

```
# NAME :- RISHIKESH MAHESH JAMADAR
# BRANCH :- ENTC(ELECTRONICS & TELECOMMUNICATION)
# COLLAGE :- WALCHAND INSTITUTE OF TECHNOLOGY,SOLAPUR
# STD:- THIRD YEAR
# TITLE :- TAKE ANY DATA SET AND PERFORM REGRESSION /CLASSIFICATION
```

```
# major project 1 parforming classification & Regression
```

```
# 1. taking a data and creat a data frame
import pandas as pd
df = pd.read_csv('/content/Sleep_health_and_lifestyle_dataset.csv')
df
```

🔗

	Person ID	Gender	Age	Occupation	Sleep Duration	Quality of Sleep	Physical Activity Level	Stress Level	Categ
0	1	Male	27	Software Engineer	6.1	6	42	6	Overwe
1	2	Male	28	Doctor	6.2	6	60	8	No
2	3	Male	28	Doctor	6.2	6	60	8	No
3	4	Male	28	Sales Representative	5.9	4	30	8	Ot
4	5	Male	28	Sales Representative	5.9	4	30	8	Ot
...	
369	370	Female	59	Nurse	8.1	9	75	3	Overwe
370	371	Female	59	Nurse	8.0	9	75	3	Overwe
				Nurse	8.1	9	75	3	Overwe

```
# 314 rows & 13 columns
df.shape
```

(374, 13)

```
df.size
```

4862

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 374 entries, 0 to 373
Data columns (total 13 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Person ID                             374 non-null    int64
1   Gender                                374 non-null    object
2   Age                                    374 non-null    int64
3   Occupation                             374 non-null    object
4   Sleep Duration                         374 non-null    float64
5   Quality of Sleep                       374 non-null    int64
6   Physical Activity Level                 374 non-null    int64
7   Stress Level                           374 non-null    int64
8   BMI Category                           374 non-null    object
9   Blood Pressure                         374 non-null    object
10  Heart Rate                             374 non-null    int64
11  Daily Steps                           374 non-null    int64
12  Sleep Disorder                         374 non-null    object
dtypes: float64(1), int64(7), object(5)
memory usage: 38.1+ KB
```

```
# we have Sleep Duration value is in float convert in int
df['Sleep Duration'] = df['Sleep Duration'].astype('int')
```

```
df.info()
```



```
[32],
[32],
r321
```

```
y = df.iloc[:,5].values
y
```

```
array([6, 6, 6, 4, 4, 4, 6, 7, 7, 7, 6, 7, 6, 6, 6, 6, 5, 6, 5, 7, 7, 7,
       7, 7, 7, 7, 7, 7, 5, 5, 8, 6, 7, 6, 6, 7, 7, 7, 7, 7, 7, 7,
       7, 7, 7, 7, 7, 8, 8, 6, 7, 6, 6, 7, 6, 6, 7, 6, 6, 6, 6, 6, 6,
       8, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 4, 4, 7, 7, 8, 8, 8, 8,
       8, 8, 8, 8, 7, 8, 8, 8, 8, 8, 8, 8, 8, 5, 8, 5, 6, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 8, 5, 7, 9, 9, 8, 8, 8,
       8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 7, 8, 8, 8, 8, 8, 8,
       8, 6, 8, 8, 8, 8, 8, 8, 6, 6, 7, 6, 7, 6, 7, 6, 6, 6, 6, 6, 6,
       6, 6, 6, 8, 8, 6, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 6,
       7, 6, 6, 6, 7, 6, 7, 6, 7, 6, 7, 6, 7, 6, 6, 7, 6, 6, 7, 6,
       6, 7, 6, 7, 6, 7, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
       7, 6, 7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 9, 9, 6, 9, 6, 6, 6, 6, 6,
       6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 9, 9, 9, 9, 7, 6, 6, 6, 7, 7,
       7, 7, 7, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,
       9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9,
       9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9])
```

```
#4. Train_test_split/train and test variables
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,random_state = 0)
```

```
# 5. train and testing
print(x.shape)
print(x_train.shape) # 75%
print(x_test.shape)# 25%
```

```
(374, 1)
(280, 1)
(94, 1)
```

```
print(y.shape)
```

Saving...

