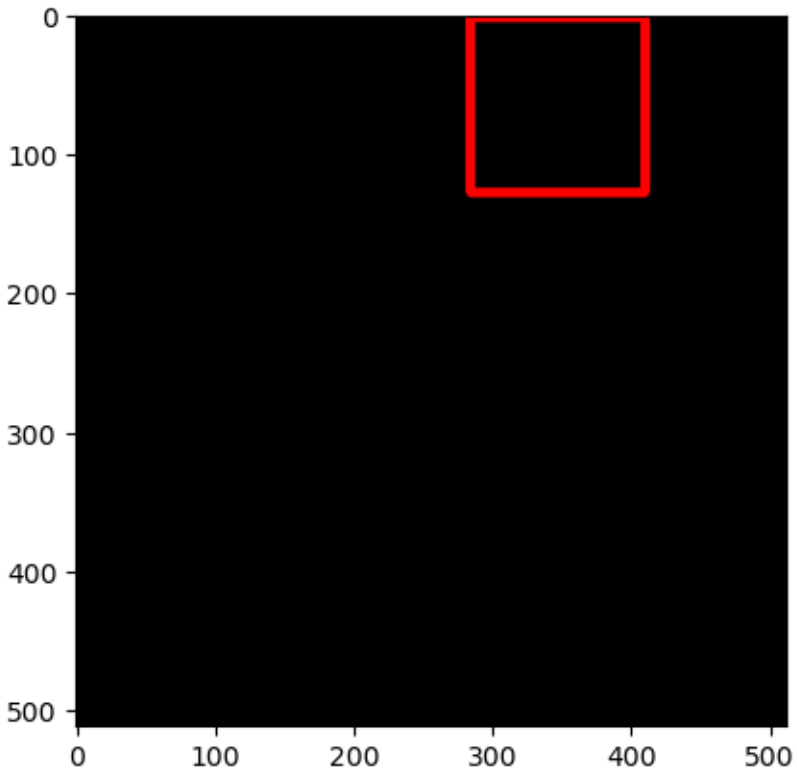
[8]:
```
cv2.rectangle(B,pt1=(284,2),pt2=(410,128),color=(0,0,255),thickness=5)
plt.imshow(B)
```

[8]:
```
<matplotlib.image.AxesImage at 0x72baa2beff10>
```



[9]:
```
cv2.rectangle(B,pt1=(284,2),pt2=(410,128),color=(255,0,0),thickness=5)
plt.imshow(B)
```

[9]: <matplotlib.image.AxesImage at 0x72ba77bf32d0>



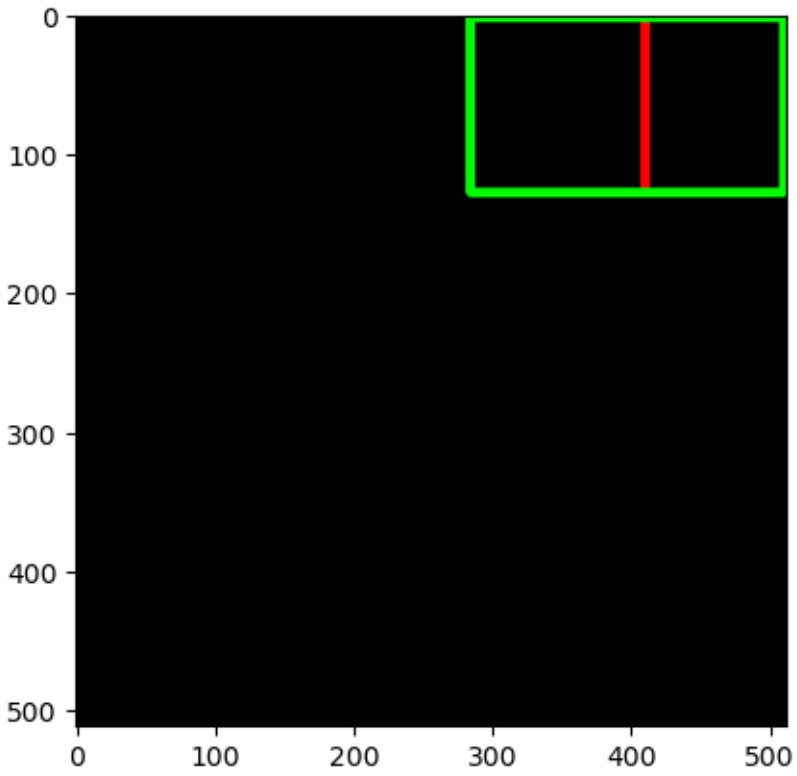
[10]:
```
# pt1 = top left
# pt2 = bottom right
cv2.rectangle(blank_img,pt1=(284,2),pt2=(510,128),color=(0,255,0),thickness=5)
```

