

```
In [ ]: # Name : Tushar Shirsath
```

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

```
In [8]: # Q.1 Create a 1D array of numbers from 0 to 9
arr = np.arange(10)
arr
```

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

```
In [15]: # Q.2 Create a 3x3 numpy array of all True's
bool_arr = np.ones((3,3), dtype=bool)
bool_arr
```

```
Out[15]: array([[ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True]])
```

```
In [16]: # Q.3 Extract all odd numbers from arr
# Input:
# arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

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

```
Out[16]: (array([1, 3, 5, 7, 9], dtype=int64),)
```

```
In [23]: # Q.4 Replace all odd numbers in arr with -1
# Input:
# arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

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

```
Out[23]: array([-1, -1,  2,  3,  4,  5,  6,  7,  8,  9])
```

```
In [29]: # 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])

arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
arr2 = np.reshape(arr, (2, -1))
arr2
```

```
Out[29]: array([[0, 1, 2, 3, 4],
 [5, 6, 7, 8, 9]])
```

```
In [34]: # Q.NO.1] Define the different ways a DataFrame can be created in pandas?
print('1.By using list method')
x = [['Tushar',25,'Male'], ['Sumit',24,'Male'], ['Twinkle',23,'Female']]
df1 = pd.DataFrame(x, columns=['Name', 'Age', 'Gender'])
print(df1)

print()

print('2.By using dictionary method')
y = {
    'Player': ['Sachin', 'Dhoni', 'Virat'],
    'Team': ['MI', 'CSK', 'BLR'],
    'Score': [50, 100, 200]
}
df2 = pd.DataFrame(y)
print(df2)
```

```
1.By using list method
   Name  Age  Gender
0  Tushar   25    Male
1   Sumit   24    Male
2 Twinkle   23  Female
```

```
2.By using dictionary method
   Player Team  Score
0  Sachin   MI     50
1   Dhoni  CSK    100
2   Virat  BLR    200
```

```
In [39]: # Q.NO.2] How will you create an empty DataFrame in Pandas?
df1 = pd.DataFrame(columns=['Name', 'Age'])
print(df1)

print()

df2 = pd.DataFrame(columns = ['Name', 'Age'], index = ['a', 'b', 'c'])
print(df2)
```

```
Empty DataFrame
Columns: [Name, Age]
Index: []
```

```
   Name  Age
a  NaN  NaN
b  NaN  NaN
c  NaN  NaN
```

```
In [ ]: # Q.NO.3] What are the key features of pandas Library ?

# -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 [49]: # 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

x = [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
df = pd.DataFrame(x, columns=['col1', 'col2', 'col3'])
df.columns = ['column1', 'column2', 'column3']
df
```

```
Out[49]:
```

	column1	column2	column3
0	1	4	7
1	2	5	8
2	3	6	9

```
In [52]: # 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}

d1 = {'a': 100, 'b': 200, 'c':300, 'd':400, 'e':800}
df = pd.Series(d1)
df
```

```
Out[52]:
```

a	100
b	200
c	300
d	400
e	800

dtype: int64

```
In [53]: # 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:

arr = np.array([12,34,45,50,24])
np.mean(arr)
```

```
Out[53]: 33.0
```

```
In [ ]: # Q.No.2 Write down Difference Between Mean, Median and Mode(with Example).

# Mean: The mean is the average where the sum of all the numbers is divided by the total number of numbers,
# Median: The median is the middle value in the list of given numbers numerically ordered from smallest to biggest and
# Mode : The mode is the value of the number which occurs most often in the List.

# Example of Mean : To find the average of the four numbers 2, 4, 6, and 8, we need to add the number first.
# 2 + 4 + 6 + 8 = 20
# Divide the sum by the total number of numbers, i. e 4.
# 20/4 = 5 is the average or mean

# Example of Median: 4, 2, 8, 10, 19.
# Arrange the numbers in ascending order. i .e., 2, 4, 8, 10, 19.
# As the total numbers are 5, so the middle number 8 is the median here.

# Example of Mode: 3, 3, 5, 6, 7, 7, 8, 1, 1, 1, 4, 5, 6.
# Find the frequency of each number.
# For number 3, it's 2. For 5, it's 2. For 6, it's 2. For 7, it's 2. For 8, it's one. For 1, it's 3. For 4, it's 1.
# The number with the highest frequency is the mode. Hence, the mode of the given sequence of numbers is 1.
```

```
In [55]: # Q.No.1 Daily Closing Prices of Two Stocks arranged as per returns. So calculate Covariance using R and Formula.

# x: 1.8 , 1.5, 2.1 , 2.4 , 0.2
# y: 2.5 , 4.3 , 4.5 , 4.1 ,2.2

x = [1.8 , 1.5, 2.1 , 2.4 , 0.2]
y = [2.5 , 4.3 , 4.5 , 4.1 ,2.2]
cov = np.stack((x,y), axis = 1)
np.cov(cov)
```

```
Out[55]: array([[0.245, 0.98 , 0.84 , 0.595, 0.7 ],
 [0.98 , 3.92 , 3.36 , 2.38 , 2.8 ],
 [0.84 , 3.36 , 2.88 , 2.04 , 2.4 ],
 [0.595, 2.38 , 2.04 , 1.445, 1.7 ],
 [0.7 , 2.8 , 2.4 , 1.7 , 2. ]])
```

```
In [ ]: # Q.No.2 Write down Difference Between Covariance and Correlation

# Covariance vs Correlation
# A.Covariance is a measure to indicate the extent to which two random variables change in tandem.
# Correlation is a measure used to represent how strongly two random variables are related to each other.

# B.Covariance is nothing but a measure of correlation.
# Correlation refers to the scaled form of covariance.

# C.Covariance indicates the direction of the linear relationship between variables.
# Correlation on the other hand measures both the strength and direction of the linear relationship between two variables.

# D.Covariance can vary between -∞ and +∞
# Correlation ranges between -1 and +1

# E.Covariance is affected by the change in scale. If all the values of one variable are multiplied by a constant and all the val
# Correlation is not influenced by the change in scale.

# F.Covariance assumes the units from the product of the units of the two variables.
# Correlation is dimensionless, i.e. It's a unit-free measure of the relationship between variables
```

```
In [ ]: 
In [ ]: 
In [ ]: 
In [ ]: 
In [ ]: 
In [ ]: 
In [ ]: 
In [ ]:
```

