

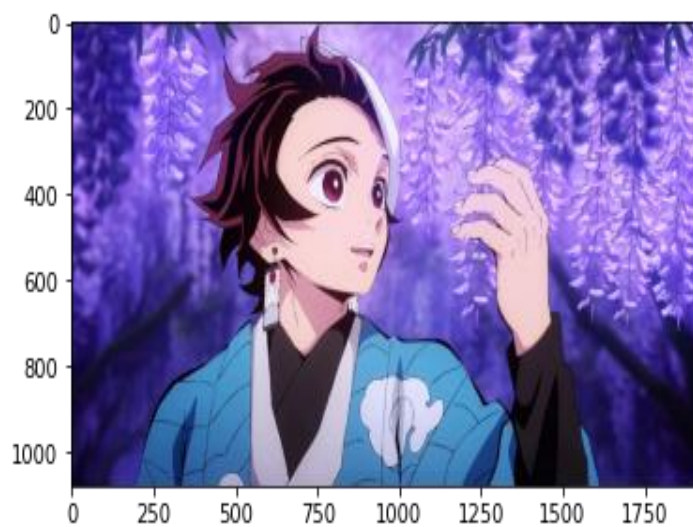
1. Write a python program to read and display an image.

[3]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
```

In [4]:

```
img1 = mpimg.imread('Tanjiri.jpg')
imgplot = plt.imshow(img1)
plt.show()
```



2. Write a python program to resize an image.

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
```

In [2]:

```
img1 = mpimg.imread('Tanjiro.jpg')
imgplot = plt.imshow(img1)
plt.show()
```



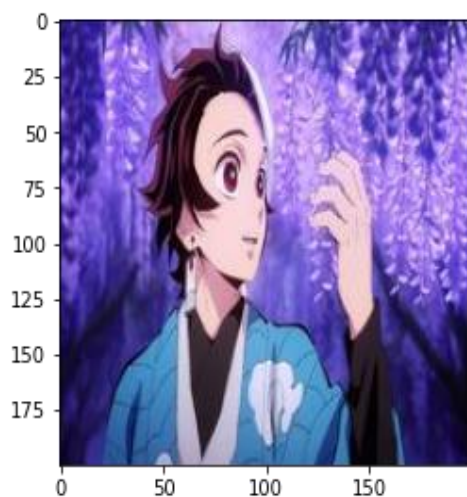
In [3]:

```
image = Image.open('Tanjiro.jpg')
print(f"Original size :{image.size}")
goku_resized= image.resize((200, 200))
goku_resizedsave('Tanjiro_400.jpeg')
```

Original size : (1920, 1080)

In

```
img = mpimg.imread('Tanjiro_400.jpeg')
imgplot = plt.imshow(img)
plt.show()
```



3. Write a python program to convert a color image into Gray-scale image

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
```

In [2]:

```
img1 = mpimg.imread('Tanjiro.jpg')
imgplot = plt.imshow(img1)
plt.show()
```



In [3]:

```
img = Image.open('Tanjiro.jpg')
imgGray = img.convert('LA') # Gray Scale
imshow = plt.imshow(imgGray)
plt.show()
```



4. Write a python program to draw the following shapes:

a) Line

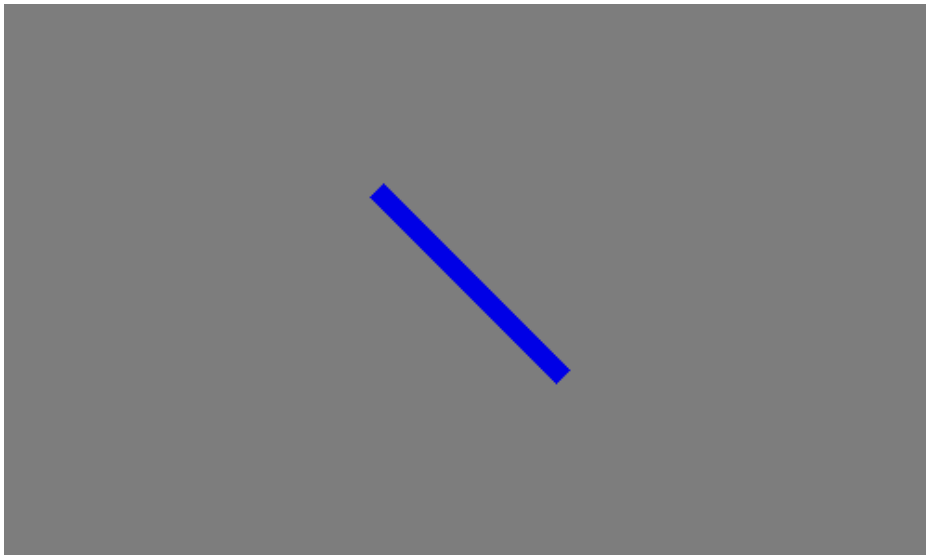
[3]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
```

