In [1]:
```
import numpy as np
import pandas as pd
```

1. Create the NumPy array with values (1,2,3).Create a second Numpy array with values (4,5,6). Perform simple arithmetic operations on these two

In [2]:
```
a=np.array([1,2,3])
b=np.array([4,5,6])
```

In [3]:
```
#a. addition
a+b
```

Out[3]:
```
array([5, 7, 9])
```

In [4]:
```
# b. Subtraction
a-b
```

Out[4]:
```
array([-3, -3, -3])
```

In [5]:
```
# c. Multiplication
a*b
```

Out[5]:
```
array([ 4, 10, 18])
```

In [6]:
```
# d. Division
a/b
```

Out[6]:
```
array([0.25, 0.4 , 0.5 ])
```

2. Create a Numpy array between values 10-90 and set the shape to be 3*3

In [7]:
```
matrix=np.arange(10,90,9).reshape(3,3)
```

In [8]:
```
matrix
```

Out[8]:
```
array([[10, 19, 28],
       [37, 46, 55],
       [64, 73, 82]])
```

a. Extract only the 3rd column from the Numpy array.

In [9]:
```
matrix[:,2]
```

Out[9]:
```
array([28, 55, 82])
```

b. Extract only the 2nd row from the Numpy array

In [10]:
```
matrix[1]
```

Out[10]:
```
array([37, 46, 55])
```

c. Extract those cells at the intersection of first two rows and first two columns

In [11]:
```
matrix[:2,:2]
```

Out[11]:
```
array([[10, 19],
       [37, 46]])
```

3. Create a dictionary with 4 items and convert it into panda’s series. Create Dictionary of Pandas series Convert above dictionary into data frame Slice last 2 rows from the data frame Slice a column from the data frame Create dictionary of tuples. Convert it into dataframe.\

In [12]:
```
d=dict({1:"adinath",2:"gurunath",3:"khamkar",4:"developer"})
```

In [13]:
```
ser=pd.Series(d)
```

In [14]:
```
pd.DataFrame(ser)
```

Out[14]:

	0
1	adinath
2	gurunath
3	khamkar
4	developer

In [15]:
```
#Series into the DataFrame
keys=pd.Series([1,2,3,4])
values=pd.Series(["a","b","c","d"])
```

In [16]:
```
df=pd.concat([keys,values],axis=1)
df
```

Out[16]:

	0	1
0	1	a
1	2	b
2	3	c
3	4	d

In [17]:
```
# Slice last 2 rows
df[-2:]
```

Out[17]:

	0	1
2	3	c
3	4	d

In [21]:
```
#dictionary of tuples
d=dict([('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)])
```

In [31]:
```
ser=pd.Series(d)
```

In [36]:
```
df=pd.DataFrame(ser)
```

In [39]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4 entries, Sachin to Rohit
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   0       4 non-null      int64
dtypes: int64(1)
memory usage: 64.0+ bytes
```

4. Explain the use of NumPy library along with the Data structures in NumPy.

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types

In [41]:
```
#datastrutures in numpy
#ndarray
```

5. Explain the use of Pandas library along with the Data structures in Pandas.

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

1. A fast and efficient DataFrame object for data manipulation.
2. Tools for reading and writing data between in-memory data structures and different formats: CSV and text files, Microsoft Excel, SQL databases.
3. Intelligent data alignment and integrated handling of missing data:easily manipulate messy data into an orderly form.
4. Flexible reshaping and pivoting of data sets.
5. Intelligent label-based slicing, fancy indexing, and subsetting of large data sets.
6. Columns can be inserted and deleted from data structures for size mutability.

In [42]:
```
# DataStructure in pandas
# pd.Series()
# pd.DataFrame()
```