Visualization

- Matplotlib
- Seaborn

Matplotlib

- Matplotlib is one of the visualization library for 2D plotting
- It was introduced John Hunter in the year of 2002
- line plot,scatter plot,bar chart,histogram

In [1]:
```
# importing matplotlib

import matplotlib.pyplot as plt
```

In [4]:
```
# image reading
img=plt.imread('logo.jpg')
img
```

...

pyplot

- It is one of the popular library in matplotlib library
- It contains collection of functions, which is easy and simple interface for constructing plots

In [5]:
```
plt.imshow(img)
```

...

In [6]:
```
# image slicing
plt.imshow(img[45:100,20:60])
```

...

Types of plots

- Line
- Bar chart
- Scatter plot
- Histogram
- Boxplot
- Piechart

Line

- Line is a graph it shows frequency of data in a number line

In [18]:
```
x=[10,12,14,16]
y=[4,5,6,7]
plt.plot(x,y,label="Line")
plt.xlabel("X-AXIS")
plt.ylabel("Y-AXIS")
plt.title("Line graph")
plt.legend(loc="lower right")
```

...

In [19]:
```
import numpy as np
x=np.linspace(0,5,20)
x
```

...

In [22]:
```
plt.plot(x,x**2,color="red",label="Line1")
plt.plot(x,x**3,color="green",label="Line2")
plt.plot(x,x**4,color="blue",label="Line3")
plt.legend()
```

...

In [23]:
```
help(plt.plot)
```

...

In [25]:
```
x1=np.arange(100)
x1
```

...

In [33]:
```
plt.plot(x1,x1+10,color='c',linestyle="--")
plt.plot(x1,x1+20,color='m',linestyle="-.")
plt.plot(x1,x1+15,color='y',linestyle=":")
plt.plot(x1,x1+25,color='g',linestyle="-")
```

...

Barchart

- along with pyplot,using bar() we can draw bar chart
- plt.bar()

In [44]:
```
year=['mar 2020','april 2021','may 2023']
goldprices=[4500,5400,5500]
silver=[1000,1200,1500]
plt.bar(year,goldprices,width=0.33,color="green")
plt.bar(year,silver,width=0.33,color="red")
```

...

In [47]:
```
x=[1,2,3,4,5,6]
y=[10,11,12,13,14,15]
plt.bar(x,y,color=['c','m','g','r','k','y'],edgecolor="blue")
```

...

In [48]:
```
# horizontal barchart
#plt.barh()
plt.barh(x,y)
```

...

scatter plot

- Scatter plot is a diagram where each data point is represented by a dot
- plt.scatter()

In [49]:
```
plt.scatter([12,14,15],[16,17,18])
```

...

In [51]:
```
x=np.random.randint(100,1000,100)
y=np.random.randint(200,1000,100)
plt.scatter(x,y)
```

...

boxplot

- It gives the summary of numerical data through quartiles
- min
- InterQuartileRange(IQR)
  - lower quartile(25%)
  - Median (50%)
  - Upper Quartile(75%)
- max()
- plt.boxplot()

In [52]:
```
numbers=np.random.randint(0,1000,1000)
plt.boxplot(numbers)
```

...

In [53]:
```
# Notched boxplot
plt.boxplot(numbers,notch=True)
```

...

Histogram

- It is a diagram ,which shows frequency distribution of numerical data
- plt.hist()

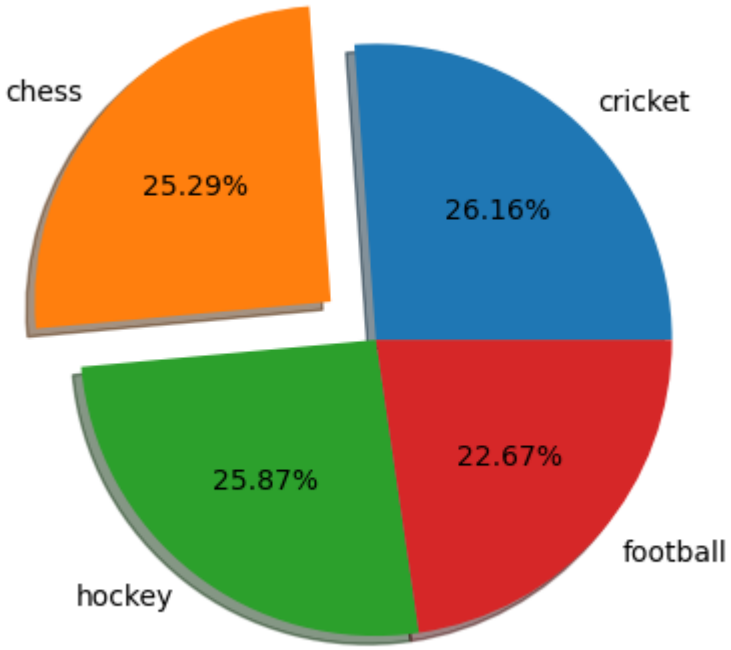
In [54]:
```
x=[10,20,30,24,35,47,56,12]
plt.hist(x)
```