```
(374,)
(280,)
(94,)
```

```
# 6. normalization and scaling
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.fit_transform(x_test)
```

```
#7.Apply Classifier,Regressor or Clusterer
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

```
#8.Fitting the model
model.fit(x_train,y_train)
```

```
LogisticRegression()
LogisticRegression()
```

```
y_pred = model.predict(x_test)
y_pred #PREDCITED VALUES
```

```
array([8, 9, 9, 8, 8, 8, 8, 8, 8, 9, 8, 8, 9, 9, 8, 8, 8, 9, 9, 8, 8,
       8, 8, 9, 8, 8, 8, 8, 8, 9, 8, 8, 8, 8, 6, 8, 8, 8, 8, 8, 9, 9, 9,
       8, 8, 9, 9, 8, 8, 8, 8, 8, 8, 9, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
       8, 9, 8, 8, 8, 8, 8, 9, 9, 8, 9, 9, 8, 8, 8, 8, 8, 9, 8, 8, 8, 8,
       8, 8, 9, 9, 9, 8])
```

```
y_test
```

```
array([8, 6, 7, 8, 8, 8, 6, 7, 6, 6, 9, 8, 6, 6, 9, 8, 6, 7, 9, 9, 8, 7,
       6, 6, 9, 6, 6, 7, 6, 6, 9, 8, 6, 6, 8, 7, 8, 8, 6, 7, 6, 6, 7, 9,
       7, 6, 9, 9, 4, 6, 7, 7, 7, 8, 7, 6, 8, 7, 8, 8, 7, 7, 6, 6, 7, 6,
       6, 6, 6, 8, 7, 7, 9, 9, 6, 6, 6, 8, 6, 8, 7, 4, 9, 6, 6, 7, 9,
       8, 7, 9, 9, 6, 7])
```

```
#Accuracy  
from sklearn.metrics import accuracy_score  
accuracy_score(y_pred,y_test)* 100
```

```
35.1063829787234
```



Saving...



```
# # NAME :- RISHIKESH MAHESH JAMADAR
# BRANCH :- ENTC(ELECTRONICS & TELECOMMUNICATION)
# COLLAGE :- WALCHAND INSTITUTE OF TECHNOLOGY,SOLAPUR
# STD:- THIRD YEAR
# PERFORM USING NUMPY AND OPEN CSV
```

```
# major project 2
# an image processing
```

```
import cv2
img = cv2.imread('bcd.jfif')
cv2.imshow('OUTPUT1',img)
```

