

Pandas (Day-2)

↳ It is a powerful data manipulation library used for data analysis & data cleaning.

↳ It provides 2 primary data structures

- ① Series
- ② Data frames.

* Import pandas as pd

Series

↳ It is a 1-dimensional array-like object that can hold any data type.

```
import pandas as pd
```

```
data = [1, 2, 3, 4, 5]
```

```
series = pd.Series(data)
```

```
Print (type (series)).
```

Create series from dictionary

```
data = {'a': 1, 'b': 2, 'c': 3}
```

```
Series_dict = pd.Series(data)
```

```
print(Series_dict)
```

Create series from two data using index

```
data = [1, 2, 3]
```

```
index = ['a', 'b', 'c']
```

```
pd.Series(data, index=index)
```

Data frame:

Create a dataframe from a dictionary

```
data = {  
    'Name': ['Krish', 'John', 'Jack'],  
    'Age': [25, 30, 45],  
    'City': ['Bangalore', 'New York', 'Florida']  
}
```

~~df = pd.DataFrame~~

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

```
print(df)
```

```
print(type(df))
```


Create a Data-frame from a list of Dictionary

```
data = [
    {'Name': 'Krish', 'Age': 32, 'City': 'Bangalore'},
    {'Name': 'John', 'Age': 24, 'City': 'Goa'}
]
```

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

```
Print(df)
```

```
Print(type(df))
```

Creating a Data

reading the data from the csv file

```
df = pd.read_csv('file name: csv')
```

```
df.head(5) ——— top 5 elements
```

```
df.tail(5) ——— last 5 elements
```

Adding a column

```
df['salary'] = [50000, 40000, 25000]
```

```
df
```

Remove a column

```
df.drop('salary', axis=1) — temporary
```

Accessing data from data frame

```
df
```

```
df['Name']
```

```
df.loc[0]
```

```
df.iloc[0]
```

Accessing at specific element

```
df.at[2, 'Age']
```

```
df.at[2, 'Name']
```

Accessing a specified element using "iat"

```
df.iat[2, 2]  
'Florida'
```

df.drop('Salary', axis=1, inplace=True)

L) permanently,

Add age to the Column

df['Age'] = df['Age'] + 1

df

df = pd.read_csv('filename.csv')

df.head(5)

print(df.dtypes) — data type

print(df.describe()) — statistical summary

['Age'] to 1b

['Salary'] to 1b

['Age'] to 1b

['Salary'] to 1b

['Age'] to 1b

['Salary'] to 1b

['Age'] to 1b

['Age'] to 1b

Data Manipulation (DAY-3)

* Handling missing values

`df.isnull().any()` — if there are any missing values present in the attribute, it will print 'True'.

`df.isnull().sum()` — if there are any missing values present in that column, it will print the count.

`df.fillna(0)` — if there are any 'na' present in that place, it will modify to '0'.

* Filling missing values with mean of the column.
`df['Sales'].fillna(df['Sales'].mean())`

`df`

* Renaming Columns:

`df = df.rename(columns = {'Sale Date 1': 'Sales Date'})`

`df.head(5)`

* Change datatype

`df['Value - new'] = df['Value'].astype(int)`

`df.head()`

```
df['New Value'] = df['value'].apply(lambda x: x*2)
```

```
df.head(5)
```

* Data Aggregating & Grouping

```
df.head(5)
```

```
group-mean = df.groupby('Product')['value'].mean()
```

```
Print(group-mean)
```

```
group-sum = df.groupby(['Product', 'Region'])['value'].sum()
```

```
Print(group-sum)
```

Aggregate multiple functions

```
Group-agg = df.groupby('Region')['value'].agg(['mean', 'sum', 'count'])
```

```
group-agg
```


Merging and joining Data frame

df1 = pd.DataFrame({ 'key': ['A', 'B', 'C'], 'value1': [1, 2, 3] })

df2 = pd.DataFrame({ 'key': ['A', 'B', 'D'], 'value2': [4, 5, 6] })

df1

df2

pd.merge(df1, df2, on = "key", how = "inner")

Common of key ↑ Print which common only ↑

Print all that are present in both data frames

pd.merge(df1, df2, on = "key", how = "outer")

how = "left" (Print which are common in both and left dataframe)

'''

'''

how = "right"

Print the right data frame and common value of both

'''

'''