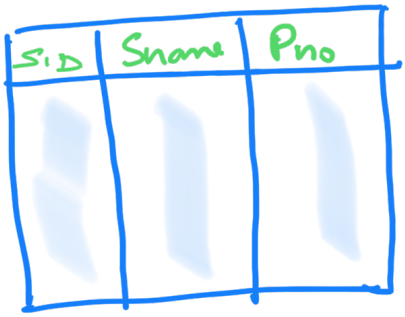
Exploring Pandas - Part 1

Module 1: Introduction to Pandas

- 1. What is Pandas?
- 2. Why Pandas?
- 3. Installing and Importing Pandas
- 4. Pandas Data Structures
- 5. Reading Data from Files



1) What is Pandas?

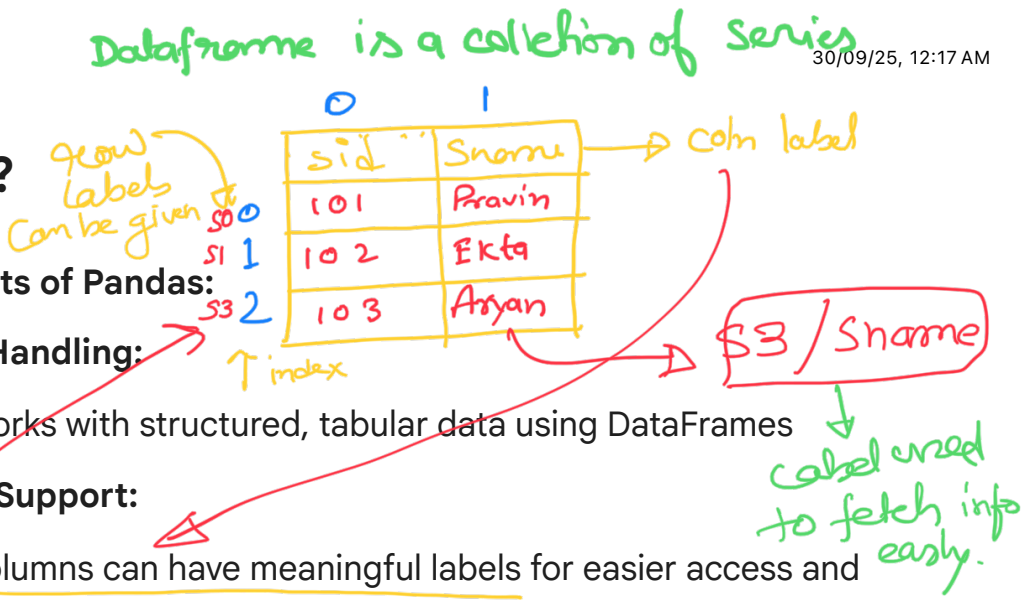
Panda is build on top of numPy.

- Pandas stands for Panel Data
- Pandas is a powerful Python library used for:
  - Data manipulation
  - Data cleaning
  - Data analysis
- Pandas provides 2 Data Structures:
  - **Series** – 1D labeled array (like a column)
  - **DataFrame** – 2D labeled data structure (like a spreadsheet or SQL table)
- Pandas is Built on top of NumPy
- Pandas Integrates well with Matplotlib, Seaborn, and Scikit-learn

all operation we have studied in Numpy will work with series

- So you won't use series DS directly. But if you Take the above Table example: sID is a Series DS

2) Why Pandas?



