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

Q.1 Create a 1D array of numbers from 0 to 9

In [4]:
```
list1 = np.array(range(10))
list1
```

Out[4]:
```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Q.2 Create a 3×3 numpy array of all True's

In [5]:
```
list2 = [True , True ,True]

np.array([[list2]*3])
```

Out[5]:
```
array([[[ True,  True,  True],
        [ True,  True,  True],
        [ True,  True,  True]]])
```

Q.3 Extract all odd numbers from arr Input: arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [8]:
```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
arr[(arr%2 != 0 )]
```

Out[8]:
```
array([1, 3, 5, 7, 9])
```

Q.4 Replace all odd numbers in arr with -1 Input: arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [11]:
```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
arr[arr%2 != 0] = -1

arr
```

Out[11]:
```
array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

Q.5 Convert a 1D array to a 2D array with 2 rows Input: arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [12]:
```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
arr.reshape(2,5)
```

Out[12]:
```
array([[0, 1, 2, 3, 4],
       [5, 6, 7, 8, 9]])
```

Q.NO.1] Define the different ways a DataFrame can be created in pandas?

In [16]:
```
list1 = [1,2,3,4,5]
list2 = [10,20,30,40,50]
list3 = [1.1, 2.1 , 3.1 , 4.1 , 5.1]

pd.DataFrame([list1 , list2, list3] , columns = ['a','b','c','d','e'])
```

Out[16]:

	a	b	c	d	e
0	1.0	2.0	3.0	4.0	5.0
1	10.0	20.0	30.0	40.0	50.0
2	1.1	2.1	3.1	4.1	5.1

In [17]:
```
dict1 = {'Name' : ['Batman','SuperMan', 'Flash' , 'WonderWoman'],
         'Power' : ['Rich' , 'SuperStrong', 'SuperSpeed' , 'Strong and Brave']}

pd.DataFrame(dict1)
```

Out[17]:

	Name	Power
0	Batman	Rich
1	SuperMan	SuperStrong
2	Flash	SuperSpeed
3	WonderWoman	Strong and Brave

Q.NO.2] How will you create an empty DataFrame in Pandas?

In [20]:
```
empty_df = pd.DataFrame(columns = [])
empty_df
```

Out[20]:

Q.NO.3] What are the key features of pandas library ?

Key Features of Pandas according to Tutorials Point

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

In [ ]:
```
Q.NO.4] Write a Pandas program to rename columns of a given DataFrame.

Sample data:
Original DataFrame
col1 col2 col3
0 1 4 7
1 2 5 8
2 3 6 9
New DataFrame after renaming columns:
Column1 Column2 Column3
0 1 4 7
```

```
1 2 5 8
2 3 6 9
```

In [22]:
```
df1 = pd.DataFrame([[1,4,7],[2,5,8],[3,6,9]], columns = ['col1' , 'col2' , 'col3'])
```

In [23]:
```
df1
```

Out[23]:

	col1	col2	col3
0	1	4	7
1	2	5	8
2	3	6	9

In [24]:
```
df1.columns = ['Column1','Column2','Column3']
```

In [25]:
```
df1
```

Out[25]:

	Column1	Column2	Column3
0	1	4	7
1	2	5	8
2	3	6	9

Q.NO.5] Write a Pandas program to convert a dictionary to a Pandas series.

SSample dictionary: d1 = {'a': 100, 'b': 200, 'c':300, 'd':400, 'e':800}

In [26]:
```
d1 = {'a': 100, 'b': 200, 'c':300, 'd':400, 'e':800}

pd.Series(d1)
```

Out[26]:
```
a    100
b    200
c    300
d    400
e    800
dtype: int64
```

In [29]:
```
pd.DataFrame(d1 , index = [0])
```

Out[29]:

	a	b	c	d	e
0	100	200	300	400	800

Q.No.1 Cricketer's scores in five ODI matches are as follows: 12, 34, 45, 50, 24. calculate mean of data using the R and python:

In [31]:
```
list1 = [12,34,45,50,24]

np.mean(list1)
```

Out[31]:
```
33.0
```

Q.No.2 Write down Difference Between Mean, Median and Mode(with Example).

Mean

Mean means to calulate the average of the given list or values .

Median

Median means to find the middle value from the list of given values

Mode

Mode means to find the data who frequency of occurence is the highest

In [ ]: