

Pandas: Complete Guide

Pandas is a powerful Python library for data analysis and manipulation. It provides easy-to-use structures like **Series** and **DataFrame**, making data handling efficient.

Installation

If you haven't installed Pandas yet, you can do so using:

```
pip install pandas
```

1. Importing Pandas

```
import pandas as pd
```

2. Data Structures in Pandas

Pandas has two primary data structures:

1. **Series** – One-dimensional labeled array
2. **DataFrame** – Two-dimensional table

Creating a Series

```
data = [10, 20, 30, 40, 50]
```

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

```
print(series)
```

Output:

0 10

1 20

2 30

3 40

4 50

dtype: int64

Creating a DataFrame

```
data = {
```

```
    'Name': ['Alice', 'Bob', 'Charlie'],
```

```
    'Age': [25, 30, 35],
```

```
    'Salary': [50000, 60000, 70000]
```

```
}
```

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

```
print(df)
```

Output:

```
   Name  Age  Salary
```

```
0  Alice   25  50000
```

```
1   Bob    30  60000
```

```
2 Charlie   35  70000
```

3. Reading and Writing Data

Reading a CSV File

```
df = pd.read_csv('data.csv')  
  
print(df.head()) # Display first 5 rows
```

Writing to a CSV File

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

Reading an Excel File

```
df = pd.read_excel('data.xlsx', sheet_name='Sheet1')
```

Writing to an Excel File

```
df.to_excel('output.xlsx', index=False)
```

4. Data Exploration

Basic Information

```
print(df.info()) # Summary of dataset  
  
print(df.describe()) # Statistical summary  
  
print(df.shape) # Number of rows and columns  
  
print(df.columns) # List of column names
```

Checking Missing Values

```
print(df.isnull().sum()) # Count missing values per column
```

5. Selecting and Filtering Data

Selecting Columns

```
print(df['Age']) # Select a single column
```

```
print(df[['Name', 'Salary']]) # Select multiple columns
```

Selecting Rows

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

```
print(df.loc[1]) # Select row with index 1
```

Filtering Data

```
print(df[df['Age'] > 30]) # Filter rows where Age > 30
```

```
print(df[(df['Age'] > 25) & (df['Salary'] > 55000)]) # Multiple conditions
```

6. Modifying Data

Adding a New Column

```
df['Experience'] = [2, 5, 7]
```

```
print(df)
```

Updating Values

```
df.loc[df['Name'] == 'Alice', 'Salary'] = 55000
```

```
print(df)
```

Deleting Columns

```
df.drop('Experience', axis=1, inplace=True) # axis=1 means column
```

```
print(df)
```

Deleting Rows

```
df.drop(0, axis=0, inplace=True) # axis=0 means row
```

```
print(df)
```

7. Handling Missing Data

Filling Missing Values

```
df.fillna(0, inplace=True) # Replace NaN with 0
```

Dropping Missing Values

```
df.dropna(inplace=True) # Remove rows with NaN values
```

8. Sorting and Aggregating Data

Sorting Data

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

Grouping and Aggregating Data

```
df.groupby('Age')['Salary'].mean()
```

9. Merging and Joining DataFrames

Concatenating DataFrames

```
df1 = pd.DataFrame({'A': [1, 2], 'B': [3, 4]})
```

```
df2 = pd.DataFrame({'A': [5, 6], 'B': [7, 8]})
```

```
result = pd.concat([df1, df2])
```

```
print(result)
```

Merging DataFrames

```
df1 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Alice', 'Bob', 'Charlie']})
```

```
df2 = pd.DataFrame({'ID': [1, 2, 4], 'Salary': [50000, 60000, 70000]})
```

```
merged_df = pd.merge(df1, df2, on='ID', how='inner') # inner, left, right, outer
```

```
print(merged_df)
```

10. Working with Date and Time

```
df['Date'] = pd.to_datetime(df['Date'])
```

```
df['Year'] = df['Date'].dt.year
```

```
df['Month'] = df['Date'].dt.month
```

11. Pivot Tables

```
df.pivot_table(index='Age', values='Salary', aggfunc='mean')
```

12. Visualizing Data with Pandas and Matplotlib

```
import matplotlib.pyplot as plt
```

```
df['Age'].plot(kind='bar') # Bar chart
```

```
plt.show()
```

```
df.plot(x='Age', y='Salary', kind='line') # Line plot
```

```
plt.show()
```

```
df['Salary'].plot(kind='hist', bins=5) # Histogram
```

```
plt.show()
```

13. Exporting Data

To CSV

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

To Excel

```
df.to_excel('output.xlsx', index=False)
```

To JSON

```
df.to_json('output.json')
```

Summary

Feature	Description
Read CSV	<code>pd.read_csv('file.csv')</code>
Read Excel	<code>pd.read_excel('file.xlsx')</code>
Basic Info	<code>df.info(), df.describe()</code>
Select Column	<code>df['column']</code>
Select Row	<code>df.iloc[0], df.loc[1]</code>
Filter Data	<code>df[df['Age'] > 30]</code>
Add Column	<code>df['NewCol'] = values</code>
Drop Column	<code>df.drop('Col', axis=1)</code>
Handle Missing	<code>df.fillna(0), df.dropna()</code>
Sort Data	<code>df.sort_values(by='Age')</code>
Group Data	<code>df.groupby('Age')['Salary'].mean()</code>
Merge Data	<code>pd.merge(df1, df2, on='ID')</code>

Pivot Table `df.pivot_table(index='Age',
values='Salary')`

Visualization `df.plot(kind='bar')`