Key Features & Benefits of Pandas:

- **a) Tabular Data Handling:**
  - Efficiently works with structured, tabular data using DataFrames
- **b) Labeled Data Support:**
  - Rows and columns can have meaningful labels for easier access and manipulation
- **c) Powerful Data Selection & Filtering:**
  - Easy row/column slicing, filtering, and conditional operations
- **d) GroupBy & Aggregation:**
  - Supports group-based operations for summarizing and transforming data
- **e) Missing Data Handling:**
  - Built-in support for detecting, filling, or dropping missing values
- **f) Data Cleaning & Transformation:**
  - Tools for reshaping, merging, joining, replacing, and converting data
- **g) Time Series Support:**
  - Built-in tools for date/time parsing, resampling, and frequency conversion
- **h) High Performance:**
  - Optimized for speed and memory efficiency, built on top of NumPy
- **i) Interoperability:**
  - Works seamlessly with libraries like NumPy, Matplotlib, Seaborn, and Scikit-learn
- **j) I/O Operations:**
  - Read and write data from various formats (CSV, Excel, JSON, SQL, etc.) with ease

3) Installing and Importing Pandas

- **a) Install Pandas**
  - If Pandas is not already installed in your environment, use the following command:

```
pip install pandas
```

- If you're using Jupyter Notebook or Google Colab, just run:

```
!pip install pandas
```

- **b) Import Pandas**
  - After installation, import NumPy using the standard alias:

```
import pandas as pd
```

  - Conventionally, pd is the alias used for Pandas.

- **c) Check Pandas Version**

```
import pandas as pd

print(pd.__version__)
```

