

AI Practicals

1. Write a program to check whether the given character is an uppercase letter or lowercase letter or a digit or a special character.

```
#Input the character to check
ch=input("Enter Any Character:")
'''Checking whether it is upperletter or
lowerletter or digit or a special character'''
if ch.isupper():
    print(ch, " is an upper case letter")
elif ch.islower():
    print(ch, " is a lower case letter")
elif ch.isdigit():
    print(ch, " is a digit")
elif ch.isspace():
    print(ch, " is a space")
else:
    print(ch," is a special character")
```

2. Write a program to find the maximum number out of the given three numbers.

```
#Take input or three number to compare
n1=int(input("Enter the Number1:"))
n2=int(input("Enter the Number2:"))
n3=int(input("Enter the Number3:"))
if n1>n2 and n1>n3:
    print(n1, " - Number 1 is greater")
elif n2>n1 and n2>n3:
    print(n2, " - Number 2 is greater")
elif n3>n1 and n3>n2:
    print(n3, " - Number 3 is greater")
else:
    print("All are same")
```

3. An electric power distribution company charges its domestic consumers as follows

Consumption Units	Rate of Charge
0-100	Rs. 1 per unit
101-300	Rs. 100 plus Rs. 1.25 per unit in excess of 100
301-500	Rs. 350 plus Rs. 1.50 per unit in excess of 300
500 and above	Rs. 650 plus Rs. 1.75 per unit in excess of 500

Write a program that read the customer number & power consumed and prints the amount to be paid by the customer. Note that output should be well formatted.

```
#Input Data
cno=int(input("Enter Cusumer Number:"))
pc=int(input("Enter power consumed:"))
#Computing bill amount based on power consumed
if pc>0 and pc<=100:
    bill_amt=pc*1
elif pc>100 and pc<=300:
    bill_amt=100+(pc-100)*1.25
elif pc>300 and pc<500:
    bill_amt=350+(pc-300)*1.50
elif pc>500:
    bill_amt=650+(pc-500)*1.75
else:
    print("Invalid Power Consumed Units")
#Printing the bill in proper format
print("~"*60)
print("\t\tABC Power Company Ltd.")
print("~"*60)
print("Consumer Number:",cno)
print("Consumed Units:",pc)
print("-----")
print("Bill Amount:",bill_amt)
```

4. Write a program to create a list of students' marks with user-defined values and find the maximum.

```

#Take input for n lines
n=int(input("Enter no. of subjects:"))
#Creating empty list
l=[]
#Accepting marks and appending marks into the list
for i in range(n):
    m=int(input("Enter marks:"))
    l.append(m)
print("Maximum marks scored:",max(l))

```

5. Write a program to create a list of numbers and swap the content with the next value divisible by 5.

For example: list = [4,25,31,7,35,44,55]

Output: [25,4,31,35,7,55,44]

```

#Take input for no of subjects
n=int(input("Enter no. of subjects:"))
#Creating empty list
l=[]
#Accepting marks and appending marks into the list
for i in range(n):
    m=int(input("Enter marks:"))
    l.append(m)
#Swaping elements
for i in range(len(l)) :
    if l[i] % 5 == 0 :
        l [ i ], l [i-1] = l [ i - 1 ] , l [i]
print("List after swap:",l)

```

6. Write a program to generate the following pattern.

```

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6

```



```

#Take input for n Lines
n=int(input("Enter n:"))
#Generating Pattern
k=1
for i in range(1,n+1):
    for j in range(1,i+1):
        print(k,end=" ")
    k=k+1
    print()

```

Unit 4 Data Science Programs

7. Write a program to create a 1D array using NumPy.

```

#import numpy package
import numpy as np
#Creating array using arange() function
arr=np.arange(1,8,2)
#printing array
print(arr)