[10]:
```
array([[[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        …,
        [  0, 255,   0],
        [  0, 255,   0],
        [  0, 255,   0]],

       [[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        …,
        [  0, 255,   0],
        [  0, 255,   0],
        [  0, 255,   0]],

       [[  0,   0,   0],
        [  0,   0,   0],
```

```
        [  0,   0,   0],
        …,
        [  0, 255,   0],
        [  0, 255,   0],
        [  0, 255,   0]],

       …,

       [[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        …,
        [  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        …,
        [  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        …,
        [  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0]]], dtype=int16)
```

[11]:
```
# cv2.rectangle(blank_img,pt1=(384,0),pt2=(510,128),color=(0,255,0))
plt.imshow(blank_img)
```

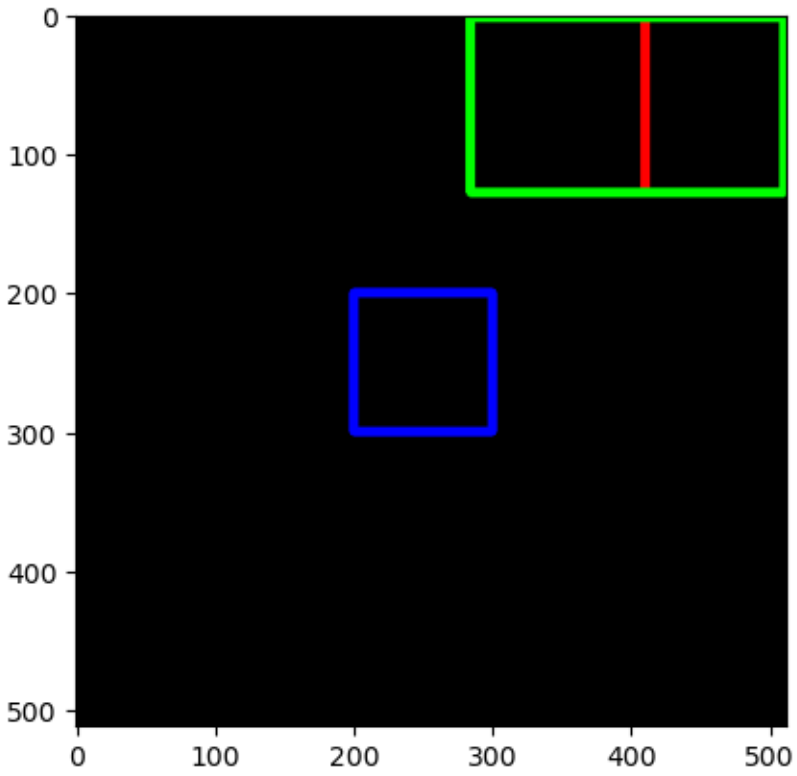
[11]:
```
<matplotlib.image.AxesImage at 0x72ba7671b2d0>
```



Let's practice by drawing a blue rectangle in the middle of the image.

