# Python Pandas

## What is Pandas?

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

### Installing Pandas

First, you need to install Pandas. You can do this using pip:

```
pip install pandas
```

### Importing Pandas

To use Pandas in your script, you need to import it:

```
import pandas as pd
```

### Data Structures in Pandas

**Series:-** A Series is a one-dimensional array-like object that can hold any data type.

```
import pandas as pd

# Creating a Series from a list
data = [1, 2, 3, 4, 5]
s = pd.Series(data)
print(s)

# Creating a Series with custom index
data = [1, 2, 3, 4, 5]
index = ['a', 'b', 'c', 'd', 'e']
s = pd.Series(data, index=index)
print(s)
```

## DataFrame

A DataFrame is a two-dimensional, size-mutable, and potentially heterogeneous tabular data structure with labeled axes (rows and columns).

```
# Creating a DataFrame from a dictionary
data = {
 'name': ['Alice', 'Bob', 'Charlie'],
 'age': [25, 30, 35],
 'city': ['New York', 'Los Angeles',
'Chicago']
}
df = pd.DataFrame(data)
print(df)
```

## Reading and Writing Data

### Reading Data

Pandas can read data from various file formats, including CSV, Excel, and SQL.

### Reading a CSV file

```
# Reading a CSV file
df = pd.read_csv('data.csv')
print(df)
```

### Reading an Excel file

```
# Reading an Excel file
df = pd.read_excel('data.xlsx')
print(df)
```

### Writing Data

Pandas can also write data to various file formats.

### Writing to a CSV file

```
# Writing to a CSV file
df.to_csv('output.csv', index=False)
```

### Writing to an Excel file

```
# Writing to an Excel file
df.to_excel('output.xlsx', index=False)
```

## DataFrame Operations

### Viewing Data

```
# Display the first few rows of the
DataFrame
print(df.head())

# Display the last few rows of the
DataFrame
print(df.tail())

# Display the DataFrame's information
print(df.info())

# Display the summary statistics of the
DataFrame
```

```
print(df.describe())
```

## Selecting Data

### Selecting Columns

```
# Selecting a single column
print(df['name'])

# Selecting multiple columns
print(df[['name', 'age']])
```

### Selecting Rows

```
# Selecting a single row by index
print(df.iloc[0])

# Selecting multiple rows by index
print(df.iloc[0:2])

# Selecting rows based on a condition
print(df[df['age'] > 30])
```

## Adding and Dropping Data

```
# Adding a new column
df['salary'] = [50000, 60000, 70000]
```

```
print(df)

# Dropping a column
df = df.drop('salary', axis=1)
print(df)

# Dropping a row
df = df.drop(0, axis=0)
print(df)
```

### Modifying Data

```
# Modifying a column
df['age'] = df['age'] + 1
print(df)

# Modifying a row
df.loc[1] = ['Bob', 31, 'San Francisco']
print(df)
```

### Handling Missing Data

```
# Creating a DataFrame with missing values
data = {
 'name': ['Alice', 'Bob', 'Charlie'],
 'age': [25, None, 35],
 'city': ['New York', None, 'Chicago']
}
df = pd.DataFrame(data)
print(df)

# Checking for missing values
print(df.isnull())

# Dropping missing values
df = df.dropna()
print(df)

# Filling missing values
df = df.fillna({'age': 30, 'city':
'Unknown'})
print(df)
```

## Grouping and Aggregating Data

```
# Creating a DataFrame
data = {
 'name': ['Alice', 'Bob', 'Charlie',
'Alice', 'Bob'],
 'age': [25, 30, 35, 25, 30],
 'city': ['New York', 'Los Angeles',
'Chicago', 'New York', 'Los Angeles'],
 'salary': [50000, 60000, 70000, 55000,
65000]
}
df = pd.DataFrame(data)

# Grouping data by a column and calculating
the mean
grouped = df.groupby('name').mean()
print(grouped)

# Grouping data by multiple columns and
calculating the sum
grouped = df.groupby(['name', 'city']).sum()
print(grouped)
```

## Merging and Joining DataFrames

```
# Creating two DataFrames
left = pd.DataFrame({
 'key': ['A', 'B', 'C'],
 'value_left': [1, 2, 3]
})
```

```
right = pd.DataFrame({
 'key': ['A', 'B', 'D'],
 'value_right': [4, 5, 6]
})

# Merging DataFrames on a key column
merged = pd.merge(left, right, on='key',
how='inner')
print(merged)
```

## Applying Functions

### Using `apply`

```
# Applying a function to each column
df['salary'] = df['salary'].apply(lambda x:
x * 1.1)
print(df)

# Applying a function to each row
df['combined'] = df.apply(lambda row:
f"{row['name']} ({row['city']})", axis=1)
print(df)
```

### Using `applymap` for Element-wise Operations

```
# Applying a function to each element
df = df.applymap(lambda x: str(x).upper())
print(df)
```

## Conclusion

Pandas is an essential library for data analysis in Python, providing powerful tools to manipulate, analyze, and visualize data.