In [4]:

```
img = Image.new('RGB', (500, 300), (125, 125, 125))
draw = ImageDraw.Draw(img)
draw.line((200, 100, 300, 200), fill=(0, 0, 230), width=10)
img.show()
```

In []:



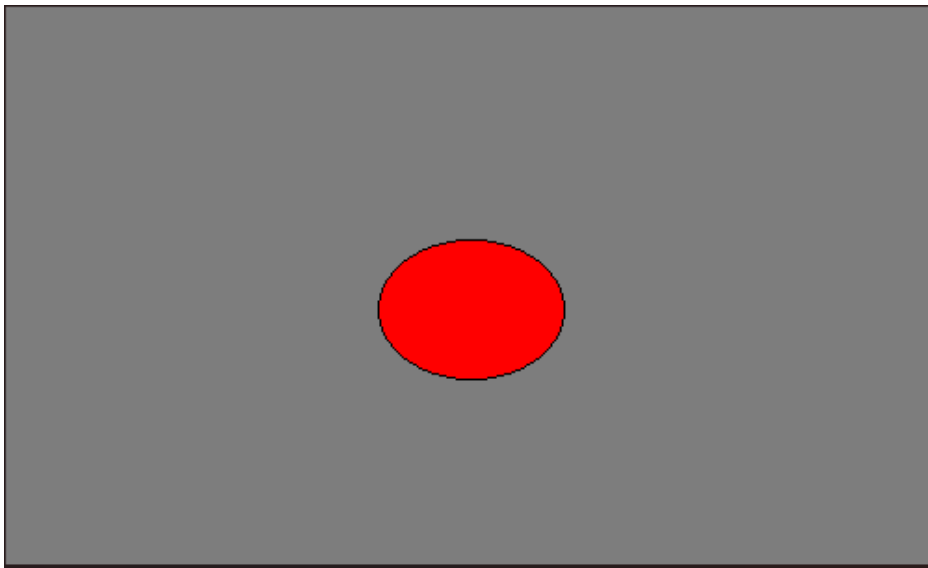
b) Ellipse

[9]:

```
from PIL import Image, ImageDraw
```

In [11]:

```
img = Image.new('RGB', (500, 300), (125, 125, 125))  
draw = ImageDrawDraw(img)  
draw.ellipse((200, 125, 300, 200), fill=(255, 0, 0), outline=(0, 0, 0))  
img.show()
```



