# 📘 Pandas

### 🔹 Introduction

* **Pandas** is an **open-source Python library** used for data analysis (evaluating big data) and data manipulation.
* Built on **NumPy** and widely used in **data science, machine learning, and analytics**.
* Provides **fast, flexible, and expressive data structures** for handling structured data.

### 🔹 Data Structures in Pandas

1. **Series**
   * A **1D labeled array** (like a column in Excel).
   * Can hold any data type: integers, floats, strings, Python objects.
   * Labels are called **index**
   * Homogenous (but can store mixed data if dtype=object). , value mutable and size immutable.
   * in series we can have sequence, list , scalar and dict
2. **DataFrame**
   * A **2D labeled data structure** (like a table in Excel or SQL).
   * Consists of rows and columns with labels.
   * Heterogenous , value mutable and size mutable.
   * Most commonly used object in Pandas.

### 🔹 Why Use Pandas?

* User-friendly and powerful for structured data.
* Replaces the need for manual Excel/SQL operations.
* Essential for **data preprocessing** in machine learning pipelines.

**what pandas can ?**

1. it can read and write data in many different data formats
2. it can easily select subsets of data from bulky data sets
3. it allows you to apply operations to independent groups
4. it supports reshaping

**# 📘 Creating a Pandas Series**

**### 1️⃣ From a \*\*Sequence\*\* (Range)**

**import pandas as pd**

**s1 = pd.Series(range(5))**

**print(s1)**

```

\*\*Output:\*\*

**0 0**

**1 1**

**2 2**

**3 3**

**4 4**

**dtype: int64**

```

**### 2️⃣ From a \*\*List\*\***

**s2 = pd.Series([10, 20, 30, 40])**

**print(s2)**

```

\*\*Output:\*\*

```

**0 10**

**1 20**

**2 30**

**3 40**

**dtype: int64**

```

---

### 3️⃣ From a \*\*NumPy Array\*\*

**import numpy as np**

**arr = np.array([100, 200, 300])**

**s3 = pd.Series(arr)**

**print(s3)**

```

\*\*Output:\*\*

```

**0 100**

**1 200**

**2 300**

**dtype: int64**

```

---

**### 4️⃣ From a \*\*Dictionary\*\***

# From a dictionary

**data = {'a': 10, 'b': 20, 'c': 30}**

**s4 = pd.Series(data)**

**print(s4)**

```

\*\*Output:\*\*

```

**a 10**

**b 20**

**c 30**

**dtype: int64**

```

---

**# 📘 Accessing Elements in a Series**

**### 1️⃣ Access by \*\*Index Position\*\* (like a list)**

**print(s2[0])**  # First element → 10

**print(s2[2])**  # Third element → 30

```

### 2️⃣ Access by \*\*Label/Index Name\*\* (when index is custom or from dict)

**print(s4['a'])**  # Access using label 'a' → 10

**print(s4['c'])**  # Access using label 'c' → 30

```

---

### 3️⃣ Access by \*\*Slicing\*\*

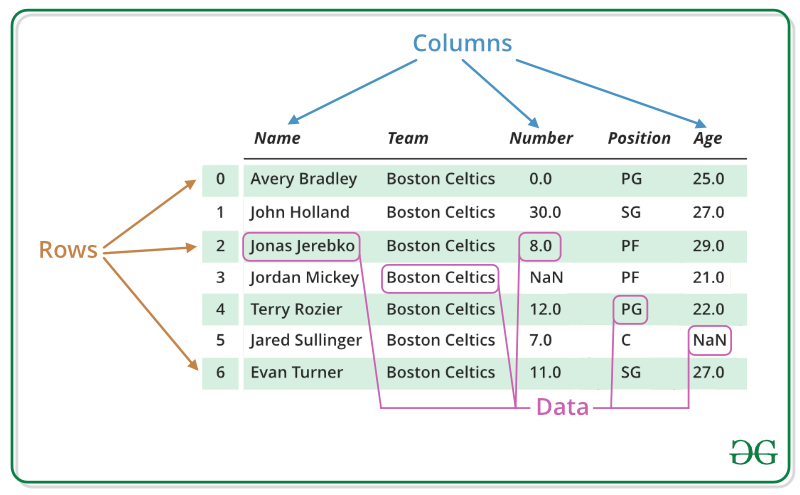
**print(s2[1:3])** # Elements from index 1 to 2 → 20, 30

**print(s4[:2])**  # First 2 elements → a:10, b:20

```

# 📘 Notes on DataFrame

## 1. What is a DataFrame?

* A **DataFrame** is a **2D labeled data structure** (rows + columns).
* A DataFrame have two indexes row index and column index
* Similar to **Excel sheets / SQL tables**.
* Each column can have a different data type.
* Built on top of **Series** objects.  
  