```

8. Write a program to convert a python list to a NumPy array.

```

#Import NumPy Package
import numpy as np
#Creating empty List
l = []
#Take input for n no. of elements
n=int(input("Enter the no. of elements:"))
#Append the values into the List
for i in range(n):
    val=int(input("Enter value "+str(i+1)+":"))
    l.append(val)
#Converting List into numpy array
arr=np.array(l)
print("Array:",arr)
print("list",l)

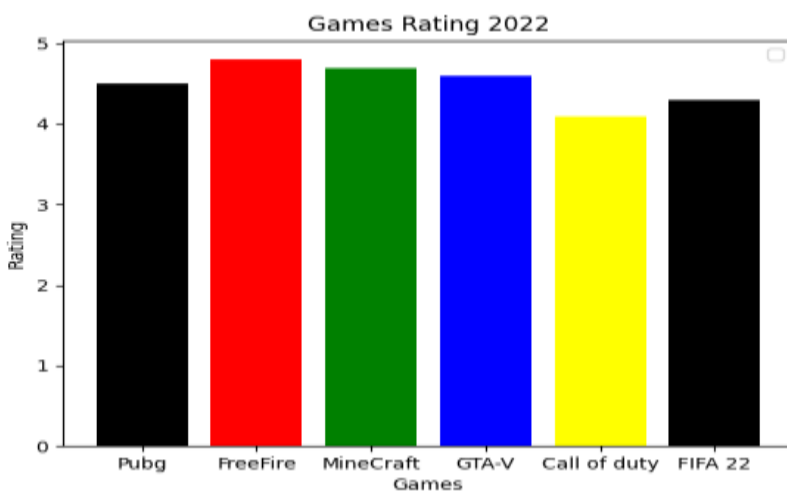
```

9. Write a program to develop a matrix of 3x3 with values from 21 to 38.

```
#import numpy package
import numpy as np
#Creating array using arange() function
arr=np.arange(21,38,2)
#reshaping array for 2D
arr=arr.reshape(3,3)
#printing array
print(arr)
```

10. Write a program to represent the data on the ratings of mobile games on barchart. The sample data is given as: Pubg, FreeFire, MineCraft, GTA-V, Call of duty, FIFA 22. The rating for each game is as: 4.5,4.8,4.7,4.6,4.1,4.3.

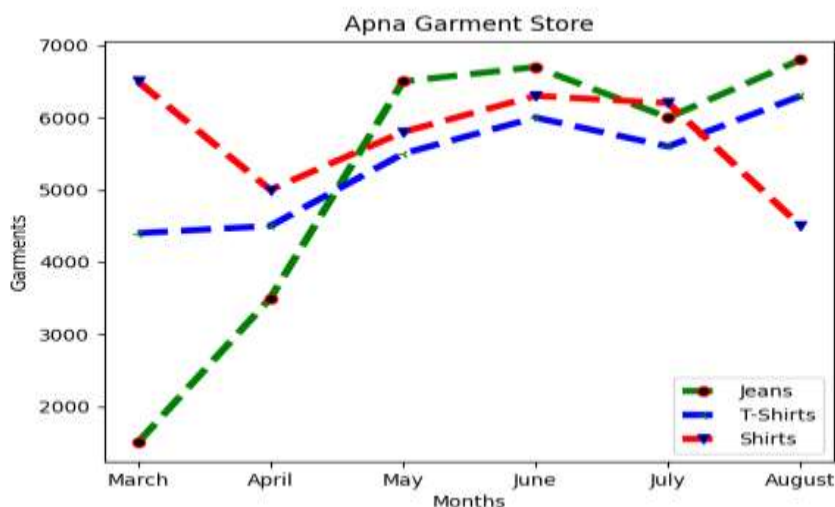
```
#Import package for Matplot Library
import matplotlib.pyplot as plt
#Creating Lists for data
games=['Pugb', 'FreeFire', 'MineCraft', 'GTA-V','Call of duty', 'FIFA 22']
rating=[4.5,4.8,4.7,4.6,4.1,4.3]
#Creating bar graph with different bar colours
plt.bar(games,rating,color=['black', 'red', 'green', 'blue', 'yellow'])
#Customizing the bar graph
plt.title("Games Rating 2022")
plt.xlabel('Games')
plt.ylabel('Rating')
plt.legend();
plt.show()
```



