Pandas

Series

```
labels = ['Prashant','Ishan','Samman']
my_list = [1,4,5]
arr = np.array([1,4,5])
d = {'Prashant':1,'Ishan':4,'Samman':5}
```

using List

`pd.Series(data=my_list)`

```
print("Familarity With Python")
pd.Series(data=my_list,index=labels)
```

Using Numpy Array

```
pd.Series(data = arr)
pd.Series(arr,labels)
```

Using Dictionary

```
1. ser = pd.Series(d)
ser['Prashant']
2. ser1 = pd.Series([1,2,3,4],index = ['USA', 'Germany','USSR', 'Japan'])
ser2 = pd.Series([1,2,5,4],index = ['USA', 'Germany','Italy', 'Japan'])
ser1+ser2
```

`ser1.add(ser2)`

DataFrame

```
1 df = pd.DataFrame(randn(5,4),index=('A', 'B' ,'C','D','E'),columns='W X Y
Z'.split())
2 type(df)
3 type(df['W'])
4 df['new'] = df['W'] + df['Y']
5. df.drop('new',axis=1)
6. use of inplace
df.drop('new',axis=1,inplace=True)
```

axis=0 axis=1

`df.drop('C',axis=0)`

Selecting Data

```
1. df.index
2. df.loc['A']
df.loc['B','Y']
df.loc['A':'D','W':'Y']
3. df.loc[df.W>0,:]
```

df.iloc()

```
1.df.iloc[2]
2.df.iloc[[2,4],1]
3.list(df.W >0)
4.df.iloc[list(df.W >0),:]
```

Operations

```
1.df>0
df[df>0]
df[df['W']>0]['Y']
df[(df['W']>0) & (df['Y'] <0)]
2 df.reset_index(inplace = True)
df.set_index('index',inplace=True)
3.df = pd.DataFrame({'CE':[1,2,0],
                     'CS':[5,0,2],
                     'EE':[11,20,0],
                     'Pharmacy':[23,15,np.nan]},index=
['C++','Math','Electrical'])
                     df.dropna()
                     df.dropna(axis=1)
```

Filling Values

```
1.df.fillna(value='FILL VALUE')
2.df['Pharmacy'].fillna(value=df['Pharmacy'].mean(),inplace=True)
3.data = {'Subject':['DBMS','DBMS','MCSC','MCSC','DIFF','DIFF'],
       'Instructor':['Santosh','Nabin','Gokul','Ram','Saraswoti','Khim'],
       'Assignments':[10,2,4,5,6,4]}
df = pd.DataFrame(data)
by_comp = df.groupby('Subject')
by_comp

by_comp.mean()
df.head()
4.df['Subject'].unique()
5.df['Instructor'].nunique()
6.df['Subject'].value_counts()
```

```
df = pd.DataFrame({'col1':[1,2,3,4],'col2':[444,555,666,444],'col3':
['abc','def','ghi','xyz']})
df.head()
def times2(x):
    return x*2
df.apply(times2)
df.isnull()
```