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

In [6]:
```
1 help(pd)
```

```
Help on package pandas:

NAME
    pandas

DESCRIPTION
    pandas - a powerful data analysis and manipulation library for Python
    =====================================================================

    **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**. It is already well on
    its way toward this goal.

    Main Features
    -------------
    Here are just a few of the things that pandas does well:

      - Easy handling of missing data in floating point as well as non-floating
        point data.
      - Size mutability: columns can be inserted and deleted from DataFrame and
        higher dimensional objects
      - Automatic and explicit data alignment: objects can be explicitly aligned
        to a set of labels, or the user can simply ignore the labels and let
        `Series`, `DataFrame`, etc. automatically align the data for you in
        computations.
      - Powerful, flexible group by functionality to perform split-apply-combine
        operations on data sets, for both aggregating and transforming data.
      - Make it easy to convert ragged, differently-indexed data in other Python
        and NumPy data structures into DataFrame objects.
      - Intelligent label-based slicing, fancy indexing, and subsetting of large
        data sets.
      - Intuitive merging and joining data sets.
      - Flexible reshaping and pivoting of data sets.
      - Hierarchical labeling of axes (possible to have multiple labels per tick).
      - Robust IO tools for loading data from flat files (CSV and delimited),
        Excel files, databases, and saving/loading data from the ultrafast HDF5
        format.
      - Time series-specific functionality: date range generation and frequency
        conversion, moving window statistics, date shifting and lagging.
```

```
PACKAGE CONTENTS
    _config (package)
    _libs (package)
    _testing
    _typing
    _version
    api (package)
    arrays (package)
    compat (package)
    conftest
    core (package)
    errors (package)
    io (package)
    plotting (package)
    testing
    tests (package)
    tseries (package)
    util (package)

SUBMODULES
    _hashtable
    _lib
    _tslib
    offsets

FUNCTIONS
    __getattr__(name)

DATA
    IndexSlice = <pandas.core.indexing._IndexSlice object>
    NA = <NA>
    NaT = NaT
    __docformat__ = 'restructuredtext'
    __git_version__ = 'db08276bc116c438d3fdee492026f8223584c477'
    describe_option = <pandas._config.config.CallableDynamicDoc object>
    get_option = <pandas._config.config.CallableDynamicDoc object>
    options = <pandas._config.config.DictWrapper object>
    reset_option = <pandas._config.config.CallableDynamicDoc object>
    set_option = <pandas._config.config.CallableDynamicDoc object>

VERSION
    1.1.3
```

```
FILE
    c:\programdata\anaconda3\lib\site-packages\pandas\__init__.py
```

In [ ]:
```
help(pd.Series)
```

In [11]:
```
friends = ['VHA','KMS','MVP','MVK','TAT']
x=pd.Series(friends)
print(x)
print(type(x))
```

```
0    VHA
1    KMS
2    MVP
3    MVK
4    TAT
dtype: object
<class 'pandas.core.series.Series'>
```

In [17]:
```
f=[10,9,9,8,5.5]
x=pd.Series(f)
print(x)
print(type(x))

```

```
0    10.0
1     9.0
2     9.0
3     8.0
4     5.5
dtype: float64
<class 'pandas.core.series.Series'>
```

In [21]:
```
# this is using list
f=[10,9,9,8,5]
sub=["python","FSD","COA","TOC","DM"]
x=pd.Series(f,index=sub)
print(x)
print(type(x))
```

```
python    10
FSD        9
COA        9
TOC        8
DM         5
dtype: int64
<class 'pandas.core.series.Series'>
```

In [26]:
```
#Using dict
m={"python":10,"FSD":9,"COA":9,"TOC":8,"DM":8}
x=pd.Series(m)
print(x)
print(type(x))
```

```
python    10
FSD        9
COA        9
TOC        8
DM         8
dtype: int64
<class 'pandas.core.series.Series'>
```

In [27]:
```
#Using dict
m={"python":10,"FSD":9,"COA":9,"TOC":8,"DM":8}
x=pd.Series(m,index=["a","b","c","d","e"])
print(x)
print(type(x))
```

```
a   NaN
b   NaN
c   NaN
d   NaN
e   NaN
dtype: float64
<class 'pandas.core.series.Series'>
```

In [31]:
```
y=pd.read_csv("Book2.csv")
print(y)
print(type(y))
```

```
   roll   FSD  Python
0     1   20      22
1     2   21      18
2     3   23      14
3     5    9      19
<class 'pandas.core.frame.DataFrame'>
```

In [43]:
```
# this is using list
roll=[[1,2,3,4,5],[10,10,10,10,9],[10,10,9,8,8]]
columns=["roll","FSD","PYTHON"]
x=pd.DataFrame(roll,columns)
print(x)
print(type(x))
```

```
         0   1   2   3  4
roll     1   2   3   4  5
FSD     10  10  10  10  9
PYTHON  10  10   9   8  8
<class 'pandas.core.frame.DataFrame'>
```

In [46]:
```
f={'roll':[1,2,3,4,5],'FSD':[10,10,10,9,8],'Python':[10,10,8,9,10]}
x=pd.DataFrame(f)
print(x)
```

```
   roll  FSD  Python
0     1   10      10
1     2   10      10
2     3   10       8
3     4    9       9
4     5    8      10
```

In [52]:
```
f={'roll':[1,2,3,4,5],'FSD':[10,10,10,9,8],'Python':[10,10,8,9,10]}
x=pd.DataFrame(f)
x=x.set_index("roll")
print(x)
```

```
      FSD  Python
roll
1      10      10
2      10      10
3      10       8
4       9       9
5       8      10
```

In [54]:
```
f={'roll':[1,2,3,4,5],'FSD':[10,10,10,9,8],'Python':[10,10,8,9,10]}
x=pd.DataFrame(f)
x.set_index("roll",inplace=True)
print(x)
```

```
      FSD  Python
roll
1      10      10
2      10      10
3      10       8
4       9       9
5       8      10
```

In [ ]:
```
1
```

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

In [5]:
```
df=pd.read_csv('Visha.csv')
df.dtypes
print(df)
```

```
   Roll  FSD  Python  COA   TOC
0     1  NaN    10.0  8.0  10.0
1     2  6.0    10.0  7.0   8.0
2     3  NaN     NaN  NaN   NaN
3     4  6.0    10.0  7.0   8.0
4     5  2.0    10.0  5.0   3.0
5     6  7.0    10.0  7.0   5.0
```

In [ ]:
```

```