11. Consider the following data of a clothes store and plot the data on the line chart:

Month	Jeans	T-Shirts	Shirts
March	1500	4400	6500
April	3500	4500	5000
May	6500	5500	5800
June	6700	6000	6300
July	6000	5600	6200
August	6800	6300	4500

```
#import package for line chart
import matplotlib.pyplot as pp
#creating list data
mon = ['March', 'April', 'May', 'June', 'July', 'August']
jeans = [1500, 3500, 6500, 6700, 6000, 6800]
ts = [4400, 4500, 5500, 6000, 5600, 6300]
sh = [6500, 5000, 5800, 6300, 6200, 4500]
#Creating Line chart
pp.plot(mon, jeans, label='Mask', color='g', linestyle='dashed', linewidth=4, \
marker='o', markerfacecolor='k', markeredgecolor='r')
pp.plot(mon, ts, label='Mask', color='b', linestyle='dashed', linewidth=4, \
marker='3', markerfacecolor='k', markeredgecolor='g')
pp.plot(mon, sh, label='Mask', color='r', linestyle='dashed', linewidth=4, \
marker='v', markerfacecolor='k', markeredgecolor='b')
#Cusotmizing plot
pp.title("Apna Garment Store")
pp.xlabel("Months")
pp.ylabel("Garments")
pp.legend()
pp.show()
```



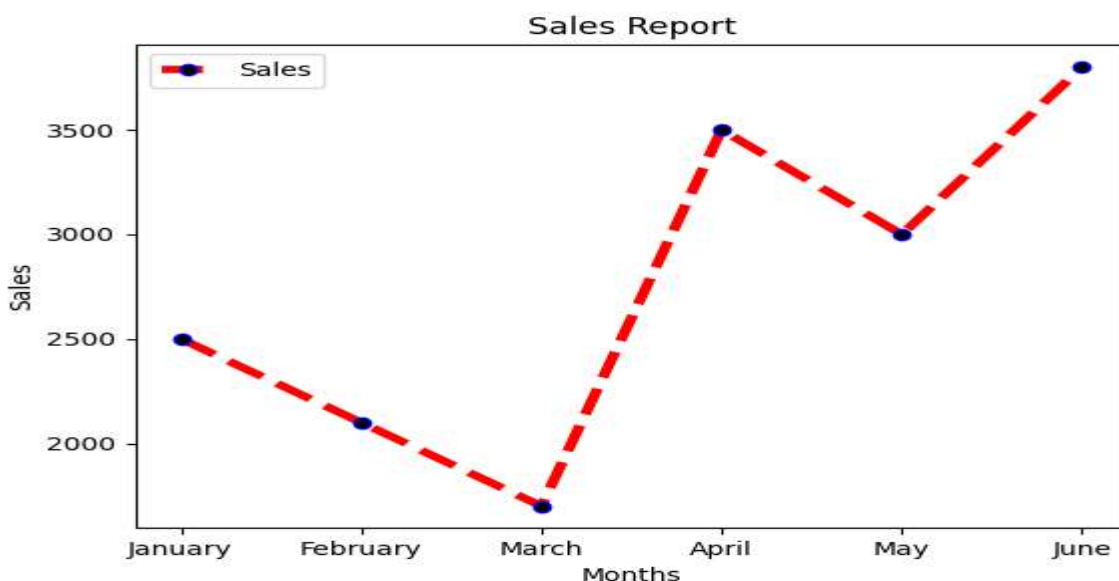
12. Observe the given data for monthly sales of one of the salesmen for 6 months. Plot them on the line chart.

Month	January	February	March	April	May	June
Sales	2500	2100	1700	3500	3000	3800

Apply the following customizations to the chart:

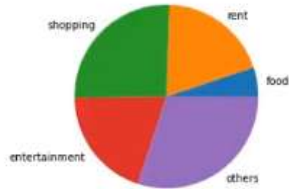
- Give the title for the chart - "Sales Stats"
- Use the "Month" label for X-Axis and "Sales" for Y-Axis.
- Display legends.
- Use dashed lines with the width 5 point.
- Use red color for the line.
- Use dot marker with blue edge color and black fill color.

```
#Import pyplot
import matplotlib.pyplot as pp
#Prepraing data
mon = ['January', 'February', 'March', 'April', 'May', 'June']
sales = [2500, 2100, 1700, 3500, 3000, 3800]
#Crating line chart
pp.plot(mon, sales, label='Sales', color='r', linestyle='dashed', linewidth=4, \
marker='o', markerfacecolor='k', markeredgecolor='b')
#Customizing the chart
pp.title("Sales Report")
pp.xlabel("Months")
pp.ylabel("Sales")
pp.legend()
pp.show()
```