[12]:
```
# pt1 = top left
# pt2 = bottom right
cv2.rectangle(blank_img,pt1=(200,200),pt2=(300,300),color=(0,0,255),thickness=5)
plt.imshow(blank_img)
```

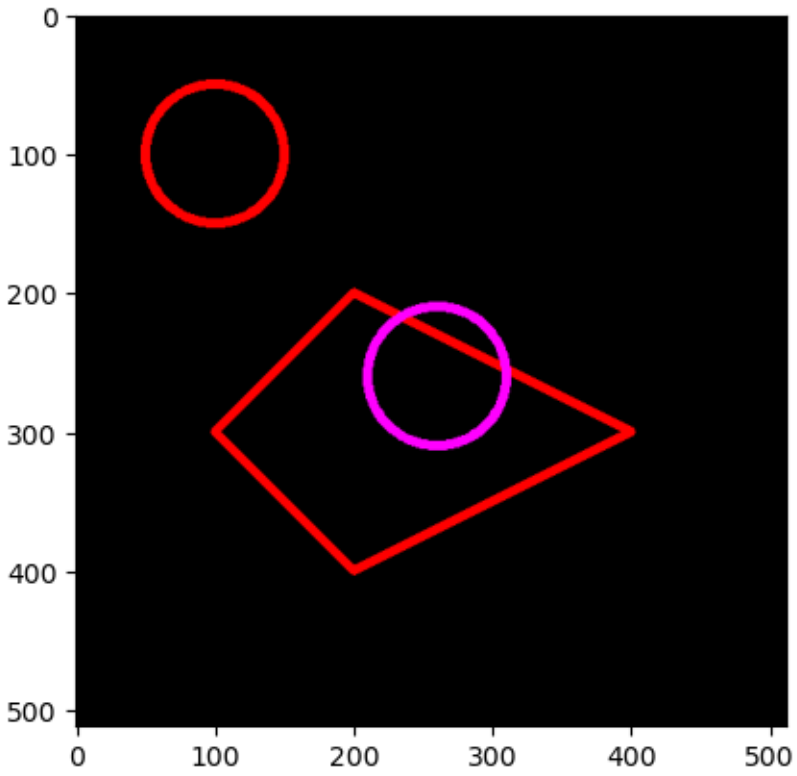
[12]:
```
<matplotlib.image.AxesImage at 0x72ba767932d0>
```



1.1 Circles

[23]:
```
cv2.circle(img=blank_img, center=(260,260), radius=50, color=(255,0,255),␣
↪thickness=5)
plt.imshow(blank_img)
```

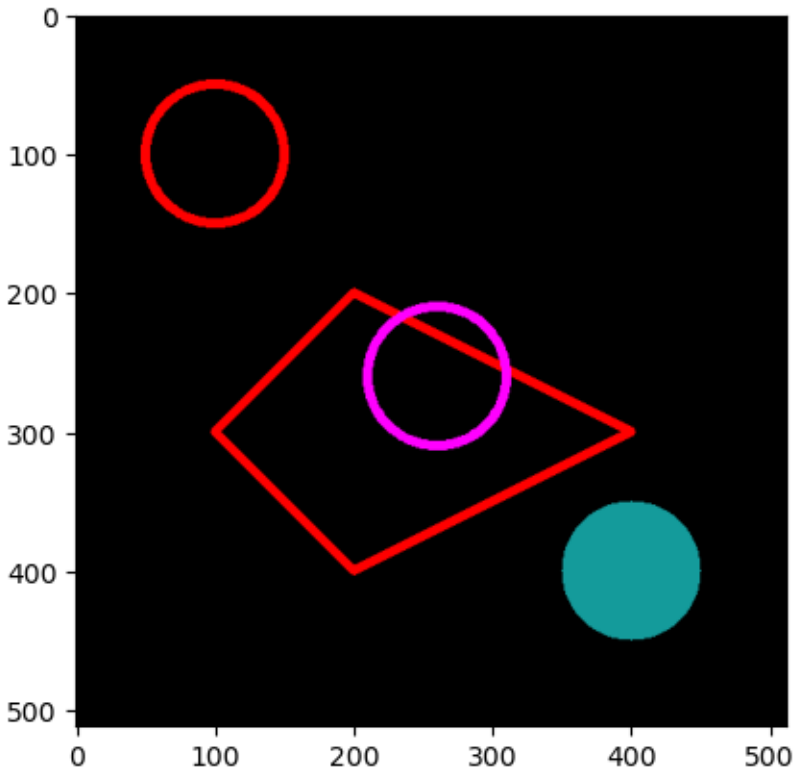
[23]:
```
<matplotlib.image.AxesImage at 0x72ba72dcb2d0>
```



