Both NumPy and Pandas have emerged to be essential libraries for any scientific computation, inc

```
# Program to create series
import pandas as pd # Import Panda Library
# Program to Create series with scalar values
Data =[1, 3, 4, 5, 6, 2, 9] # Numeric data
# Creating series with default index values
s = pd.Series(Data)
print(s)
# predefined index values
Index =['a', 'b', 'c', 'd', 'e', 'f', 'g']
# Create series with Data, and Index
a = pd.Series(Data, index = Index)
a
```

```
0    1
1    3
2    4
3    5
4    6
5    2
6    9
dtype: int64
a    1
b    3
c    4
d    5
e    6
f    2
g    9
dtype: int64
```

Converting Pandas Series to Python list

```
import pandas as pd
ds = pd.Series([2, 4, 6, 8, 10])
print("Pandas Series")
print(ds)
print("Convert Pandas Series to Python list")
ds.tolist()
```

```
Pandas Series
0     2
1     4
2     6
3     8
4    10
dtype: int64
Convert Pandas Series to Python list
[2, 4, 6, 8, 10]
```

Program to Create Dictionary into Series

```
dictionary ={'a':1, 'b':2, 'c':3, 'd':4, 'e':5, 'f':6}
# Creating series of Dictionary type
sd = pd.Series(dictionary)
sd
```

```
a    1
b    2
c    3
d    4
e    5
f    6
dtype: int64
```

Program to Create ndarray series

```
Data =[[2, 3, 4], [5, 6, 7]] # Defining 2darray
print(Data)
# Creating series of 2darray
snd = pd.Series(Data)
snd
```

```
[[2, 3, 4], [5, 6, 7]]
0    [2, 3, 4]
1    [5, 6, 7]
dtype: object
```

Program to Create DataFrame

```
import pandas as pd
# Import Library
a = pd.DataFrame(Data)
a
```

	0	**1**	**2**
0	2	3	4
1	5	6	7

converting multiple dictionaries into pandas data frame

```
dict1 ={'a':35.2,'b':47,'c':77,'d':49} # Define Dictionary 1
dict2 ={'a':53,'b':69,'c':79,'d':81,'e':91} # Define Dictionary 2
dict3 ={'a':56,'b':98,'c':77,'d':81,'e':90} # Define Dictionary 3
```

```
dict3 ={'a':56,'b':98,'c':77,'d':81,'e':90} # Define Dictionary 3
Data = {'Maths':dict1, 'Physics':dict2,'Chemistry': dict3} # Define Data with dict
df = pd.DataFrame(Data) # Create DataFrame
df
```

	Maths	Physics	Chemistry
a	35.2	53	56
b	47.0	69	98
c	77.0	79	77
d	49.0	81	81
e	NaN	91	90

Program to create Dataframe of three series

```
import pandas as pd
s1 = pd.Series([1, 3, 4, 5, 6, 2, 9])# Define series 1
s2 = pd.Series([1.1, 3.5, 4.7, 5.8, 2.9, 9.3]) # Define series 2
s3 = pd.Series(['a', 'b', 'c', 'd', 'e'])# Define series 3
Data ={'first':s1, 'second':s2, 'third':s3} # Define Data
dfseries = pd.DataFrame(Data)# Create DataFrame
dfseries
```

	first	second	third
0	1	1.1	a
1	3	3.5	b
2	4	4.7	c
3	5	5.8	d
4	6	2.9	e
5	2	9.3	NaN
6	9	NaN	NaN

Program to create DataFrame from 2D array

```
import pandas as pd # Import Library
d1 =[[2, 3, 4], [5, 6, 7]] # Define 2d array 1
d2 =[[2, 4, 8], [1, 3, 9]] # Define 2d array 2
Data ={'first': d1, 'second': d2} # Define Data
df2d = pd.DataFrame(Data)
# Create DataFrame
df2d
```

	first	second
0	[2, 3, 4]	[2, 4, 8]