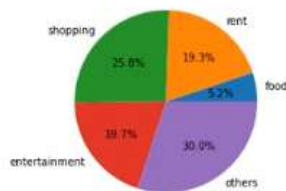
Data Science Practical (Pie Chart)

```
In [5]: import matplotlib.pyplot as plt
values=[12,45,60,46,70]
Necessity=['food','rent','shopping','entertainment','others']
plt.pie(values,labels=Necessity,radius=1.0)
plt.show()
```



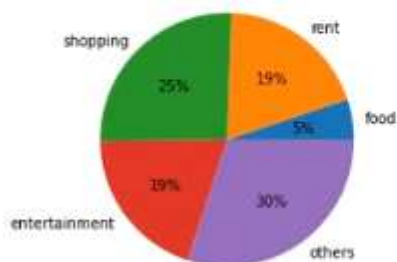
F use for complete floating no.

```
In [6]: import matplotlib.pyplot as plt
values=[12,45,60,46,70]
Necessity=['food','rent','shopping','entertainment','others']
plt.pie(values,labels=Necessity,radius=1.0,autopct='%0.1f%%')
plt.show()
```

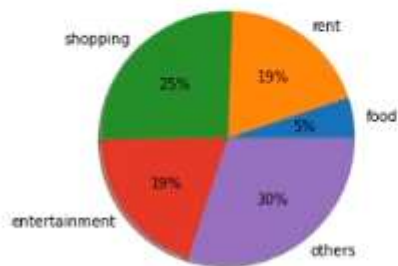


D use for complete no.

```
In [7]: import matplotlib.pyplot as plt
values=[12,45,60,46,70]
Necessity=['food','rent','shopping','entertainment','others']
plt.pie(values,labels=Necessity,radius=1.0,autopct='%d%%')
plt.show()
```



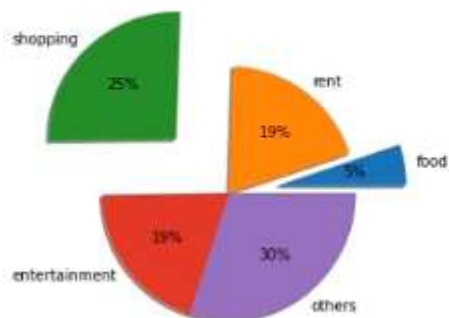

```
In [12]: import matplotlib.pyplot as plt
values=[12,45,60,46,70]
Necessity=['food','rent','shopping','entertainment','others']
plt.pie(values,labels=Necessity,radius=1.0,autopct='%d%%',shadow=True)
plt.show()
```



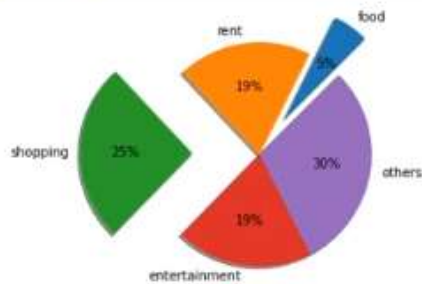
```
In [15]: import matplotlib.pyplot as plt
values=[12,45,60,46,70]
Necessity=['food','rent','shopping','entertainment','others']
plt.pie(values,labels=Necessity,radius=1.0,autopct='%d%%',shadow=True,explode=[0.4,0,0,0,0])
plt.show()
```



```
In [16]: import matplotlib.pyplot as plt
values=[12,45,60,46,70]
Necessity=['food','rent','shopping','entertainment','others']
plt.pie(values,labels=Necessity,radius=1.0,autopct='%d%%',shadow=True,explode=[0.4,0,0.6,0,0])
plt.show()
```