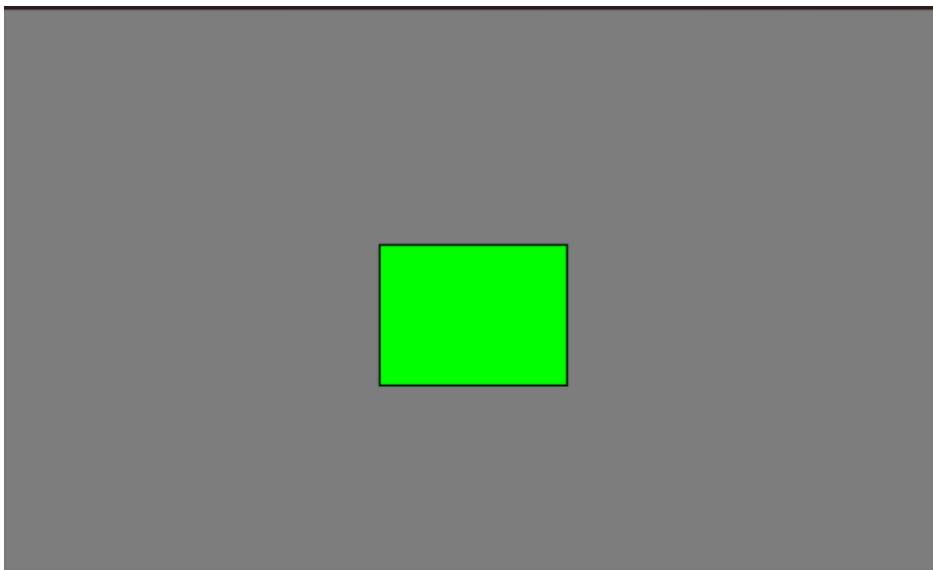
c) Rectangle

[3]:

```
from PIL import ImageDraw
```

In [4]:

```
img = Image.new('RGB', (500, 300), (125, 125, 125)) draw = ImageDraw.Draw(img)  
draw.rectangle((200, 125, 300, 200), fill=(0, 255, 0), outline=(0, 0, 0)) img.show() In [ ]:
```



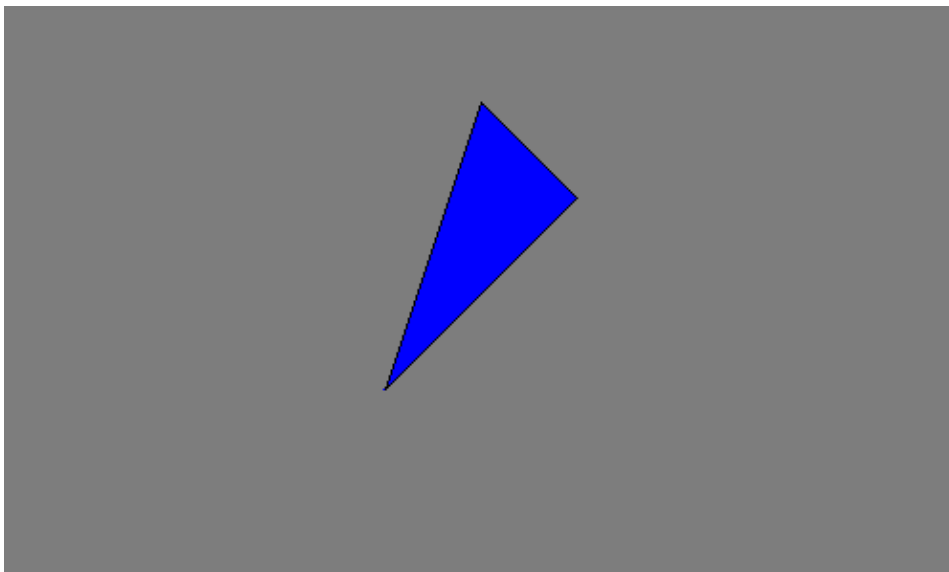
d) Polygon

[1]:

```
from PIL import Image, ImageDraw
```

In [2]:

```
img = Image.new('RGB', (500, 300), (125, 125, 125)) draw = ImageDraw.Draw(img)  
draw.polygon(((200, 200), (300, 100), (250, 50)), fill=(0, 0, 255), outline=(0, 0, 0)) img.show()
```



5. Write a python program to Flip an image into LEFT to RIGHT, TOP to BOTTOM, and ROTATE in an angle.

[]:

```
from PIL import Image
```

In []:

```
imageObject = Image.open("Tanjirō.jpg") hori_flippedImage  
= imageObject.transpose(Image.FLIP_LEFT_RIGHT) In [ ]:
```

```
imageObject.show()
```

In []:

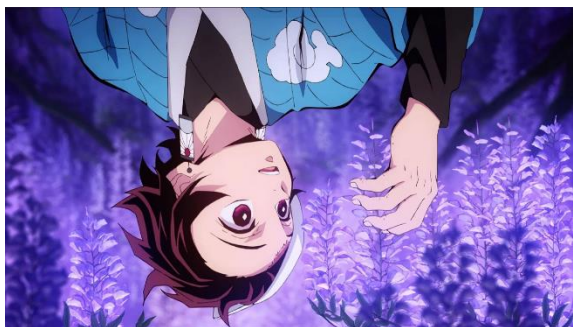
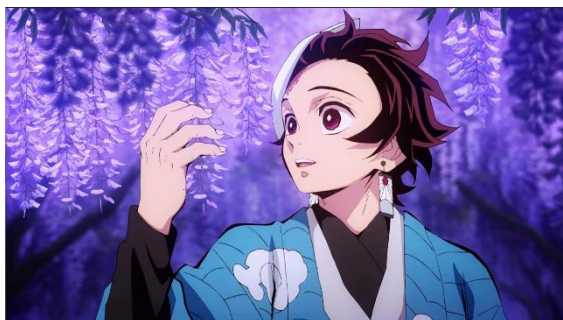
```
hori_flippedImage.show()
```

In []:

```
Vert_flippedImage= imageObject.transpose(Image.FLIP_TOP_BOTTOM)  
Vert_flippedImage.show()
```

In []:

```
degree_flippedImage= imageObject.transpose(Image.ROTATE_90)  
degree_flippedImage.show()
```

6. Write a python program to (a) Blur an image

```
from PIL import Image, ImageFilter
```

```
In [2]:
```

```
OriImage = Image.open('Tanjиро.jpg')  
blurImage = OriImage.filter(ImageFilter.BLUR)  
blurImage.show()
```

```
In [ ]:
```



b) Crop an image.

[1]:

```
from PIL import Image
```

In [2]:

```
im = Image.open("Tanjiro.jpg")  
left = 155  
top = 65  
right = 360  
bottom = 270
```

In [3]:

```
im1 = im.crop((left, top, right, bottom))  
im1.show()
```

In []:



7. Write a python program to (a) Print the array from an image

```
from PIL import Image
from numpy import array
```

```
img = Image.open('Tanjiro.jpg' )
img2arr = array ( img)
print ( img2arr )
```

```
[[[ 62  42 153]
 [ 58  38 149]
 [ 56  36 147] ...
 [108  74 246]
 [123  88 253]
 [139 104 255]]
 [[ 65  45 156]
 [ 61  41 152]
 [ 60  40 151] ...
 [100  66 238]
 [111  76 241]
 [125  90 254]]
 [[ 67  47 158]
 [ 64  44 155]
 [ 63  43 154] ...
 [ 98  62 234]
 [104  68 236]
 [117  80 246]]
 ...
 [[ 14  6 45]
 [ 14  6 45]
 [ 14  6 45] ...
 [ 15  4 44]
 [ 15  4 44]
 [ 16  6 43]]
```

[[14 6 45]

[14 6 45]

[14 6 45] ...

[14 3 43]

[15 4 44]

[15 5 42]]

[[14 6 45]

[14 6 45]

[14 6 45] ...

[15 4 44]

[15 4 44]

[16 6 43]]]

b) Put a watermark to an image

```
from PIL import Image, ImageDraw, ImageFont
```