1.1.1 Filled In

[25]:
```
cv2.circle(img=blank_img, center=(400,400), radius=50, color=(20,155,155),␣
↪thickness=-1)
plt.imshow(blank_img)
```

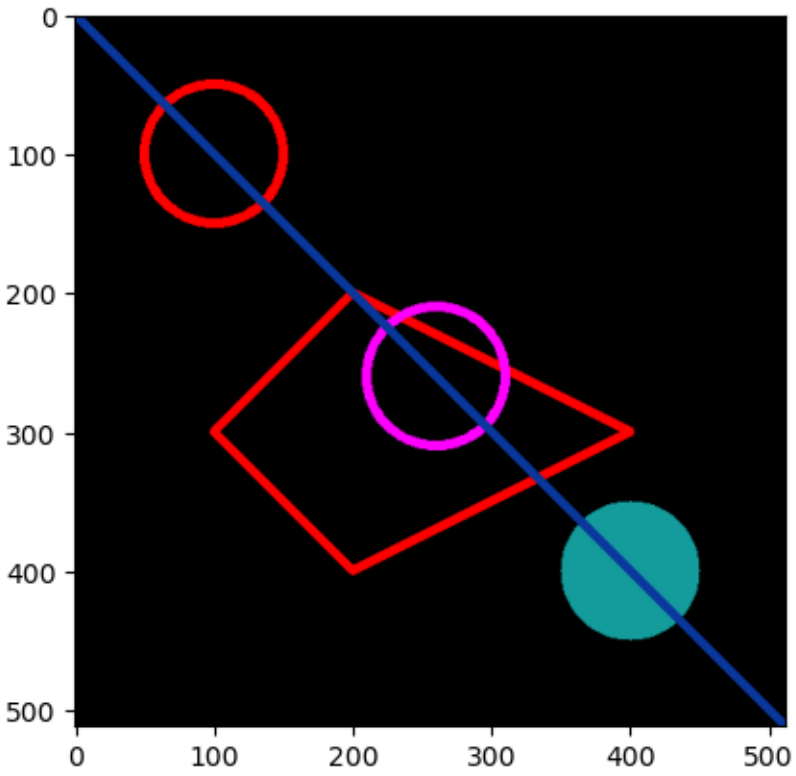
[25]:
```
<matplotlib.image.AxesImage at 0x72ba72d5ff10>
```



1.1.2 Lines

[26]:
```
# Draw a diagonal blue line with thickness of 5 px
cv2.line(blank_img,pt1=(0,0),pt2=(511,511),color=(10, 55, 155),thickness=5)
plt.imshow(blank_img)
```

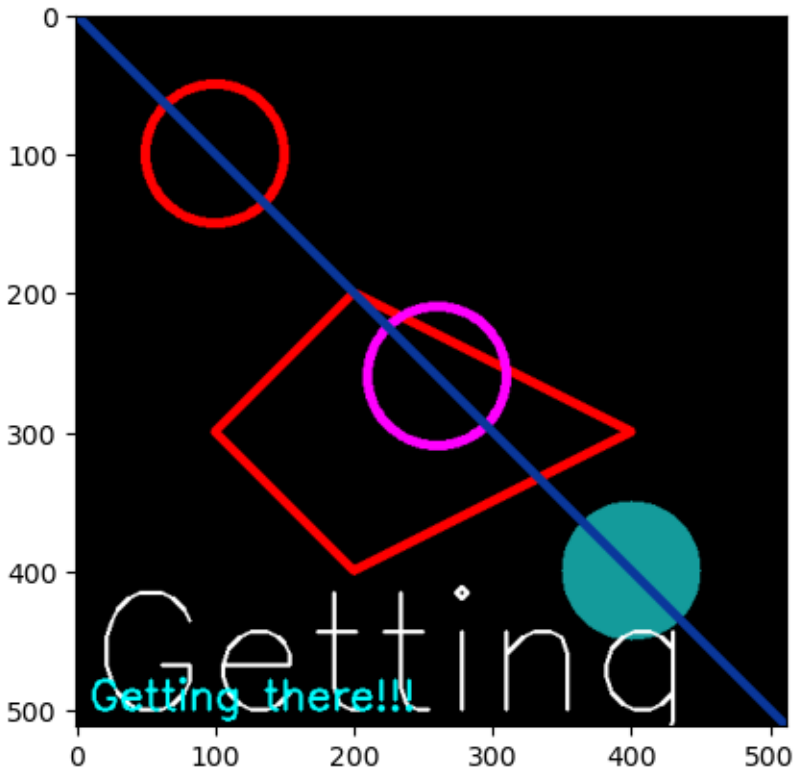
[26]:
```
<matplotlib.image.AxesImage at 0x72ba72728650>
```