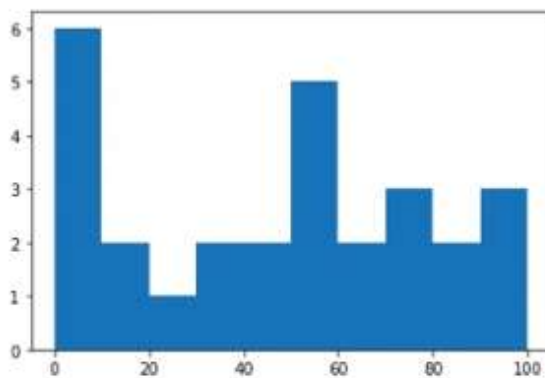


```
In [17]: import matplotlib.pyplot as plt
values=[12,45,60,46,70]
Necessity=['food','rent','shopping','entertainment','others']
plt.pie(values,labels=Necessity,radius=1.0,autopct='%0d%',shadow=True,explode=[0.4,0,0.6,0,0],startangle=45)
plt.show()
```



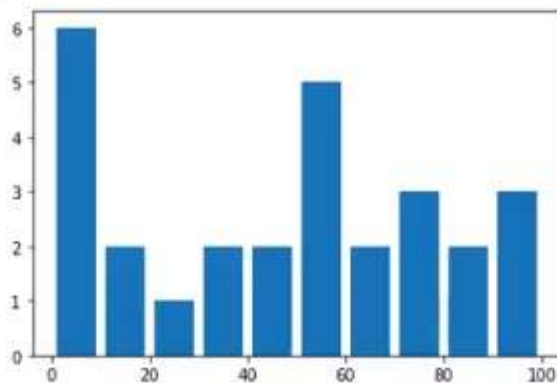
Data Science Practical's (Histogram Plot)

```
In [3]: import matplotlib.pyplot as plt
ages=[12,34,55,6,4,56,77,89,90,78,67,56,44,3,2,12,2,4,67,89,90,99,78,56,45,34,23,56]
age_group=[0,10,20,30,40,50,60,70,80,90,100]
plt.hist(ages,age_group)
plt.show()
```



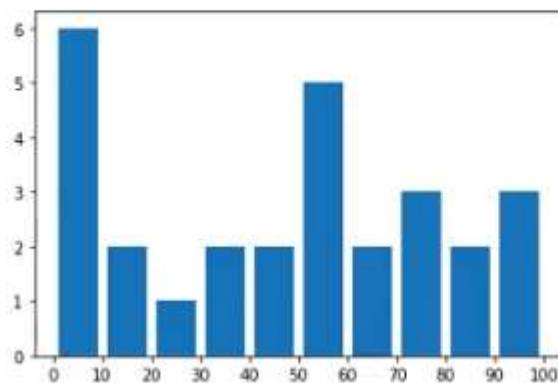
Attribute: rwidth=0.8

```
In [4]: import matplotlib.pyplot as plt
ages=[12,34,55,6,4,56,77,89,90,78,67,56,44,3,2,12,2,4,67,89,90,99,78,56,45,34,23,56]
age_group=[0,10,20,30,40,50,60,70,80,90,100]
plt.hist(ages,age_group,rwidth=0.8)
plt.show()
```



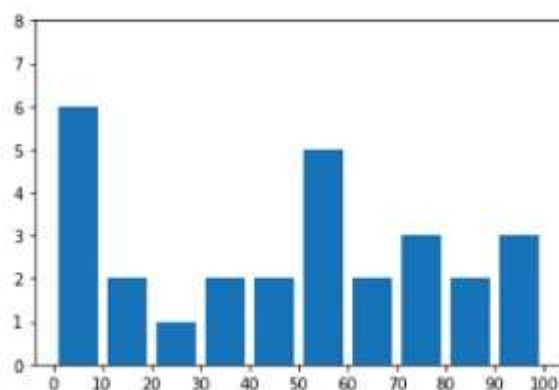
Function: xticks(age_group)

```
In [5]: import matplotlib.pyplot as plt
ages=[12,34,55,6,4,56,77,89,90,78,67,56,44,3,2,12,2,4,67,89,90,99,78,56,45,34,23,56]
age_group=[0,10,20,30,40,50,60,70,80,90,100]
plt.hist(ages,age_group,rwidth=0.8)
plt.xticks(age_group)
plt.show()
```



Function: `yticks(range(0,9,1))`

```
In [8]: import matplotlib.pyplot as plt
ages=[12,34,55,6,4,56,77,89,90,78,67,56,44,3,2,12,2,4,67,89,90,99,78,56,45,34,23,56]
age_group=[0,10,20,30,40,50,60,70,80,90,100]
plt.hist(ages,age_group,rwidth=0.8)
plt.xticks(age_group)
plt.yticks(range(0,9,1))
plt.show()
```



