In [2]:
```
import pandas as pd
import numpy as np
```

1. Write a Python program to create and display a one-dimensional array-like object containing an array of data using Pandas module

In [3]:
```
a=[1,2,3,4,5]
```

2. Write a Python program to convert a Pandas module Series to Python list and it's type.

In [7]:
```
s=pd.Series([1,2,3,4,5])
```

3. Write a Python program to add, subtract, multiple and divide two Pandas Series.

In [8]:
```
s1=pd.Series([1,2,3,4,5])
s2=pd.Series([1,2,3,4,5])
```

4. Write a Python program to get the largest integer smaller or equal to the division of the inputs series. Sample Series: [2, 4, 6, 8, 10], [1, 3, 5, 7, 9]

In [9]:
```
s1,s2=[2, 4, 6, 8, 10], [1, 3, 5, 7, 9]
```

5. Write a Python program to convert a dictionary to a Pandas series

In [10]:
```
d={'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}
```

6. Write a Python program to convert a NumPy array to a Pandas series.

In [11]:
```
n=np.array([10, 20, 30, 40, 50])
```

7. Write a Python program to change the data type of given a column or a Series to numeric / flot.

In [27]:
```
s1=pd.Series([1,2,'Python',4,5.12])
s1
```

```
0         1
1         2
2    Python
3         4
4      5.12
dtype: object
```

8. Write a Python Pandas program to convert the first column of a DataFrame as a Series.

In [31]:
```
df=pd.DataFrame([{'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800},{'a': 100, 'b': 200, 'c': 300, 'd':
df
```

	a	b	c	d	e
0	100	200	300	400	800
1	100	200	300	400	800

9. Write a Pandas program to convert a given Series to an array.

In [12]:
```
s=pd.Series([1,2,'Python',4,5.12])
```

10. Write a Pandas program to convert Series of lists to one Series.

In [16]:
```
s = pd.Series([['Red', 'Green', 'White'],
               ['Red', 'Black'],
               ['Yellow']])
s
```

```
0    [Red, Green, White]
1          [Red, Black]
2              [Yellow]
dtype: object
```

Expected output

```
0     Red
1     Green
2     White
3     Red
4     Black
5     Yellow
```

11. Write a Pandas program to sort a given Series.

In [13]:
```
s = pd.Series(['100', '200', 'python', '300.12', '400'])
```

12. Write a Pandas program to add some data to an existing Series.

In [34]:
```
s = pd.Series(['100', '200', 'python', '300.12', '400'])
```

13. Write a Pandas program to create a subset of a given series based on value and condition S>2 & <8

In [14]:
```
s = pd.Series([0,1,2,3,4,5,6,7,8,9,10])
```

14. Write a Pandas program to change the order of index of a given series.

In [15]:
```
s = pd.Series(data = [1,2,3,4,5], index = ['A', 'B', 'C','D','E'])
```

15. Write a Pandas program to create the mean and standard deviation of the data of a given Series.

In [ ]:
```
s = pd.Series(data = [1,2,3,4,5,6,7,8,9,5,3])
```

16. Write a Pandas program to get the items of a given series not present in another given series.

In [111]:
```
s1 = pd.Series([1, 2, 3, 4, 5])
s2 = pd.Series([2, 4, 6, 8, 10])
```

17. Write a Pandas program to get the items which are not common of two given series.

In [117]:
```
s1 = pd.Series([1, 2, 3, 4, 5])
s2 = pd.Series([2, 4, 6, 8, 10])
```

18. Write a Pandas & Numpy program to generate a ramdom normorl distrubuted series of 20 no's and compute the minimum, 25th percentile, median, 75th, and maximum of a given series.

19. Write a Pandas & Numpy program to generate a series of 20 ramdom repeated intigers between 0-10 and calculate the frequency counts of each unique value of a that series.

20. Write a Pandas program to display most frequent value in a above series and replace everything else as

'Other' in the series.

21. Write a Pandas & Numpy to generate a series of ramdom integiers of len 10 between 1-10 and program to find the positions of numbers that are multiples of 5 of a given series.