...

piechart

- plt.pie()
- you can use pie() to draw a pie chart

In [61]:
```
games=["cricket","chess","hockey","football"]
scores=[90,87,89,78]
plt.pie(scores,labels=games,autopct="%1.2f%%",
        explode=[0,0.2,0,0],shadow=True)
plt.show()
plt.savefig("piechart.png")
```




```
<Figure size 640x480 with 0 Axes>
```

Seaborn

- Based on Matplotlib
- Seaborn is one of visualization library in python , for statistical plotting
- It is designed to work with data frame objects in pandas
- It contains default attractive styles
- It contains high level interface for drawing attractive and informative statistical plots

In [62]:
```
import seaborn as sns
```

In [63]:
```
sns.__version__
```

Out[63]:
```
'0.11.2'
```

In [64]:
```
dir(sns)
```

...

Types of plots

- Color_palette
- Categorical plot
- jointplot
- pairplot
- Heatmaps

Color_palette

- It is an interface to generate few colors in seaborn
- sns.color_palette()

In [65]:
```
1 sns.color_palette()
```

...

In [66]:
```
1 help(sns.color_palette())
```

...

In [68]:
```
1 sns.palplot(sns.color_palette())
```

...

In [70]:
```
1 sns.palplot(sns.color_palette("deep"))
```

...

In [71]:
```
1 sns.palplot(sns.color_palette("muted"))
```

...

In [76]:
```
1 sns.palplot(sns.dark_palette(color="green"))
```

...

In [77]:
```
1 sns.palplot(sns.light_palette(color="green"))
```

...

In [78]:
```
1 sns.get_dataset_names()
```

...

In [80]:
```
1 ir=sns.load_dataset("iris")
2 ir
```

...

In [81]:
```
1 ir.head()
```

...

In [82]:
```
1 ir.tail()
```

...

In [84]:
```
1 ir.shape
```

Out[84]: (150, 5)

In [85]:
```
1 ir.isnull().sum()
```

...

categorical plot

- By default it returns scatter plot
- sns.catplot()

In [87]:
```
1 sns.catplot(x="species",y="sepal_length",data=ir)
```

...

Categorical distribution data

- boxplot

In [88]:
```
1 sns.catplot(x="species",y="sepal_length",data=ir,kind="box")
```

...

In [89]:
```
1 sns.catplot(data=ir,kind='box')
```

...

In [90]:
```
1 sns.catplot(data=ir,kind='box',orient='h')
```

...

In [91]:
```
1 sns.catplot(x="species",y="sepal_length",data=ir,kind="violin")
```

...

In [92]:
```
1 sns.catplot(x="species",y="sepal_length",data=ir,kind="bar")
```

...

jointplot

- it is combination of 2 plots, bydefault scatter plot,histogram
- sns.jointplot()

In [94]:
```
1 sns.jointplot(x="species",y="sepal_length",data=ir,color='g')
```

...

In [95]:
```
1 sns.jointplot(x="sepal_width",y="sepal_length",data=ir,color='g')
```

...

pairplot()

- It displays multiple plots at a time in a single graph
- pairwise relationship
- By default it draws scatter plot
- sns.pairplot()

In [97]:
```
1 sns.pairplot(ir)
```

...

In [98]:
```
1 help(sns.pairplot)
```

...

In [99]:
```
1 sns.pairplot(ir,hue="species")
```

...

In [100]:
```
1 sns.pairplot(ir,hue="species",palette="dark")
```

...

In [ ]:
```
1
```