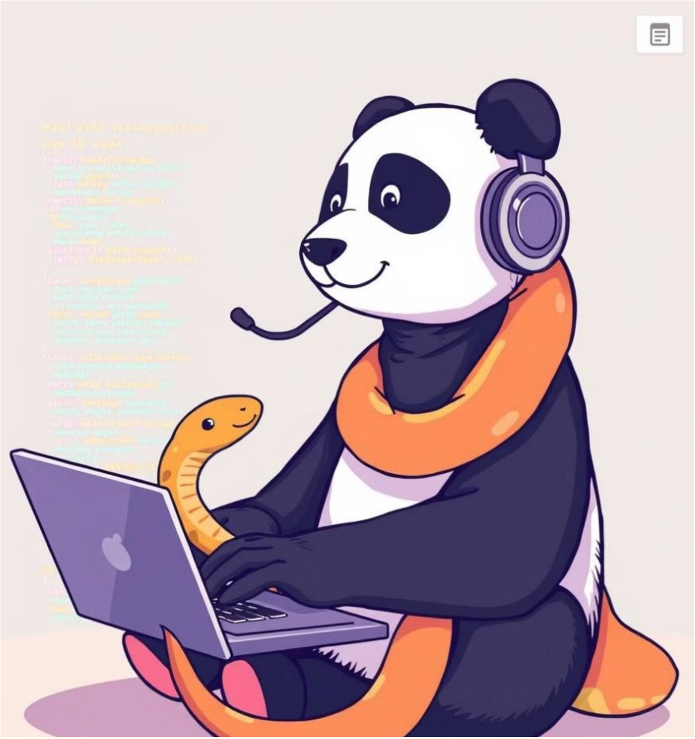
Unit:4; Class:1, 2

Pandas and NumPy

Instructor: Musfique Ahmed

What is Pandas?

- Pandas is a powerful Python library for data manipulation and analysis.
- Built on top of NumPy.
- Provides easy-to-use data structures: **Series** and **DataFrame**.
- Commonly used in data science, machine learning, and data analysis.



Installing Pandas:

```
pip install pandas
```

- Importing Pandas:

```
import pandas as pd
```

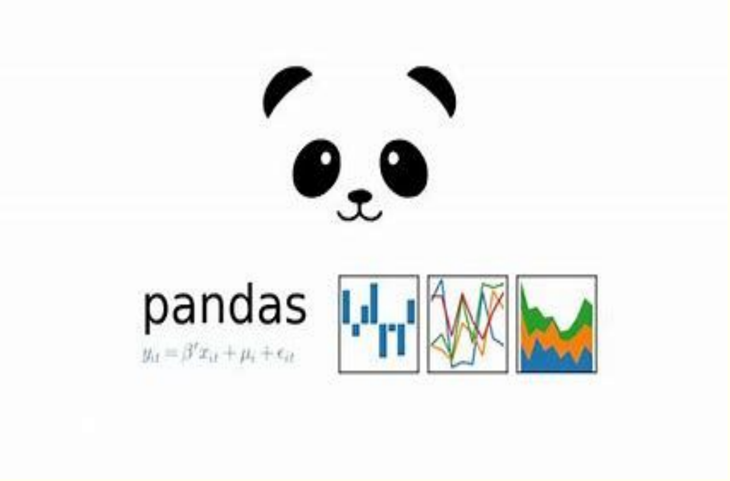
Pandas Data Structures

1. Series (1D Array-like object)

```
import pandas as pd
s = pd.Series([10, 20, 30, 40])
print(s)
```

2. DataFrame (2D Table-like structure)

```
data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}
df = pd.DataFrame(data)
print(df)
```



Creating DataFrames

- From dictionaries
- From lists of lists
- From CSV files
- From Excel files
- From NumPy arrays

```
df = pd.DataFrame({’A’: [1, 2, 3], ’B’: [4, 5, 6]})
```

Basic DataFrame Operations

- `df.head()` – View the first few rows.
- `df.tail()` – View the last few rows.
- `df.info()` – Summary of the DataFrame.
- `df.describe()` – Summary statistics.
- `df.shape` – Number of rows and columns.
- `df.columns` – Column names.
- `df.dtypes` – Data types of each column.

```
print(df.head())
print(df.info())
```

Selecting Data

- Selecting a single column: `df['column_name']`
- Selecting multiple columns: `df[['col1', 'col2']]`
- Selecting rows using `.loc[]` and `.iloc[]`

```
print(df.loc[0])  # Select row by label
print(df.iloc[0])  # Select row by index
```

Filtering Data

Using conditions to filter data.

```
filtered_df = df[df[’Age’] > 25]
print(filtered_df)
```

Modifying Data

- Adding a new column.
- Updating values.
- Removing columns and rows.

```
df['New_Column'] = df['Age'] * 2
print(df)
df.drop('New_Column', axis=1, inplace=True)
```

Handling Missing Values

- Checking for missing values: `df.isnull().sum()`
- Filling missing values: `df.fillna(value)`
- Dropping missing values: `df.dropna()`

```
df.fillna(0, inplace=True)
df.dropna(inplace=True)
```

Grouping and Aggregation

- `groupby()` for grouping data.
- Aggregation functions: `mean()`, `sum()`, `count()`, `max()`, `min()`

```
grouped_df = df.groupby('Age').sum()
print(grouped_df)
```

Sorting and Reordering

- Sorting DataFrame by values.

```
df.sort_values(by='Age', ascending=False)
```

Working with CSV & Excel

- Reading CSV files: `pd.read_csv('file.csv')`
- Writing to CSV: `df.to_csv('output.csv', index=False)`
- Reading Excel: `pd.read_excel('file.xlsx')`
- Writing to Excel: `df.to_excel('output.xlsx', index=False)`

```
df.to_csv('output.csv', index=False)
```

Thank You

Do the Quiz Please, you have 20 minutes to do that!