Mastering Pandas: From Basics to Expert Level

Powerful Data Analysis & Manipulation with Python

pandas

Dated: 24 July 2025

Instructor : Fawad Bahadur

Github: bahadurCorvit

Training Objectives & Outcomes

1. Understand the fundamentals of Pandas Series and DataFrames
2. Perform efficient data cleaning, filtering, and transformation
3. Master advanced techniques like grouping, merging, reshaping, and time series handling
4. Apply Pandas to real-world data analysis, reporting, and machine learning workflows

Instructor : Fawad Bahadur

Introduction to Pandas

What is Pandas?

- Open-source Python library for **data manipulation and analysis.**
- Provides powerful data structures: Series and DataFrame.
- Built on top of NumPy, designed for working with structured data.

Why Use Pandas?

- Easy handling of missing data.
- Flexible reshaping and pivoting.
- Powerful group-by functionality.
- Supports time series data.

Example:

```
import pandas as pd
 data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}
df = pd.DataFrame(data)
print(df)
```

Series + Series = DataFrame

	apples
0	3
1	2
2	0
3	1

	oranges
0	0
1	3
2	7
3	2

	apples	oranges
0	3	0
1	2	3
2	0	7
3	1	2

Instructor : Fawad Bahadur

Setting Up Pandas in Your Environment

What is Pandas?

- Open-source Python library for data analysis and manipulation
- Built on top of NumPy

Prerequisites-

- Python installed (preferably 3.6 or above)
- pip (Python package installer)

Installation Using pip:

```
pip install pandas
```

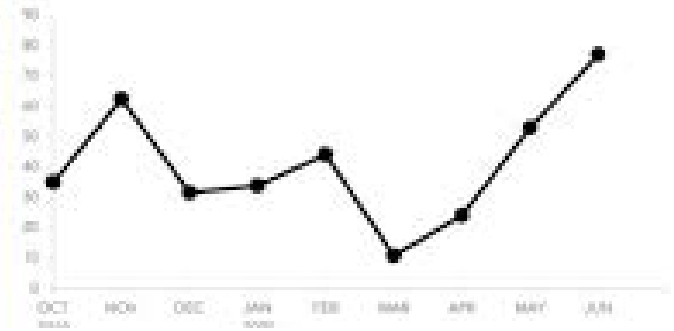
Verify Installation:

```
import pandas as pd
print(pd._version_)
```

Pandas Series

What is a Series?

- One-dimensional labeled array.
- Can hold any data type (integers, strings, floats, etc.).
- Labels (index) identify each element.



Creating a Series

```
import pandas as pd
 s = pd.Series([10, 20, 30, 40], index=['a', 'b', 'c', 'd'])
print(s)
```

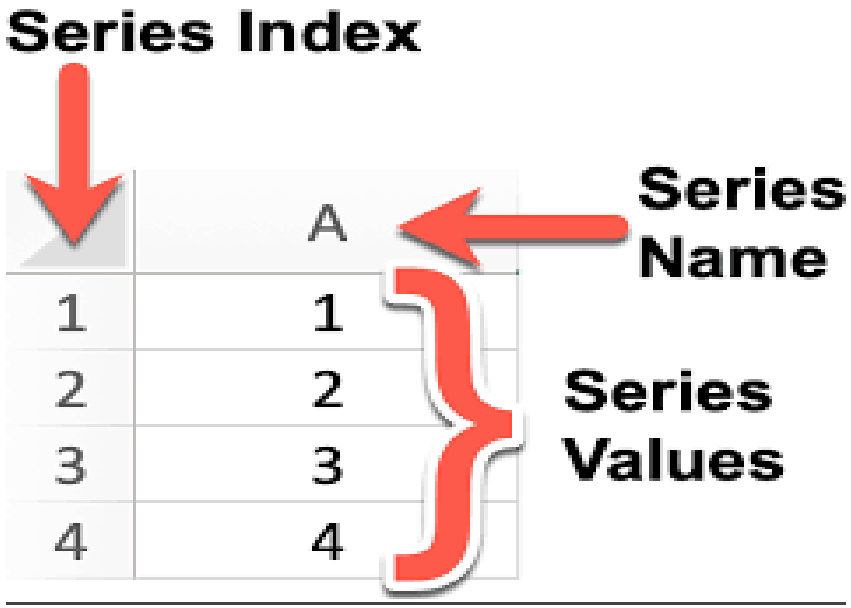
Accessing Data

```
print(s['b'])    # 20
print(s[1])      # 20 (by position)
```