1.1.3 Text

[30]:
```
font = cv2.FONT_ITALIC
cv2.putText(blank_img,text='Getting there!!!',org=(10,500),␣
 ↪fontFace=font,fontScale= 1,color=(0,255,255),thickness=2,lineType=cv2.
 ↪LINE_AA)
plt.imshow(blank_img)
```

[30]:
```
<matplotlib.image.AxesImage at 0x72ba7186c5d0>
```



1.1.4 Polygons

To draw a polygon, first you need coordinates of vertices. Make those points into an array of shape ROWSx1x2 where ROWS are number of vertices and it should be of type int32.

[31]:
```
blank_img1 = np.zeros(shape=(512,512,3),dtype=np.int32)
```

[33]:
```
vertices = np.array([[100,300],[200,200],[400,300],[200,400]],np.int32)
```

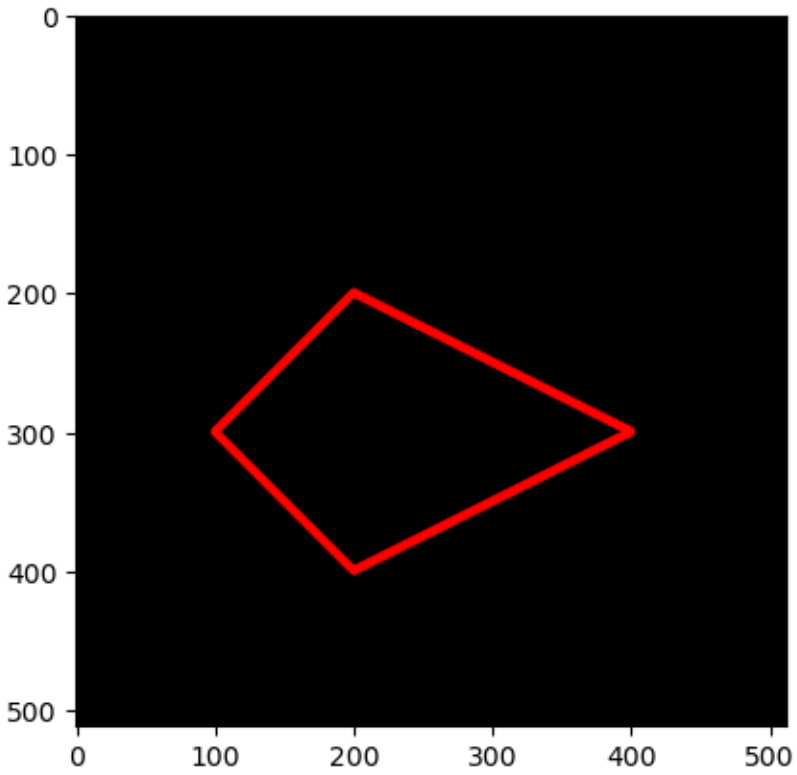
[34]:
```
pts = vertices.reshape((-1,1,2))
```

[35]:
```
pts
```

[35]:
```
array([[[100, 300]],

       [[200, 200]],

       [[400, 300]],

       [[200, 400]]], dtype=int32)
```

[36]:
```
cv2.polylines(blank_img1,[pts],isClosed=True,color=(255,0,0),thickness=5)
plt.imshow(blank_img1)
```

[36]: <matplotlib.image.AxesImage at 0x72ba7194b2d0>



Play around with this! Add shapes.