22. Write a Pandas program to extract items at given positions of a given series.

In [215]:
```
num_series = pd.Series(list('2390238923902390239023'))
element_pos = [0, 2, 6, 11, 21]
```

```
Expected output
0     2
2     9
6     8
11    0
21    3
```

23. Write a Pandas program to get the positions of items of a given series in another given series.

In [228]:
```
series1 = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
series2 = pd.Series([1, 3, 5, 7, 9])
```

expected output the position of series2 elements in series1 is
[0, 2, 4, 6, 8]

24. Write a Pandas program convert the first and last character of each word to upper case in each word of a given series.

In [232]:
```
series1 = pd.Series(['php', 'python', 'java', 'c#'])
```

25. Write a Pandas program to calculate the number of characters in each word in a given series.

In [246]:
```
series1 = pd.Series(['Php', 'Python', 'Java', 'C#'])
```

26. Write a Pandas program to compute difference of differences between consecutive numbers of a given series.

In [18]:
```
s1 = pd.Series([1, 3, 5, 8, 10, 11, 15])
```

expected output
[nan, 2.0, 2.0, 3.0, 2.0, 1.0, 4.0]

27. Write a Pandas program to convert a series of date strings to a timeseries.

In [285]:
```
date_series = pd.Series(['01 Jan 2015', '10-02-2016', '20180307', '2014/05/06', '2016-04-12',
                         '2019-04-06T11:20'])
date_series
```

```
0         01 Jan 2015
1          10-02-2016
2            20180307
3          2014/05/06
4          2016-04-12
5    2019-04-06T11:20
dtype: object
```

Expected output

```
0   2015-01-01 00:00:00
1   2016-10-02 00:00:00
2   2018-03-07 00:00:00
3   2014-05-06 00:00:00
4   2016-04-12 00:00:00
5   2019-04-06 11:20:00
dtype: datetime64[ns]
```

28. Write a Pandas program to get the day of month, day of year, week number and day of week from a given series of date strings.

In [295]:
```
date_series1 = pd.Series(['01 Jan 2015', '10-02-2016', '20180307', '2014/05/06', '2016-04-12',
                          '2019-04-06T11:20'])
date_series1
```

```
0         01 Jan 2015
1          10-02-2016
2            20180307
3          2014/05/06
4          2016-04-12
5    2019-04-06T11:20
dtype: object
```

```
Expected output
Day of month:
[1, 2, 7, 6, 12, 6]
Day of year:
[1, 276, 66, 126, 103, 96]
Week number:
[1, 39, 10, 19, 15, 14]
Day of weekday:
```

```
[3, 6, 2, 1, 1, 5]
Day of day name:
['Thursday', 'Sunday', 'Wednesday', 'Tuesday', 'Tuesday', 'Saturday']
```

29. Write a Pandas program to convert year-month string to dates adding a specified day of the month.

In [20]:
```
date_series = pd.Series(['Jan 2015', 'Feb 2016', 'Mar 2017', 'Apr 2018', 'May 2019'])
date_series
```

```
0    Jan 2015
1    Feb 2016
2    Mar 2017
3    Apr 2018
4    May 2019
dtype: object
```

```
Expected Output
New dates:
0   2015-01-11
1   2016-02-11
2   2017-03-11
3   2018-04-11
4   2019-05-11
dtype: datetime64[ns]
```

30. Write a Pandas program to filter words from a given series that contain atleast two vowels.

In [22]:
```
color_series = pd.Series(['Red', 'Green', 'Orange', 'Pink', 'Yellow', 'White'])
color_series
```

```
0       Red
1     Green
2    Orange
3      Pink
4    Yellow
5     White
dtype: object
```

Expected Output

```
Filtered words:
1     Green
2    Orange
4    Yellow
5     White
dtype: object
```

31. Write a Pandas program to compute the Euclidean distance between two given series.

From Wikipedia,
In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space. With this distance, Euclidean space becomes a metric space. The associated norm is called the Euclidean norm.