Data science, Scatter plot (Practical)

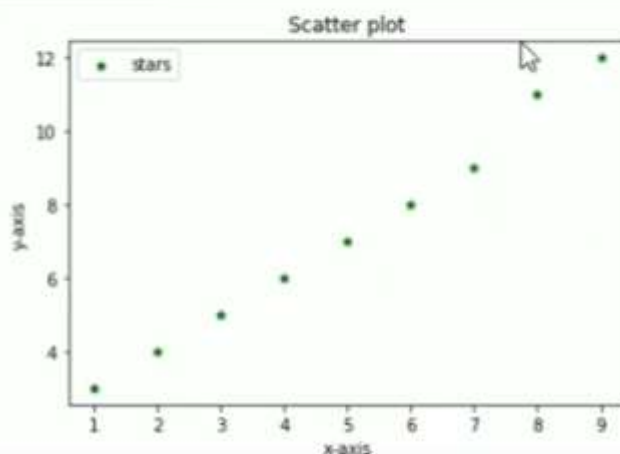

```
In [4]: import matplotlib.pyplot as plt

x=[1,2,3,4,5,6,7,8,9]
y=[3,4,5,6,7,8,9,11,12]

plt.scatter(x,y,label='stars',color='green',marker='*',s=30)

plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('Scatter plot')

plt.legend()
plt.show()
```

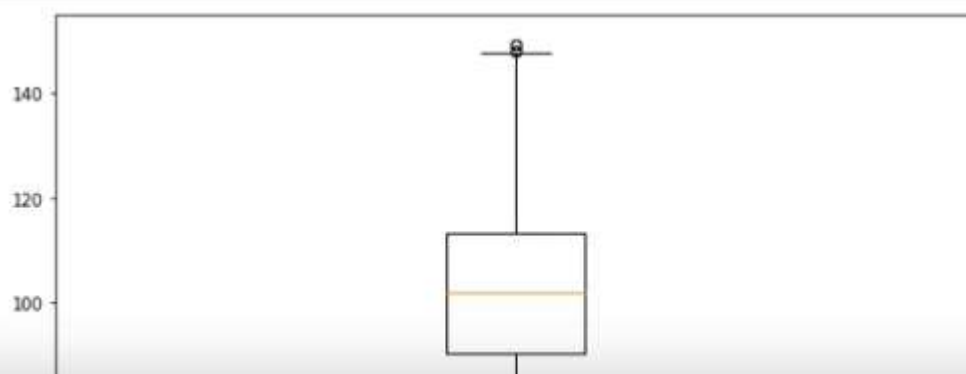


Boxplot Chart

```
In [2]: import matplotlib.pyplot as plt
import numpy as np

#creating dataset
np.random.seed(10)
data = np.random.normal(100,20,200)
fig = plt.figure(figsize=(10,7))

#creating plot
plt.boxplot(data)
plt.show()
```



For multiple boxplot

```
In [3]: import matplotlib.pyplot as plt
import numpy as np

#creating dataset
np.random.seed(10)
data_1 = np.random.normal(100,20,200)
data_2 = np.random.normal(90, 20, 200)
data_3 = np.random.normal(80, 30, 200)
data_4 = np.random.normal(70, 40, 200)
data = [data_1,data_2,data_3,data_4]
fig = plt.figure(figsize=(10,7))

#creating axes instance
ax = fig.add_axes([0,0,1,1])

#creating plot
bp=ax.boxplot(data)
plt.show()
```



Computer vision practical

Q. how to insert image in jupyter notebook

Method 1

Step 1. Goto code section and select the **Markdown**

Step2. Click on edit and click on **insert image** option and ask to you select image and choose an image. And choose file and ok.

```
![School_logo.jpg](attachment:School_logo.jpg) I
```

Step3. Click on run button.