## 2. Creating a DataFrame

### (a) From Dictionary of Lists

Each key = column name,

Each value (list) = column data.

**import pandas as pd**

**data = {**

**"Name": ["Alice", "Bob", "Charlie"],**

**"Age": [24, 27, 22],**

**"City": ["New York", "London", "Paris"]**

**}**

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

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** |
| --- | --- | --- | --- |
| **0** | **Alice** | **24** | **New York** |
| **1** | **Bob** | **27** | **London** |
| **2** | **Charlie** | **22** | **Paris** |

### (b) From Dictionary of Dictionaries

Outer dict keys = column names.  
Inner dict keys = row labels (index),

**data = {**

**"row1": {"Name": "Alice", "Age": 24},**

**"row2": {"Name": "Bob", "Age": 27},**

**"row3": {"Name": "Charlie", "Age": 22}**

**}**

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

**print(df)**

📊 **Output:**

|  | **Name** | **Age** |
| --- | --- | --- |
| **row1** | **Alice** | **24** |
| **row2** | **Bob** | **27** |
| **row3** | **Charlie** | **22** |

### (c) From Dictionary of Series

Each Series = one column,

Index of Series = row labels.

**s1 = pd.Series([24, 27, 22], index=["a", "b", "c"])**

**s2 = pd.Series(["NY", "London", "Paris"], index=["a", "b", "c"])**

**data = {"Age": s1, "City": s2}**

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

**print(df)**

📊 **Output:**

|  | **Age** | **City** |
| --- | --- | --- |
| **a** | **24** | **NY** |
| **b** | **27** | **London** |
| **c** | **22** | **Paris** |

### (d) From Another DataFrame

**data = {"Name": ["Alice", "Bob", "Charlie"], "Age": [24, 27, 22]}**

**df1 = pd.DataFrame(data)**

**df2 = pd.DataFrame(df1, columns=["Name"])**

**print(df2)**

📊 **Output:**

|  | **Name** |
| --- | --- |
| **0** | **Alice** |
| **1** | **Bob** |
| **2** | **Charlie** |

# 📊 Viewing Data in Pandas

### Head & Tail

## df.head(n) → First *n* rows (default 5).

## df.tail(n) → Last *n* rows (default 5).

## 

## 3. Accessing Data in DataFrame

### (a) Access Columns

**print(df["Name"])**

**print(df[["Name", "Age"]])**

📊 **Output (single column):**

**0 Alice**

**1 Bob**

**2 Charlie**

**Name: Name, dtype: object**

📊 **Output (multiple columns):**

|  | **Name** | **Age** |
| --- | --- | --- |
| **0** | **Alice** | **24** |
| **1** | **Bob** | **27** |
| **2** | **Charlie** | **22** |

### (b) Access Rows

**print(df.loc[0]) # by label/index**

**print(df.iloc[1]) # by position**

📊 **Output (df.loc[0]):**

**Name Alice**

**Age 24**

**City New York**

**Name: 0, dtype: object**

📊 **Output (df.iloc[1]):**

**Name Bob**

**Age 27**

**City London**

**Name: 1, dtype: object**

### (c) Access Specific Cell

***print(df.loc[0, "Name"])***

***print(df.iloc[1, 0])***

📊 **Output:**

**Alice**

**Bob**

## d) Setting Custom Index at Creation

**import pandas as pd**

**data = {"Name": ["Alice", "Bob", "Charlie"], "Age": [24, 27, 22]}**

**df = pd.DataFrame(data, index=["a", "b", "c"])**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** |
| --- | --- | --- |
| **a** | **Alice** | **24** |
| **b** | **Bob** | **27** |
| **c** | **Charlie** | **22** |

