In [ ]:

```
# pandas is a Python package providing fast, flexible, and expressive data
# structures designed to make working with “relational” or “labeled” data
# both easy and intuitive. It aims to be the fundamental high-level building
# block for doing practical, real world data analysis in Python. Additionally,
# it has the broader goal of becoming the most powerful and flexible open
# source data analysis / manipulation tool available in any language.
```

In [ ]:

```
import numpy as np #This is the suggested way of importing numpy
import pandas as pd #This is the suggested way of importing pandas
```

In [ ]:

```
print 'Query Help for pandas :', help(pd)
```

In [ ]:

```
#Pandas comes with two components 1. Series 2. Dataframes
```

In [ ]:

```
x = pd.Series([10,20,30,40,50])
```

In [ ]:

```
print 'The series values of x is\n', x
```

In [ ]:

```
y = pd.Series([10,20,30,40,50], dtype = float)
```

In [ ]:

```
print 'The series values of y is\n', y
```

In [ ]:

```
z = pd.Series([10,"Wipro",123.45])
```

In [ ]:

```
print 'The series values of x is\n', z
```

In [ ]:

```
print 'The length of series x is', len(x)
```

In [ ]:

```
print 'The shape of the series x is ', x.shape
```

In [ ]:

```
print 'The type of x, y, z objects are', type(x), type(y), type(z)
```

In [ ]:

```
print 'The data type of x[0],y[0],z[1]', type(x[0]), type(y[0]), type(z[1])
```

In [ ]:

```
a = np.array([10,20,30,40,50])
```

In [ ]:

```
print 'The type of a is :', type(a)
```

In [ ]:

```
x = pd.Series(a)
```

In [ ]:

```
print 'The type of x is ', type(x)
```

In [ ]:

```
print 'The values of series x is\n', x
```

In [ ]:

```
print 'The value of 0th index in series : ', x[0]
```

In [ ]:

```
print 'The range of values in series x is\n', x[:3]
```

In [ ]:

```
print 'The last value of series x is\n', x
```

In [ ]:

```
print 'The first two values of series is\n',x[[0,3]]
```

In [ ]:

```
x = pd.Series([10,20,30,40,50], index =['r1','r2','r3','r4','r5'])
```

In [ ]:

```
print  'The values of series x is\n', x
```

In [ ]:

```
```

```
print 'The value of 0th index in series : ', x['r1']
```

In [ ]:

```
print 'The value of 0th index in series : ', x[0]
```

In [ ]:

```
print 'The values of 0th and 1st index in series is\n',x[['r1','r2']]
```

In [ ]:

```
print  'The values of series x based on condition is\n', x[x>30]
```

In [ ]:

```
print  'The values of series x based on condition is\n', x[x==30]
```

In [ ]:

```
print  'The values of series x based on condition is\n', x[x<30]
```

In [ ]:

```
#x = pd.Series([10,20,30,40,50])
```

In [ ]:

```
x = pd.Series([10,20,30,40,50], index =['r1','r2','r3','r4','r5'])
```

In [ ]:

```
print 'The values of the series x is\n', x
print 'The size of series is ', len(x)
del x['r1']
del x['r4']
print 'The values of the series x is\n', x
print 'The size of series is ', len(x)
```

In [ ]:

```
#Reindexing in case of data manipulation which affects the index
```

In [ ]:

```
x = pd.Series([10,20,30,40,50])
```

In [ ]:

```
print x
```

In [ ]:

```
del x[1]
del x[3]
```

In [ ]:

```
print x
```

In [ ]:

```
#reindexing the missing values of indexes using the reindex function.
#This will fill NaN if the values are not available for the appropriate indexes
y = x.reindex(range(6))
#z = x.reindex(range(6), method='ffill')
```

In [ ]:

```
print y
```

In [ ]:

```
#reindexing the missing values of indexes using the reindex function and fill
#the previous element values if the values are not available for the appropriate
#indexes
z = x.reindex(range(6), method='ffill')
```

In [ ]:

```
print z
```

In [ ]:

```
#Fill values for the NaN with default values like 0 or anything else.
z1 = x.reindex(range(6), fill_value=0)#, inplace = True)
```

In [ ]:

```
print z1
```

In [ ]:

```
print 'The original values of the series is',x
```

In [ ]:

```
#Creating a DataFrame to store tabular values
```

In [ ]:

```

```

```
dict1 = {
        'FirstName' : pd.Series(['Surya','Nagashree','Chandra','Ramya','Keshav', 'Rajeshwar
        'LastName' : pd.Series(['narayana','Ramanath','Sekhar','Narayana','Koushik', 'Sheka
        'Age' : pd.Series([40,36,70,35,12,55,52]),
        'Sex' : pd.Series(['Male','Female','Male','Female','Male','Female','Female']),
        'City' : pd.Series(['Bangalore','Dubai','Mysore','Chennai','Mumbai','Pune','Kolkata
        'Phone' : pd.Series([10,20,30,40,50,60,70])
}
```

In [ ]:

```
print 'The type of the dict1 is : ', type(dict1)
print 'The values of the dict1 object is\n',dict1
```

In [ ]:

```
df = pd.DataFrame(dict1)
```

In [ ]:

```
print 'The values of the DataFrame df is\n',df
```

In [ ]:

```
print 'The shape of the Dataframe object df is : ', df.shape
```

In [ ]:

```
print  'The values of a specific Column in the DataFrame df\n',df['FirstName']
```

In [ ]:

```
print  'The values of a specific Column in the DataFrame df\n',df['FirstName']
```

In [ ]:

```
df['Fullname'] =  df['FirstName'] + df ['LastName']
```

In [ ]:

```
print 'The values of the DataFrame df is\n',df
```

In [ ]:

```
df['Salary']= round(np.random.random()) * df['Age'] * 5000
```

In [ ]:

```
print 'The values of the DataFrame df is\n',df
```

In [ ]:

```
sal = df['Salary'] > 250000
```

In [ ]:

```
print 'The values returned after comparision\n', sal
```

In [ ]:

```
#Try applying functions any(), all(), apply() on numerical columns to evaluate expression
```

In [ ]:

```
print 'The overview of df\n', df.describe()
```

In [ ]:

```
print 'The overview of df\n', df['Age'].describe()
```

In [ ]:

```
print 'The age column of the df DataFrame df is\n', df['Age']
```

In [ ]:

```
print 'Length of DataFrame is:', len(df)
```

In [ ]:

```
print 'The first few default rows of the DataFrame df is\n',  df.head()
```

In [ ]:

```
print 'The last few default rows of the DataFrame df is\n',  df.tail(n=3)
```

In [ ]:

```
print 'The first 3 rows of the DataFrame df is\n', df.head(n=3)
```

In [ ]:

```
#Reading / Loading data from files
#Text File data a) read_csv b) read_table
# Structured Data (JSON, XML, HTML)
#Excel
#Database
```

In [ ]:

```
data = pd.read_csv('C:\\Python27\\wipronew.csv')
```

In [ ]:

```
print 'The type of object data is', type(data)
```

In [ ]:

```
print 'The contents of the loaded data is\n', data
```

In [ ]:

```
print 'Length of the loaded data is : ', len(data)
```