Expected output
Euclidean distance between two said series:
16.492422502470642

In [23]:
```
x = pd.Series([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
y = pd.Series([11, 8, 7, 5, 6, 5, 3, 4, 7, 1])
```

32. Write a Pandas program to find the positions of the values neighboured by smaller values on both sides in a given series.

In [24]:
```
nums = pd.Series([1, 8, 7, 5, 6, 5, 3, 4, 7, 1])
```

```
Expected output
Positions of the values surrounded by smaller values on both sides:
[1 4 8]
```

33. Write a Pandas program to replace missing white spaces in a given string with the least frequent character.

In [25]:
```
str1 = 'abc def abcdef icd'
```

Expected Output abcidefiabcdefiicd

34. Write a Pandas program to compute the autocorrelations of a given numeric series.

In [26]:
```
num_series = pd.Series(np.arange(15) + np.random.normal(1, 10, 15))
```

```
Expected output like this not the sameone
Autocorrelations of the said series:
[-0.04, -0.37, -0.18, 0.41, 0.43, -0.4, -0.53, 0.67, 0.39, 0.31]
```

35. Write a Pandas program to create a TimeSeries to display all the Sundays of given year.

36. Write a Pandas program to convert given series into a dataframe with its index as another column on the dataframe.

In [31]:
```
char_list = list('ABCDEFGH')
```

Expected Output index 0 0 A 0 1 B 1 2 C 2 3 D 3 4 E 4 5 F 5 6 G 6 7 H 7

37. Write a Pandas program to stack two given series vertically and horizontally.

In [33]:
```
series1 = pd.Series(range(10))
series2 = pd.Series(list('pqrstuvwxy'))
```

```
Stack two given series vertically:
0    0
1    1
2    2
3    3
4    4
5    5
6    6
7    7
8    8
9    9
0    p
1    q
2    r
3    s
4    t
5    u
6    v
7    w
8    x
9    y
dtype: object

Stack two given series horizontally:
   0  1
0  0  p
1  1  q
2  2  r
3  3  s
4  4  t
5  5  u
6  6  v
7  7  w
8  8  x
9  9  y
```

Expected Output Stack two given series vertically: 0 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 0 p 1 q 2 r 3 s 4 t 5 u 6 v 7 w 8 x 9 y dtype: object

Stack two given series horizontally: 0 1 0 0 p 1 1 q 2 2 r 3 3 s 4 4 t 5 5 u 6 6 v 7 7 w 8 8 x 9 9 y

38. Write a Pandas program to check the equality of two given series.

In [34]:
```
nums1 = pd.Series([1, 8, 7, 5, 6, 5, 3, 4, 7, 1])
nums2 = pd.Series([1, 8, 7, 5, 6, 5, 3, 4, 7, 1])
```

```
Expected Output
Check 2 series are equal or not?
0    True
1    True
2    True
3    True
4    True
5    True
6    True
7    True
8    True
9    True
```

39. Write a Pandas program to find the index of the first occurrence of the smallest and largest value of a given series.

In [35]:
```
nums = pd.Series([1, 3, 7, 12, 88, 23, 3, 1, 9, 0])
```

```
Expected Output
Index of the first occurrence of the smallest and largest value of the said series:
9
4
```

40. Write a Pandas program to check inequality over the index axis of a given dataframe and a given series.

In [36]:
```
df_data = pd.DataFrame({'W':[68,75,86,80,None],'X':[78,75,None,80,86], 'Y':[84,94,89,86,86],'Z':[86,97,
sr_data = pd.Series([68, 75, 86, 80, None])
```

```
Expected Output
Original DataFrame:
      W     X   Y   Z
0  68.0  78.0  84  86
1  75.0  75.0  94  97
2  86.0   NaN  89  96
3  80.0  80.0  86  72
4   NaN  86.0  86  83

Original Series:
0    68.0
1    75.0
2    86.0
3    80.0
4     NaN
dtype: float64

Output
```

```
Check for inequality of the said series & dataframe:
       W      X     Y     Z
0  False   True  True  True
1  False  False  True  True
2  False   True  True  True
3  False  False  True  True
4   True   True  True  True
```