Installation

conda install pandas or pip install pandas

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

```
pd.__version__
```

Series

```
s = pd.Series(np.random.randn(6))
s
```

```
s.values
```

```
s.index
```

```
s.index = [1, 2, 3, 4, 5, 6]
s
```

```
s = pd.Series(np.arange(1, 6), index=["a", "b", "c", "d", "e"])
s
```

```
s["d"]
```

```
s[["b", "d"]]
```

```
s["b":]
```

```
s > 2
```

```
s[s > 2]
```

```
s * 3
```

```
s
```

```
# Series from dictionary
s_d = pd.Series({'name': 'sandip', 'age': 20, 'class': 14 })
```

```
s_d
```

```
# Series from list
s_l = pd.Series(['Kathmandu', 'Pokhara', 'Butwal', 'Biratnagar', 'Chitwan'])
s_l
```

DataFrame

```
# Create data frame from reading files
df = pd.read_csv('supermarket_sales.csv')
```

Reading other types of files

pd.read_table(filename) | Reading a tab separated file.
pd.read_excel(filename) | Reading an excel file.

Converting back to csv or excel

df.to_csv(filename) | Back to csv
df.to_excel(filename) | Back to excel

```
# First five items
df.head()
```

```
# Last five
df.tail()
```

```
# Information about the dataframe
df.info()
```

```
# Some mathematical descriptions
df.describe()
```

```
df.dtypes
```

```
df.isnull().any()
```

```
df['Total'].isnull().any()
```

```
df['Invoice ID'].isnull().any()
```

DataFrame from lists

```
l_name = ['Sandip', 'Prerna', 'Aakash', 'Nisha']
l_address =['Kathamandu', 'Pokhara', 'Butwal', 'Chitwan']

# df = pd.DataFrame(zip(l_name, l_address), columns=['name', 'address'])
df = pd.DataFrame(zip(l_name, l_address))
df.columns = ['name', 'address']
```

```
df
```

```
# With index
l_name = ['Sandip', 'Prerna', 'Aakash', 'Nisha']
l_address =['Kathamandu', 'Pokhara', 'Butwal', 'Chitwan']

df = pd.DataFrame(zip(l_name, l_address), columns=['name', 'address'], index=['a','b','c','d'])
```

```
# From 2d list
l = [['Sandip', 'Kathmandu', 22], ['Prerna', 'Pokhara', 18],
     ['Aakash', 'Butwal', 23], ['Nisha', 'Chitwan', 19]]

df = pd.DataFrame(l, columns=['name', 'address', 'age'])
df
```

Dataframe from series

```
# Could use zip name as creating dataframe from list
```

```
s1 = pd.Series(l_name)
s2 = pd.Series(l_address).transpose()
df = pd.DataFrame([s1, s2 ]).transpose()
df.columns = ['name', 'address']
```

```
df
```

Using concat

```
df = pd.concat([s1, s2], axis=1)
df
```

```
# axis=0 is default
df_row = pd.concat([s1, s2])
df_row
```

From Dictionary

```
d = {'name': ['Sandip', 'Prerna', 'Aakash', 'Nisha'],
     'address': ['Kathamandu', 'Pokhara', 'Butwal', 'Chitwan'],
     'age': [22, 18, 20, 19]}

df = pd.DataFrame(d)
# df.index = ['a', 'b', 'c', 'd']
```

```
df
```

```
# Checking the datatypes
df.dtypes
```

```
df.index
```

```
df.columns
```

```
df.shape
```

```
df.index = ['one', 'two', 'three', 'four']
```

```
# Getting one column
# df['name']
df.name
```

Deleting or removing a column or whole frame

```
# Copying a frame
df_copy = df.copy()
```

```
df_copy
```

```
df
```

```
df.loc[0, ['address', 'age']]
```

```
# Deleting a series
del df_copy['name']
df_copy
```

```
del df_copy
```

```
df_copy
```

```
df
```

```
df.drop('age', axis=1)
```

```
# Dropping a series or multiple series from dataframe
df.drop('age', axis=1) # Dropping a column requires axis to be mentioned
df.drop(['name', 'age'], axis=1)
```

```
df
```

```
# Dropping a row  or multiple rows
df.drop('one')
df.drop(['two', 'three'])
```

```
df_copy = df.copy()
df_copy
```

```
# Removes permanently the given series from dataframe
df_copy.drop('name', axis=1, inplace=True)
```

```
df_copy
```

```
df
```

```
df['email'] = ['sandip@gmail.com', 'prerna@gmail.com', 'aakash@gmail.com', 'nisha@gmail.com']
```

```
df
```

```
df.loc[:, 'name']
```

Multiple rows and columns

```
# Getting more than one columns
df.loc[:, ['name', 'address']]
```

```
# Multiple columns without loc
df[['name', 'address']]
```

```
# Multiple row and columns with iloc
df.iloc[0, 0:2]
```

```
df.iloc[0:3, 0:3]
```

```
# Reindexing with rows
df1 = df.reindex['one', 'two', 'three', 'four', 'five'])
df1
```

```
df
```

```
# Reindexing with columns
df2 = df.reindex(columns = ['address', 'age', 'name'])
df2
```

```
# Adding a new column
df2['email'] = ['sandip@gmail.com', 'prerna@gmail.com', 'aakash@gmail.com', 'nisha@gmail.com']
df2
```

```
# Values of a dataframe
df2.values
type(df2.values)
```

```
df2
```

Indexing and Filtering

```
df2[:2] # Slicing
```

```
df2[['address', 'age']]
```

```
df2[df2['age'] > 18]
```

```
df2['age'] > 19
```

```
df2
```

Sorting

```
df
```

```
df2.sort_values(by='age') # Sort a dataframe in ascending order with one column
```

```
df2.age.sort_values(ascending=False) # Sort a series in descending order
```

```
df3 = df2.copy()
```

```
# Any change in new data frame affects the original
# To assign a copy
df4 = df3
```

```
df4['sex'] = ['Male', 'Female', 'Male', 'Female']
```

```
df4
```

```
df3
```

Basic statistics

```
df = pd.read_csv('supermarket_sales.csv')
```

```
df['Total']
```

```
df.tail()
```

```
df.describe()
```

```
# The maximum sale
df.Total.max()
```

```
# Minimum sale
df.Total.min()
```

```
# Sum of sales
df.Total.sum()
```

```
df
```

```
# Mean of sales
df.Total.mean()
```

Groupby mechanism

```
df1 = df.groupby('City')
df1
```

```
# Each city with it's dataframe
for city, city_df in df1:
    print(city)
    print(city_df)
```

```
# Get individual item dataframe
df1.get_group('Mandalay')
```

```
df1.Total.max()
```

```
df1.Total.min()
```

```
df1.describe()
```

```
df1.Total.agg(['count', 'min', 'max', 'mean'])
```

```
df[df.City == 'Mandalay'].Total.max()
```

```
arr = np.arange(1,10)
np.where(arr == 2)
```

```
```