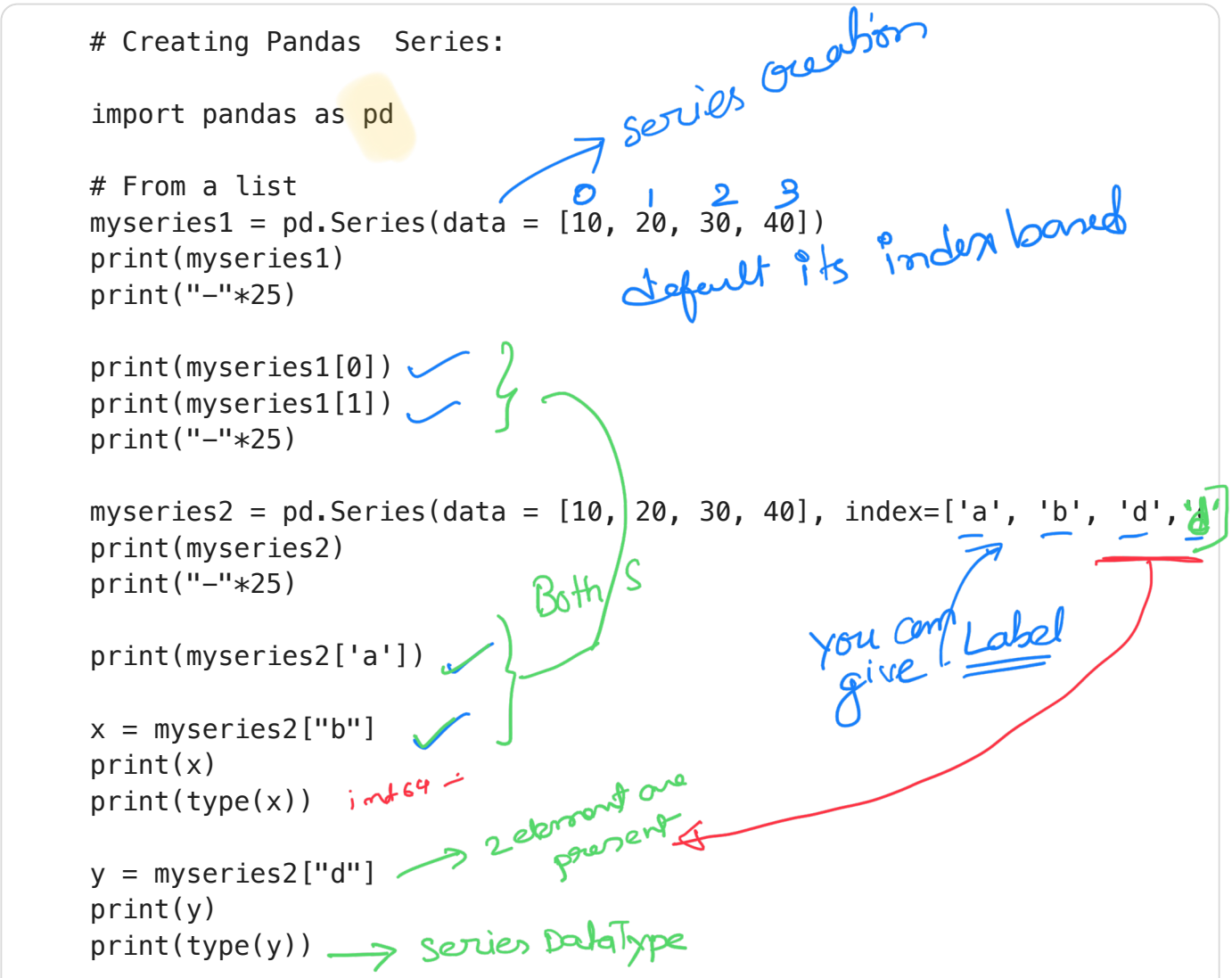
```
2.2.2
```

4) Pandas Data Structures

- Pandas provides 2 Data Structures:
  - Pandas Series
  - DataFrame

- **Pandas Series**
  - One-dimensional labeled array
  - Think of it like a single column in Excel or a single column of data in a table

```
# Creating Pandas  Series:

import pandas as pd

# From a list
myseries1 = pd.Series(data = [10, 20, 30, 40])
print(myseries1)
print("-"*25)

print(myseries1[0])
print(myseries1[1])
print("-"*25)

myseries2 = pd.Series(data = [10, 20, 30, 40], index=['a', 'b', 'd', 'd'])
print(myseries2)
print("-"*25)

print(myseries2['a'])

x = myseries2["b"]
print(x)
print(type(x))

y = myseries2["d"]
print(y)
print(type(y))
```




```
0    10
1    20
2    30
3    40
dtype: int64
-------------------------
10
20
```

```
------------------------
a    10
b    20
d    30
d    40
dtype: int64
------------------------
10
20
<class 'numpy.int64'>
d    30
d    40
dtype: int64
<class 'pandas.core.series.Series'>
```

- **Pandas DataFrame** → only 2D – X 3D
  - Two-dimensional labeled data structure
  - Similar to an Excel sheet or SQL table with rows and columns
  - DataFrame = Collection of Series

```
# Creating Pandas  DataFrame:

import pandas as pd

# From a dictionary of lists
mydata = {
  'Name': ['Hello', 'Sri', 'Vas'],
  'Age': [25, 30, 35]
}

df = pd.DataFrame(mydata)

df
```

For Test we always create dictionary

Coln Label

is Numpy array internally

	Name	Age
0	Hello	25
1	Sri	30
2	Vas	35

Next steps: Generate code with df | New interactive sheet

5) Reading Data from Files

- You can read data from files and create DataFrame

```
import pandas as pd

df = pd.read_csv("myjlc.csv")

df
```

Panda provides multiple methods to read data from different files.

	sid	sname	phone
0	101	Pravin	8884766830
1	102	Ekta	9353791014
2	103	Aryan	123456
3	104	Dhrub	22221

Note: First row in csv file always consider as coln Name.

Next steps: Generate code with df | New interactive sheet

```
# Requires openpyxl library for working with Excel

!pip install openpyxl
```

```
Requirement already satisfied: openpyxl in /usr/local/lib/python3.12/di
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.12/
```

```
import pandas as pd

df = pd.read_excel("myjlc.xlsx")

df
```

	sid	sname	phone
0	101	Pravin	8884766830
1	102	Ekta	9353791014
2	103	Aryan	123456
3	104	Dhrub	22221

Next steps: Generate code with df | New interactive sheet

```
import pandas as pd

df = pd.read_csv("myjlc1.csv")

df
```

Dataframes is like a Table.

Row index

	sid	sname	phone
0	101	Pravin	8884766830
1	102	Ekta	9353791014
2	103	Aryan	123456
3	104	Dhrub	22221

Next steps: Generate code with df | New interactive sheet

```
import pandas as pd

df = pd.read_json("myjlc.json")

df
```