In [1]:
```
from PIL import Image
import PIL
import os
import glob
```

In [2]:
```
dir(Image)
```

Out[2]:
```
['ADAPTIVE',
 'AFFINE',
 'ANTIALIAS',
 'BICUBIC',
 'BILINEAR',
 'BOX',
 'CUBIC',
 'Callable',
 'DECODERS',
 'DEFAULT_STRATEGY',
 'DecompressionBombError',
 'DecompressionBombWarning',
 'ENCODERS',
 'EXTENSION',
 'EXTENT',
 'Exif',
 'FASTOCTREE',
 'FILTERED',
 'FIXED',
```

In [3]:
```
im = Image.open(r'C:\Users\hp\Downloads/tree.jpg')
```

In [4]:
```
print(f"The image size dimensions are: {im.size}")
```

```
The image size dimensions are: (259, 194)
```

In [7]:
```
file_name = 'image-1-compressed.jpg'
picture = Image.open(r'C:\Users\hp\Downloads/tree.jpg')
dim = picture.size
print(f"This is the current width and height of the image: {dim}")
```

```
This is the current width and height of the image: (259, 194)
```

In [8]:
```
picture.save("Compressed_"+file_name,optimize=True,quality=30)
```

In [3]:
```
from PIL import Image
import PIL
import os
import glob
```

In [4]:
```
dir(Image)
```

Out[4]:
```
['ADAPTIVE',
 'AFFINE',
 'ANTIALIAS',
 'BICUBIC',
 'BILINEAR',
 'BOX',
 'CUBIC',
 'Callable',
 'DECODERS',
 'DEFAULT_STRATEGY',
 'DecompressionBombError',
 'DecompressionBombWarning',
 'ENCODERS',
 'EXTENSION',
 'EXTENT',
 'Exif',
 'FASTOCTREE',
 'FILTERED',
 'FIXED',
```

In [12]:
```
im = Image.open(r'C:\Users\hp\Downloads/tree.jpg')
```

In [13]:
```
print(f"The image size dimensions are: {im.size}")
```

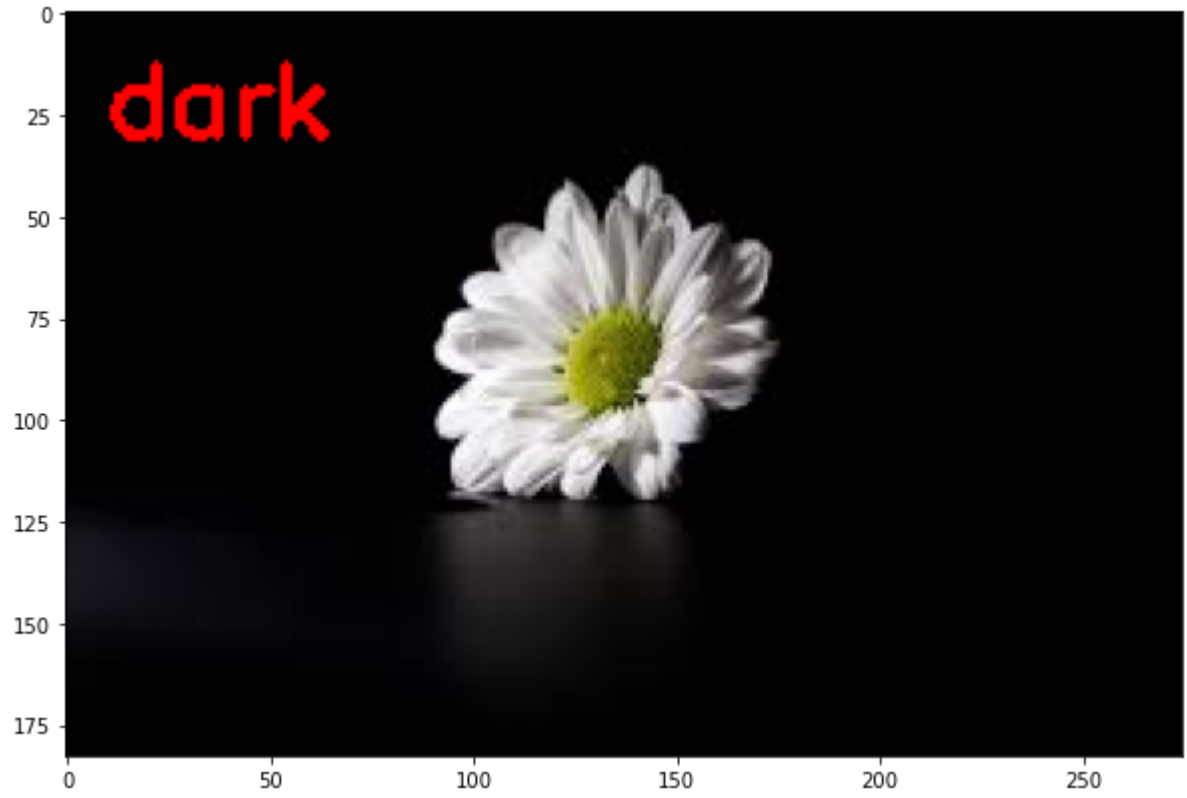
```
The image size dimensions are: (150, 100)
```

In [9]:
```
import os
import glob
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [10]:
```
def isbright(image, dim=10, thresh=0.5):
    # Resize image to 10x10
    image = cv2.resize(image, (dim, dim))
    # Convert color space to LAB format and extract L channel
    L, A, B = cv2.split(cv2.cvtColor(image, cv2.COLOR_BGR2LAB))
    # Normalize L channel by dividing all pixel values with maximum pixel value
    L = L/np.max(L)
    # Return True if mean is greater than thresh else False
    return np.mean(L) > thresh
```

In [12]:
```
# load image freom disk
image = cv2.imread(r'C:\Users\hp\Downloads/tree.jpg')

# find if image is bright or dark
text = "bright" if isbright(image) else "dark"

# write text on image
cv2.putText(image, "{}".format(text), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0

# show image
plt.figure(figsize=(10,10))
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.show()
```

In [13]:
```
# load image freom disk
image = cv2.imread(r'C:\Users\hp\Downloads/flower.jpg')

# find if image is bright or dark
text = "bright" if isbright(image) else "dark"

# write text on image
cv2.putText(image, "{}".format(text), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0

# show image
plt.figure(figsize=(10,10))
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
plt.show()
```



In [23]:
```
from PIL import Image

basewidth = 300
img = Image.open(r'C:\Users\hp\Downloads/flower.jpg')
wpercent = (basewidth / float(img.size[0]))
hsize = int((float(img.size[1]) * float(wpercent)))
img = img.resize((basewidth, hsize), Image.ANTIALIAS)
img.save('resized_image.jpg')
```