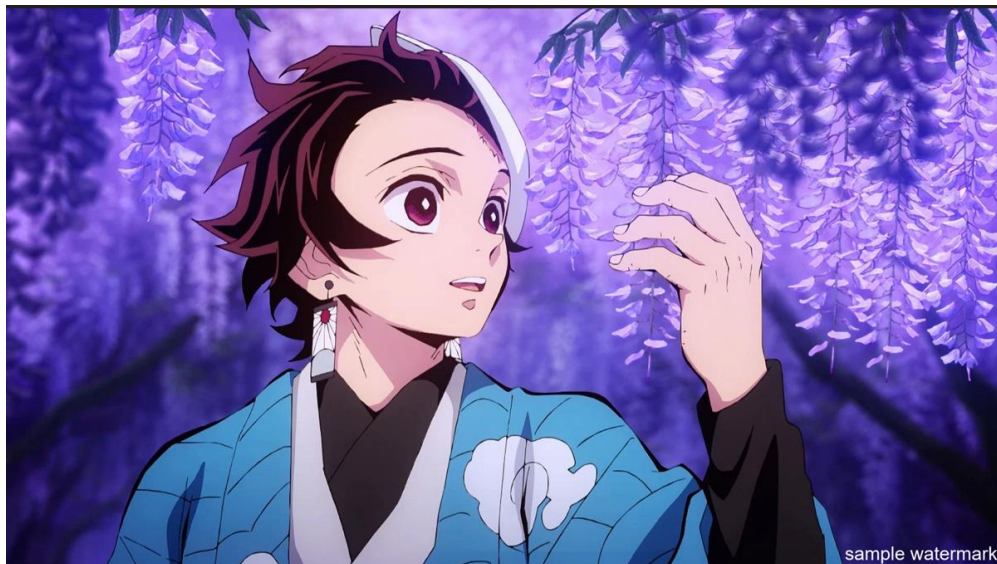
```
In [ ]:
```

```
im = Image.open('Tanjiro.jpg')  
width, height = im.size  
draw = ImageDraw.Draw(im)  
text = "sample watermark"  
font = ImageFont.truetype('arial.ttf', 36)  
textwidth, textheight = draw.textsize(text, font)
```

```
In [ ]:
```

```
margin = 10  
x = width - textwidth - margin  
y = height - textheight - margin
```

```
draw.text((x, y), text, font=font)  
im.show()
```



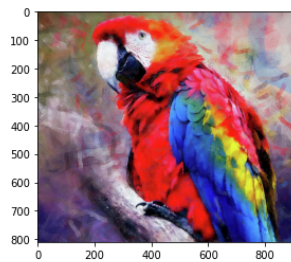
8. Write a python program to print the profiles for a color image and separate a color image in three R G & B planes.

```
In [4]: import imageio
import matplotlib.pyplot as plt
```

```
In [12]: image = imageio.imread('parrot.jpg')
```

```
In [13]: plt.imshow(image)
```

```
Out[13]: <matplotlib.image.AxesImage at 0x2b60d3a94f0>
```



```
In [6]: print('Type of the image : ', type(image))
print()
print('Shape of the image : {}'.format(image.shape))
print('Image Height : {}'.format(image.shape[0]))
print('Image Width : {}'.format(image.shape[1]))
print('Dimension of Image : {}'.format(image.ndim))

Type of the image : <class 'imageio.core.util.Array'>

Shape of the image : (810, 900, 3)
Image Height : 810
Image Width : 900
Dimension of Image : 3
```

```
In [7]: print('Image size {}'.format(image.size))
print('Maximum RGB value in this image {}'.format(image.max()))
print('Minimum RGB value in this image {}'.format(image.min()))

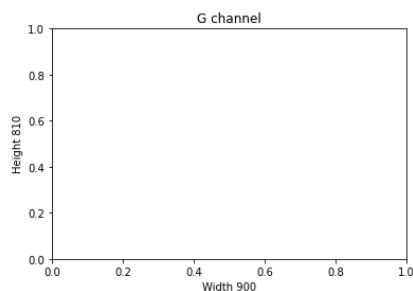
Image size 2187000
Maximum RGB value in this image 255
Minimum RGB value in this image 0
```

```
In [8]: print('Value of only R channel {}'.format(image[100, 50, 0]))
print('Value of only G channel {}'.format(image[100, 50, 1]))
print('Value of only B channel {}'.format(image[100, 50, 2]))

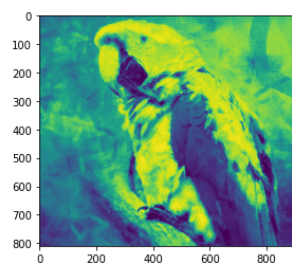
Value of only R channel 107
Value of only G channel 85
Value of only B channel 64
```

```
In [11]: plt.title('G channel')
plt.ylabel('Height {}'.format(image.shape[0]))
plt.xlabel('Width {}'.format(image.shape[1]))
```

```
Out[11]: Text(0.5, 0, 'Width 900')
```



```
In [10]: plt.imshow(image[ :, :, 0])  
plt.show()
```



```
In [ ]:
```