Converting Pandas Series to Python list

```
import pandas as pd
ds = pd.Series([2, 4, 6, 8, 10])
print("Pandas Series and type")
print(ds)
print(type(ds))
print("Convert Pandas Series to Python list")
print(ds.tolist())
print(type(ds.tolist()))
```

```
Pandas Series and type
0     2
1     4
2     6
3     8
4    10
dtype: int64
<class 'pandas.core.series.Series'>
Convert Pandas Series to Python list
[2, 4, 6, 8, 10]
<class 'list'>
```

Write a Pandas program to add, subtract, multiple and divide two Pandas Series. Sample Series: [2

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 9])
ds = ds1 + ds2
print("Add two Series:")
print(ds)
print("Subtract two Series:")
ds = ds1 - ds2
print(ds)
print("Multiply two Series:")
ds = ds1 * ds2
print(ds)
print("Divide Series1 by Series2:")
ds = ds1 / ds2
print(ds)
```

```
Add two Series:
0     3
1     7
2    11
3    15
4    19
dtype: int64
Subtract two Series:
0    1
1    1
2    1
3    1
4    1
dtype: int64
Multiply two Series:
0     2
1    12
2    30
3    56
4    90
dtype: int64
Divide Series1 by Series2:
0    2.000000
```

Write a Pandas program to compare the elements of the two Pandas Series. Sample Series: [2, 4, 6

```
3    1.142857
4    1.111111
```

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 7, 10])
print("Series1:")
print(ds1)
print("Series2:")
print(ds2)
print("Compare the elements of the said Series:")
print("Equals:")
print(ds1 == ds2)
print("Greater than:")
print(ds1 > ds2)
print("Less than:")
print(ds1 < ds2)
```

```
Series1:
0     2
1     4
2     6
3     8
4    10
dtype: int64
Series2:
0     1
1     3
2     5
3     7
4    10
dtype: int64
Compare the elements of the said Series:
Equals:
0    False
1    False
2    False
3    False
4     True
dtype: bool
Greater than:
0     True
1     True
2     True
3     True
4    False
dtype: bool
Less than:
0    False
1    False
```

Write a Pandas program to compare (equivalance,less than, greater than) the elements of the two [1, 3, 5, 7, 9] and put them all series along with comparisons results into a panda frame

```
dtype: bool
]
```

```
import pandas as pd
ds1 = pd.Series([2, 4, 6, 8, 10])
ds2 = pd.Series([1, 3, 5, 9, 10])

Equals=ds1 == ds2
Greaterthan = ds1 > ds2
Lessthan=ds1 < ds2
# converting pandas series into dictionary and then convert into dataframe
Data={'ds1':ds1,'ds2':ds2,'Equals':Equals,'Greaterthan':Greaterthan,'Lessthan':Les
df1=pd.DataFrame(Data)
df1
```

Hiding the index in panda data frame

```
print(df1.to_string(index= False))
```

```
 ds1  ds2  Equals  Greaterthan  Lessthan
   2    1   False         True     False
   4    3   False         True     False
   6    5   False         True     False
   8    9   False        False      True
  10   10    True        False     False
```

Numpy provides a high-performance multidimensional array and basic tools to compute with and and provides a large number of functions that operate on numpy arrays and are useful for differen applications.

Write a Pandas program to convert a NumPy array to a Pandas series.Sample NumPy array: d1 = [

```
import numpy as np
import pandas as pd
np_array = np.array([1, 2, 3, 4, 5])
print("NumPy array:")
print(np_array)
new_series = pd.Series(np_array)
print("Converted Pandas series:")
print(new_series)
```

```
NumPy array:
[1 2 3 4 5]
Converted Pandas series:
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

Numpyt example program

```
import numpy as np
np.random.seed(0)  # seed for reproducibility

x1 = np.random.randint(100, size=6)  # One-dimensional array
x2 = np.random.randint(100, size=(3, 4))  # Two-dimensional array
x3 = np.random.randint(100, size=(3, 4, 5))  # Three-dimensional array
print(x1)
print(x2)
print(x3)
```