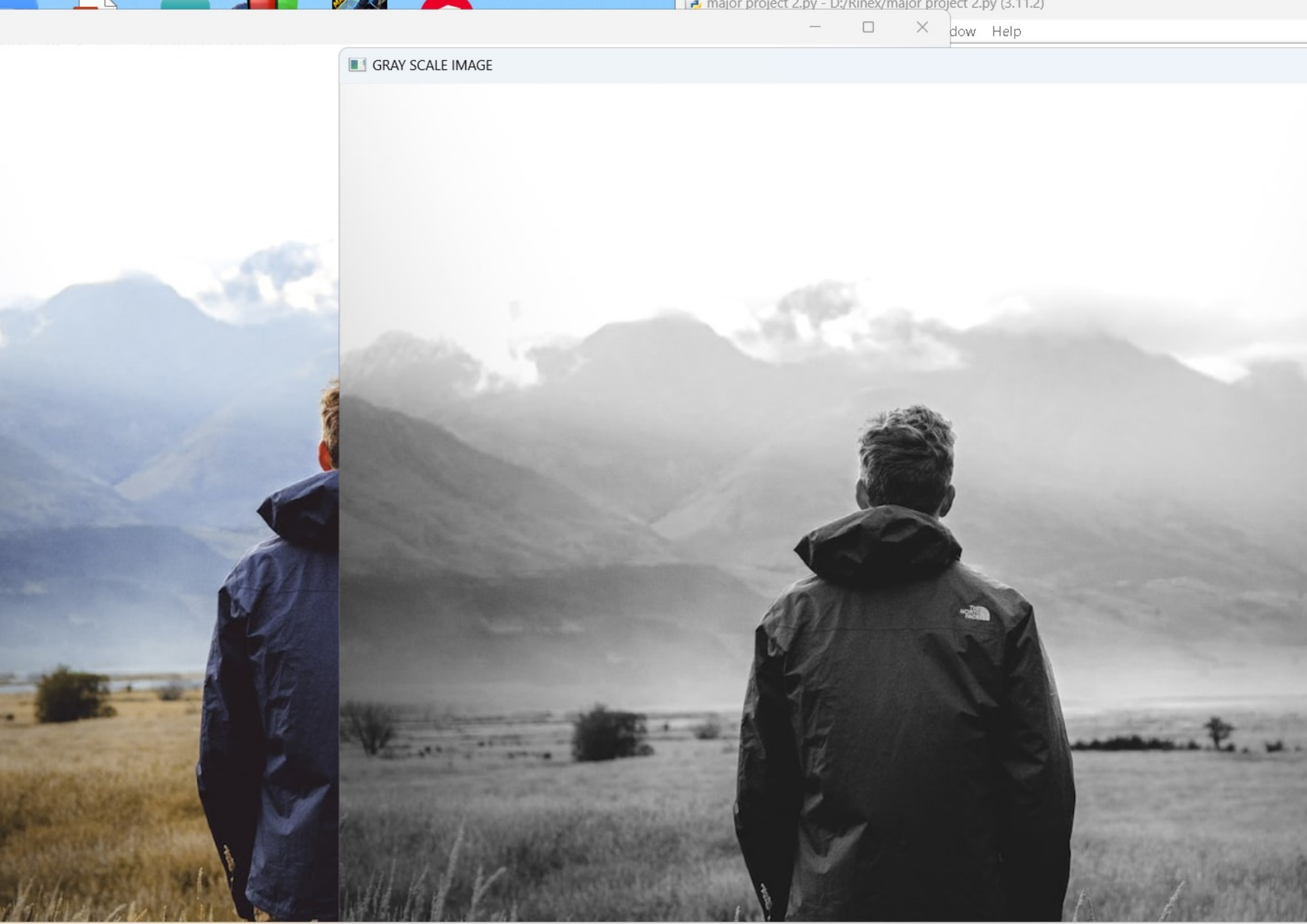
```
cv2.waitKey(0)
cv2.destroyAllWindows()
```



```
# GRAYSCALE IMAGE - BLACK & WHITE
import cv2
img = cv2.imread('bcd.jpg')

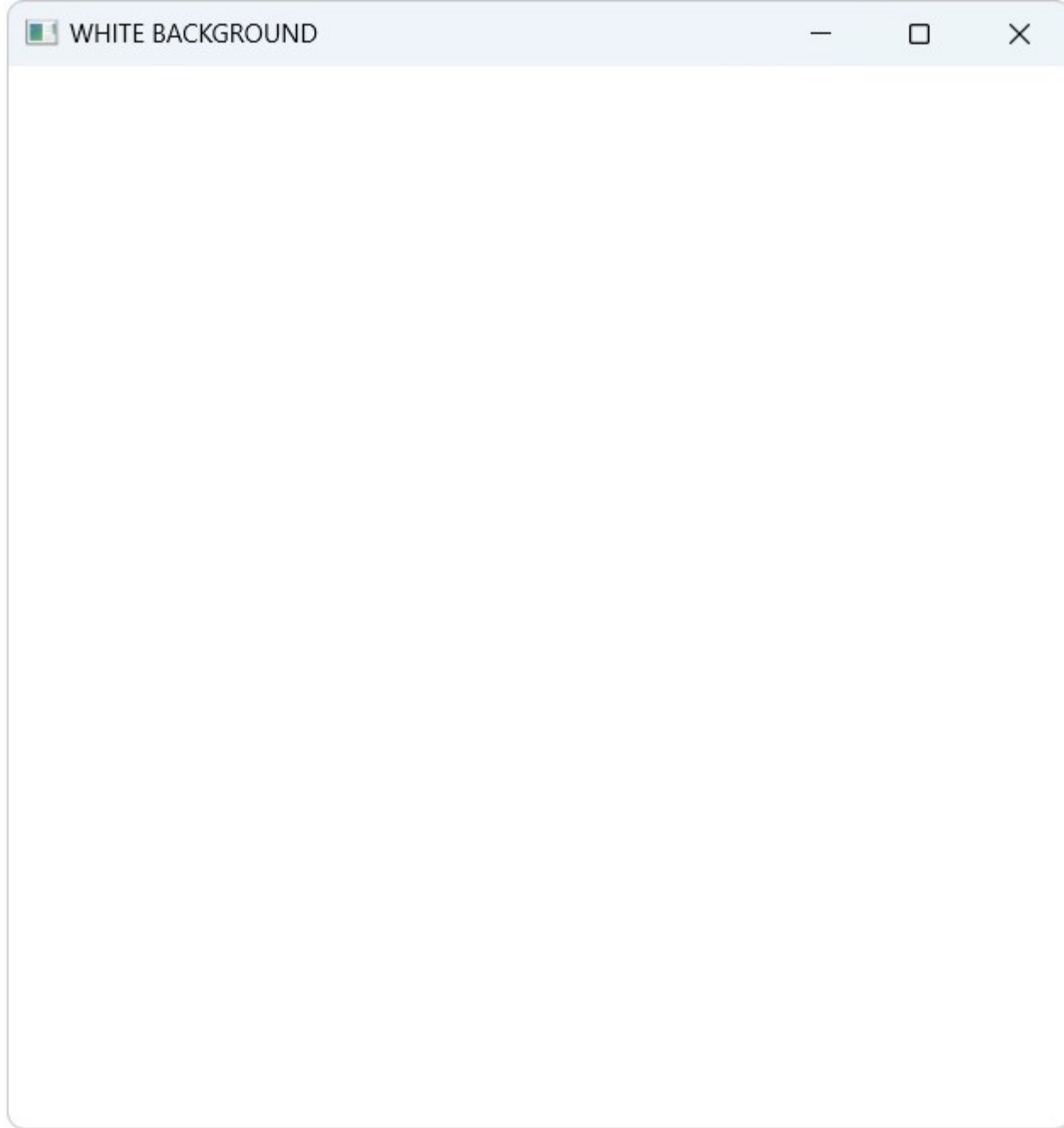
gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
cv2.imshow('NORMAL IMAGE',img)
cv2.imshow('GRAY SCALE IMAGE',gray)

cv2.waitKey(0)
cv2.destroyAllWindows()
```




```
# white backgraound
import numpy as np
import cv2

img = np.ones((500,500,3))
cv2.imshow('WHITE BACKGROUND ',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



```
#scaling images
import cv2
import numpy as np
img = cv2.imread('bcd.jfif')
cv2.imshow('Original Image',img)
cv2.waitKey(500)

#Now let us reduce the scale of our image from 100 to 75%
img1 = cv2.resize(img, None, fx = 0.75, fy = 0.75)
cv2.imshow('Scale down',img1)

#Now let us increase the scale of our image from 100 to 150%
img2 = cv2.resize(img, None, fx = 1.5, fy = 1.5)
cv2.imshow('Scale up',img2)

#Now let us give custom dimensions(length and width given in pixels)
img3 = cv2.resize(img, (200,60))
cv2.imshow('Custom Dimensions',img3)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