Series Attributes

Values: returns the data as NumPy array

Index: returns the index labels



Tip

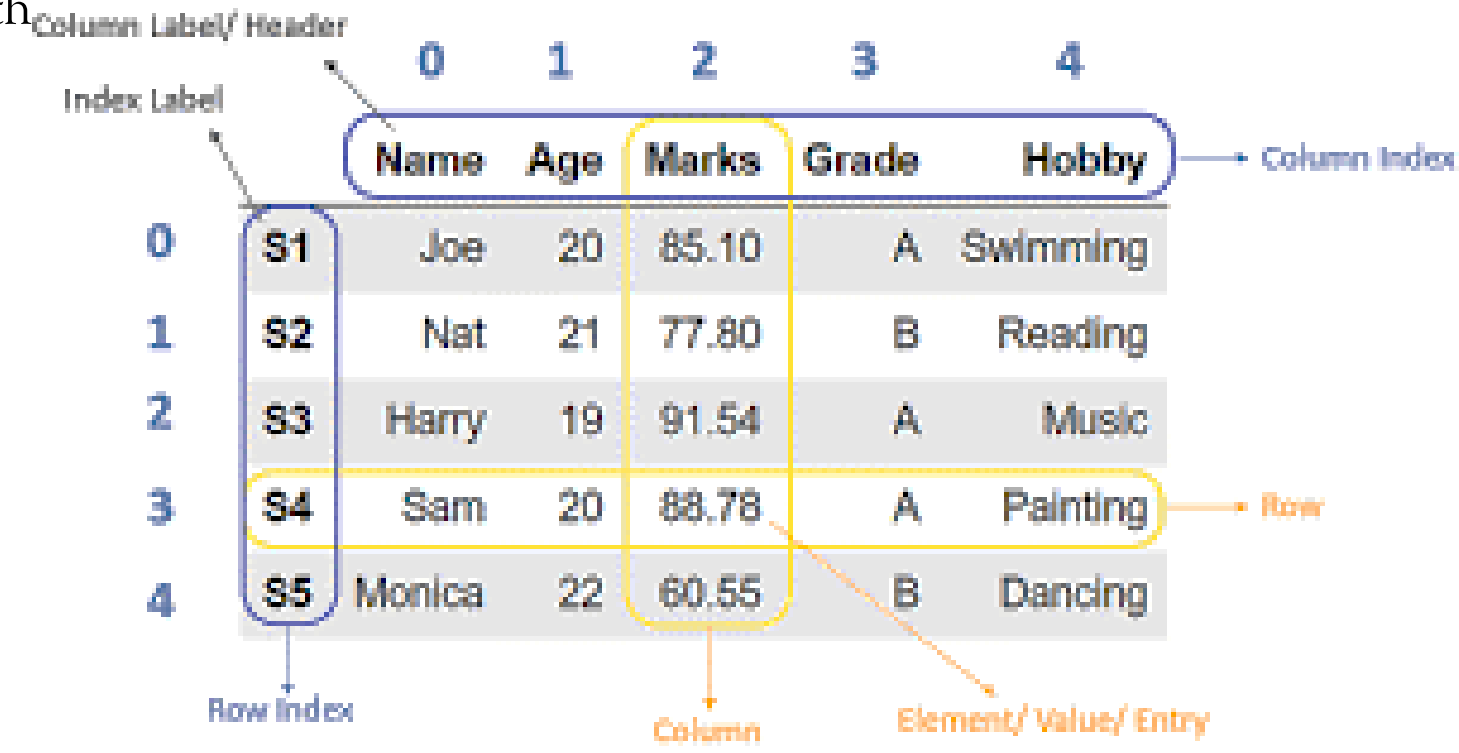
Series is the building block for DataFrame columns.

Instructor : Fawad Bahadur

Pandas DataFrame Basics

What is a DataFrame?

- Two-dimensional labeled data structure with columns of potentially different types.
- Like a spreadsheet or SQL table.