```
[44 47 64 67 67  9]
[[83 21 36 87]
 [70 88 88 12]
 [58 65 39 87]]
[[[46 88 81 37 25]
  [77 72  9 20 80]
  [69 79 47 64 82]
  [99 88 49 29 19]]

 [[19 14 39 32 65]
  [ 9 57 32 31 74]
  [23 35 75 55 28]
  [34  0  0 36 53]]

 [[ 5 38 17 79  4]
  [42 58 31  1 65]
  [41 57 35 11 46]
  [82 91  0 14 99]]]
```

Write a Pandas program to change the data type of the given meric ['100', '200', 'python', '300.12', '4

```
import pandas as pd
s1 = pd.Series(['100', '200', 'python', '300.12', '400a'])
print("Original Data Series:")
print(s1)
print("Change the said data type to numeric:")
s2 = pd.to_numeric(s1, errors='coerce')
print(s2)
```

```
Original Data Series:
0       100
1       200
2    python
3    300.12
4      400a
dtype: object
Change the said data type to numeric:
0    100.00
1    200.00
2       NaN
3    300.12
4       NaN
dtype: float64
```

program to sort a given Series.

```
import pandas as pd
s = pd.Series(['200','100', 'python', '300.12', '400'])
print("Original Data Series:")
print(s)
new_s = pd.Series(s).sort_values()
```

```
print("sorted series are")
print(new_s)
```

```
Original Data Series:
0       200
1       100
2    python
3    300.12
4       400
dtype: object
sorted series are
1       100
0       200
3    300.12
4       400
2    python
dtype: object
```

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

```
import pandas as pd
s = pd.Series(['100', '200', 'python', '300.12', '400'])
print("Original Data Series:")
print(s)
print("\nData Series after adding some data:")
new_s = s.append(pd.Series(['500', 'php']))
print(new_s)
```

```
Original Data Series:
0       100
1       200
2    python
3    300.12
4       400
dtype: object

Data Series after adding some data:
0       100
1       200
2    python
3    300.12
4       400
0       500
1       php
dtype: object
```

Write a Pandas program to create a subset of a given series based on value and condition

```
import pandas as pd
s = pd.Series([0, 1,2,3,4,5,6,7,8,9,10])
print("Original Data Series:")
print(s)
```

```
n = 6
new_s = s[s > n]
print("The values greater than 6")
print(new_s)
new_s1=s[s%2!=0]
print("The odd values ")
print(new_s1)
```

```
Original Data Series:
0      0
1      1
2      2
3      3
4      4
5      5
6      6
7      7
8      8
9      9
10    10
dtype: int64
The values greater than 6
7      7
8      8
9      9
10    10
dtype: int64
The odd values
1    1
3    3
5    5
7    7
9    9
dtype: int64
```

Write a program to create the mean and standard deviation, maximum and minimum of the data o

```
import pandas as pd
s = pd.Series(data = [1,2,3,4,5,6,7,8,9,5,3])
print("Original Data Series:")
print(s)
print("Mean of the said Data Series:")
print(s.mean())
print("Standard deviation of the said Data Series:")
print(s.std())
print("Max")
print(s.max())
print("Min")
print(s.min())
```

```
Original Data Series:
0     1
1     2
2     3
3     4
4     5
5     6
6     7
7     8
8     9
9     5
10    3
dtype: int64
Mean of the said Data Series:
4.818181818181818
Standard deviation of the said Data Series:
2.522624895547565
Max
9
```

Write a program to get the elements of an array values into column-wise using pandas Sample dat [86,97,96,72,83]}

```
import pandas as pd
df = pd.DataFrame({'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,
72,83]});
print(df)
```

```
    X   Y   Z
0  78  84  86
1  85  94  97
2  96  89  96
3  80  83  72
4  86  86  83
```

The previous program's index has to be removed instead convert into the data frame of the subjec