Method 2

```
In [1]: from PIL import Image
im=Image.open(r"C:\Users\91812\Desktop\School_logo.jpg")
im.show()
```



Resolution of image

Q. To find out the height and width of an image.

```
In [2]: from PIL import Image
        im=Image.open(r"C:\Users\91812\Desktop\School_logo.jpg")
        wid, hgt=im.size
        print(str(wid)+"*"+str(hgt))
```

200*200

Method 2

```
In [10]: import cv2
         im=cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg")
         wid = im.shape[0]
         hgt = im.shape[1]
         print(str(wid)+"*"+str(hgt))
```

I

200*200

Image segmentation


```
In [ ]: import cv2
import numpy as np
import matplotlib.pyplot as plt

img=cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg",0)
ret, thresh1=cv2.threshold(img,120,255,cv2.THRESH_BINARY)
ret, thresh2=cv2.threshold(img,120,255,cv2.THRESH_BINARY_INV)
ret, thresh3=cv2.threshold(img,120,255,cv2.THRESH_TRUNC)
ret, thresh4=cv2.threshold(img,120,255,cv2.THRESH_TOZERO)
ret, thresh5=cv2.threshold(img,120,255,cv2.THRESH_TOZERO_INV)

titles=['original','binary','binary_inv','trunc','tozero','tozero_inv']
images=[img,thresh1,thresh2,thresh3,thresh4,thresh5]

for i in range(6):
    plt.subplot(2,3,i+1)
    plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([])
    plt.yticks([])

plt.show()
```



```
In [4]: import cv2
import numpy as np
import matplotlib.pyplot as plt

img=cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg",0)
ret, thresh1=cv2.threshold(img,120,255,cv2.THRESH_BINARY)
ret, thresh2=cv2.threshold(img,120,255,cv2.THRESH_BINARY_INV)
ret, thresh3=cv2.threshold(img,120,255,cv2.THRESH_TRUNC)
ret, thresh4=cv2.threshold(img,120,255,cv2.THRESH_TOZERO)
ret, thresh5=cv2.threshold(img,120,255,cv2.THRESH_TOZERO_INV)

titles=['original','binary','binary_inv','trunc','tozero','tozero_inv']
images=[img,thresh1,thresh2,thresh3,thresh4,thresh5]

for i in range(6):
    plt.subplot(2,3,i+1)
    plt.imshow(images[i])
    plt.title(titles[i])
    plt.xticks([])
    plt.yticks([])

plt.show()
```



pixel value

```
In [9]: from PIL import Image
# creating an image object
im=Image.open(r"C:\Users\91812\Desktop\School_logo.jpg")
px=im.load()
print(px[4,4])

(255, 255, 255)
```

If $px[4,4]=(0,0,0)$

```
In [10]: from PIL import Image
# creating an image object
im=Image.open(r"C:\Users\91812\Desktop\School_logo.jpg")
px=im.load()
print(px[4,4])
px[4,4]=(0,0,0)
print(px[4,4])

(255, 255, 255)
(0, 0, 0)
```

```
In [11]: from PIL import Image
# creating an image object
im=Image.open(r"C:\Users\91812\Desktop\School_logo.jpg")
px=im.load()
print(px[4,4])
px[4,4]=(0,0,0)
print(px[4,4])
coordinate= x,y=150,59
# using getpixel method
print(im.getpixel(coordinate));

(255, 255, 255)
(0, 0, 0)
(176, 35, 28)
```

RGB images (show original image)

```
In [*]: import cv2
image= cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg")
cv2.imshow("original",image)
cv2.waitKey()
```

Gray_image: convert color

```
In [*]: import cv2
image= cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg")
cv2.imshow("original",image)
cv2.waitKey()
gray_image=cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
cv2.imshow("Grayscale",gray_image)
cv2.waitKey()
cv2.destroyAllWindows()
```

Method 2: if set 1 then original image and set 0 then grayscale image

```
In [9]: import cv2
image=cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg",1)
cv2.imshow("Grayscale",image)
cv2.waitKey()
cv2.destroyAllWindows()
```

```
In [19]: import cv2
import matplotlib.pyplot as plt
image=cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg",1)
plt.imshow(image)
```

Out[19]: <matplotlib.image.AxesImage at 0x17884b803a0>



```
In [20]: import cv2
import matplotlib.pyplot as plt
image=cv2.imread(r"C:\Users\91812\Desktop\School_logo.jpg",0)
plt.imshow(image)
```

Out[20]: <matplotlib.image.AxesImage at 0x17884bdf30>