Creating a DataFrame

```
import pandas as pd
 data = {
'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35],
    'Salary': [50000, 60000, 70000]
}
df = pd.DataFrame(data)
print(df)
```

Accessing Columns

```
print(df['Name'])  # Returns Series of names
```

Accessing Rows By

index label: `df.loc[0]`

By integer position: `df.iloc[0]`

Tip

- DataFrame is the core structure for data analysis in Pandas.

Instructor : Fawad Bahadur

Data Selection & Indexing?

Selecting Columns

```
df['Age']          # Single column (Series)
df[['Name', 'Age']] # Multiple columns (DataFrame)
```

Selecting Rows

Using .loc (label-based)

```
df.loc[0]          # First row by index label
df.loc[0:2]        # Rows 0 to 2 inclusive
```

Using .iloc (integer position-based)

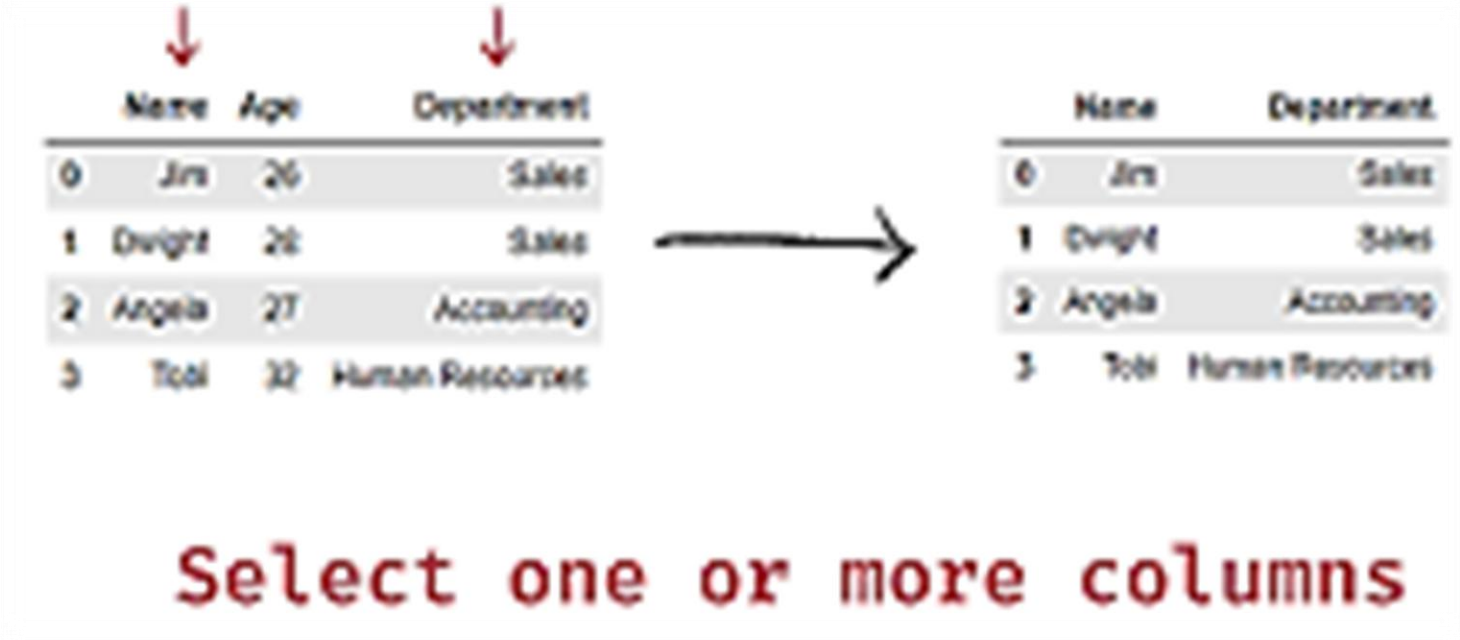
```
df.iloc[0]         # First row by position
df.iloc[0:2]       # Rows 0 and 1
```

Setting Values

```
df.loc[1, 'Salary'] = 65000
```

Aditional Selection

- `df[df['Age'] > 25]  # Rows where Age > 25`



Tip

Use .loc for label indexing and .iloc for positional indexing.

Instructor : Fawad Bahadur

THANK YOU