```
df = pd.DataFrame({'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,
72,83], 'Subjects':['Kanada','English','Maths','Science','Social']});
print(df.set_index('Subjects'))
```

```
          X   Y   Z
Subjects
Kanada    78  84  86
English   85  94  97
Maths     96  89  96
Science   80  83  72
Social    86  86  83
```

Write a program to get the columns of the DataFrame phone_data.csv using pandas

The following lines have to be added to include CSV file into colab import io

uploaded = files.upload()

df2 = pd.read_csv(io.BytesIO(uploaded['phone_data.csv']))

Double-click (or enter) to edit

```
import pandas as pd
import numpy as np
import io

from google.colab import files
uploaded = files.upload()
df2 = pd.read_csv(io.BytesIO(uploaded['phone_data.csv']))

print("Columns of the DataFrame:")
print(df2.columns)
df2
df2.shape
```

Choose Files phone_data.csv

- **phone_data.csv**(text/csv) - 40576 bytes, last modified: 18/01/2019 - 100% done

```
Saving phone_data.csv to phone_data (3).csv
Columns of the DataFrame:
Index(['index', 'date', 'duration', 'item', 'month', 'network',
       'network_type'],
      dtype='object')
(830, 7)
```

sorting based on date

```
d_df = df2[['index','date','duration','item','month','network']]
result = d_df.sort_values('network')
print("DataFrame based on network.")
print(result)
result1 = d_df.sort_values('date')
print("DataFrame based on date.")
print(result1)
```

```
DataFrame based on network.
     index            date  duration  item    month network
293    293  25/11/14 16:09       1.0   sms  2014-12  Meteor
430    430  22/12/14 11:22       1.0   sms  2015-01  Meteor
524    524  05/01/15 11:58      99.0  call  2015-01  Meteor
425    425  21/12/14 00:05      54.0  call  2015-01  Meteor
423    423  20/12/14 15:53     553.0  call  2015-01  Meteor
..     ...             ...       ...   ...      ...     ...
370    370  07/12/14 23:22       1.0   sms  2014-12   world
371    371  07/12/14 23:22       1.0   sms  2014-12   world
828    828  14/03/15 00:13       1.0   sms  2015-03   world
361    361  06/12/14 18:28       1.0   sms  2014-12   world
829    829  14/03/15 00:16       1.0   sms  2015-03   world

[830 rows x 6 columns]
DataFrame based on date.
     index            date  duration  item    month   network
504    504  01/01/15 06:58    34.429  data  2015-01      data
673    673  01/02/15 06:58    34.429  data  2015-02      data
674    674  01/02/15 13:33   103.000  call  2015-02  landline
791    791  01/03/15 06:58    34.429  data  2015-03      data
```

Write a Pandas program to display the first 10 rows of the DataFrame.

```
500    500  31/12/14 23:05     1.000   sms  2015-01  Vodafone
```

```
import pandas as pd

#Display the first 10 rows
result = df2.head(10)
print("First 10 rows of the DataFrame:")
print(result)
```

```
First 10 rows of the DataFrame:
   index            date  duration  item    month    network network_type
0      0  15/10/14 06:58    34.429  data  2014-11       data         data
1      1  15/10/14 06:58    13.000  call  2014-11   Vodafone       mobile
2      2  15/10/14 14:46    23.000  call  2014-11     Meteor       mobile
3      3  15/10/14 14:48     4.000  call  2014-11      Tesco       mobile
4      4  15/10/14 17:27     4.000  call  2014-11      Tesco       mobile
5      5  15/10/14 18:55     4.000  call  2014-11      Tesco       mobile
6      6  16/10/14 06:58    34.429  data  2014-11       data         data
7      7  16/10/14 15:01   602.000  call  2014-11      Three       mobile
8      8  16/10/14 15:12  1050.000  call  2014-11      Three       mobile
9      9  16/10/14 15:30    19.000  call  2014-11  voicemail    voicemail
```