## e) Setting Index after Creation

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

**df.set\_index("Name", inplace=True)**

**print(df)**

📊 **Output:**

| **Name** | **Age** |
| --- | --- |
| **Alice** | **24** |
| **Bob** | **27** |
| **Charlie** | **22** |

## f) Resetting Index

**df=\_reset = df.reset\_index()**

**print(df\_reset)**

# 📘 Adding, Modifying & Deleting Columns and Rows in Pandas

## 1. Adding Columns

### (a) Add New Column with a List/Series

**import pandas as pd**

**data = {"Name": ["Alice", "Bob", "Charlie"], "Age": [24, 27, 22]}**

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

**df["City"] = ["New York", "London", "Paris"] # new column**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** |
| --- | --- | --- | --- |
| **0** | **Alice** | **24** | **New York** |
| **1** | **Bob** | **27** | **London** |
| **2** | **Charlie** | **22** | **Paris** |

### (b) Add Column with Same Value

**df["Country"] = "USA"**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** | **Country** |
| --- | --- | --- | --- | --- |
| **0** | **Alice** | **24** | **New York** | **USA** |
| **1** | **Bob** | **27** | **London** | **USA** |
| **2** | **Charlie** | **22** | **Paris** | **USA** |

### (c) Add Column using Existing Columns

**df["Age+5"] = df["Age"] + 5**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** | **Country** | **Age+5** |
| --- | --- | --- | --- | --- | --- |
| **0** | **Alice** | **24** | **New York** | **USA** | **29** |
| **1** | **Bob** | **27** | **London** | **USA** | **32** |
| **2** | **Charlie** | **22** | **Paris** | **USA** | **27** |

## 2. Modifying Columns

### (a) Update Column Values

**df["Age"] = df["Age"] + 1 # increase all ages by 1**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** |
| --- | --- | --- | --- |
| **0** | **Alice** | **25** | **New York** |
| **1** | **Bob** | **28** | **London** |
| **2** | **Charlie** | **23** | **Paris** |

### (b) Modify Specific Cell

**df.loc[1, "City"] = "Berlin"**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** |
| --- | --- | --- | --- |
| **0** | **Alice** | **25** | **New York** |
| **1** | **Bob** | **28** | **Berlin** |
| **2** | **Charlie** | **23** | **Paris** |

## 3. Deleting Columns

### (a) Drop Single Column

**df.drop("City", axis=1, inplace=True)**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** |
| --- | --- | --- |
| **0** | **Alice** | **25** |
| **1** | **Bob** | **28** |
| **2** | **Charlie** | **23** |

### (b) Drop Multiple Columns

**df.drop(["Age", "Country"], axis=1, inplace=True)**

**print(df)**

📊 **Output:**

|  | **Name** |
| --- | --- |
| **0** | **Alice** |
| **1** | **Bob** |
| **2** | **Charlie** |

## 4. Adding Rows

### (a) Append Dictionary (deprecated, use concat)

**new\_row = {"Name": "David", "Age": 30, "City": "Tokyo"}**

**df2 = pd.DataFrame([new\_row])**

**df = pd.concat([df, df2], ignore\_index=True)**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** |
| --- | --- | --- | --- |
| **0** | **Alice** | **25** | **New York** |
| **1** | **Bob** | **28** | **London** |
| **2** | **Charlie** | **23** | **Paris** |
| **3** | **David** | **30** | **Tokyo** |

## 5. Modifying Rows

### (a) Update Row using .loc

**df.loc[2] = ["Charlie", 24, "Paris-France"]**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** |
| --- | --- | --- | --- |
| **0** | **Alice** | **25** | **New York** |
| **1** | **Bob** | **28** | **London** |
| **2** | **Charlie** | **24** | **Paris-France** |
| **3** | **David** | **30** | **Tokyo** |

## 6. Deleting Rows

### (a) Drop by Index

**df.drop(1, inplace=True)**

**print(df)**

📊 **Output:**

|  | **Name** | **Age** | **City** |
| --- | --- | --- | --- |
| **0** | **Alice** | **25** | **New York** |
| **2** | **Charlie** | **24** | **Paris-France** |
| **3** | **David** | **30** | **Tokyo** |