

# Python Coding Challenge

**Q.1 Explain Pandas for Data Processing & execute Reading CSV Data using Pandas & Read Data from CSV Files to Pandas Dataframes & Filter Data in Pandas Dataframe using query.**

**Ans.**

**Pandas for Data Processing:**

**Pandas Overview:**

Pandas is an open-source data manipulation and analysis library for Python. It provides data structures like Series and DataFrame, which are powerful tools for working with structured data. Pandas is widely used in data preprocessing, cleaning, exploration, and analysis.

**Key Components:**

- **DataFrame:** A two-dimensional table with labeled axes (rows and columns).
- **Series:** A one-dimensional labeled array capable of holding any data type.
- **Index:** Labels for rows and columns in DataFrames.

**Reading CSV Data using Pandas:**

**Reading CSV Files:**

Pandas provides the `read_csv()` function to read data from CSV files into a DataFrame.

```
import pandas as pd

# Read CSV data into a DataFrame
df = pd.read_csv('your_file.csv')
```

**Read Data from CSV Files to Pandas Dataframes:**

Creating DataFrame from CSV:

The `read_csv()` function reads CSV data and creates a DataFrame, allowing us to perform various operations on the data.

The screenshot displays a Jupyter Notebook interface with the following content:

```
In [1]: import pandas as pd
from datetime import datetime, timedelta

# Sample DataFrame creation
data = {
    'ID': [1, 2, 3, 4, 5],
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
    'Age': [25, 30, 22, 35, 28],
    'Status': ['Active', 'Inactive', 'Active', 'Active', 'Inactive'],
    'Date': [datetime(2022, 1, 1) + timedelta(days=i) for i in range(5)]
}

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

Out[1]:

	ID	Name	Age	Status	Date
0	1	Alice	25	Active	2022-01-01
1	2	Bob	30	Inactive	2022-01-02
2	3	Charlie	22	Active	2022-01-03
3	4	David	35	Active	2022-01-04
4	5	Eva	28	Inactive	2022-01-05

```
In [2]: # Save DataFrame to CSV file
df.to_csv('python_coding_challenge_data.csv', index=False)

In [4]: # Read CSV data into a DataFrame
new_df = pd.read_csv('python_coding_challenge_data.csv')
new_df.head()
```

Out[4]:

	ID	Name	Age	Status	Date
0	1	Alice	25	Active	2022-01-01
1	2	Bob	30	Inactive	2022-01-02
2	3	Charlie	22	Active	2022-01-03
3	4	David	35	Active	2022-01-04
4	5	Eva	28	Inactive	2022-01-05

```
In [5]: # Filter Data using query
Filtered_df = new_df.query('Age > 25')

# Display the filtered DataFrame
print("\nFiltered DataFrame:")
print(Filtered_df)
```

Filtered DataFrame:

	ID	Name	Age	Status	Date
1	2	Bob	30	Inactive	2022-01-02
3	4	David	35	Active	2022-01-04
4	5	Eva	28	Inactive	2022-01-05

```
In [6]: filtered_df = df.query('Age > 25 and Status == "Active"')
print(filtered_df)
```

Filtered DataFrame:

	ID	Name	Age	Status	Date
3	4	David	35	Active	2022-01-04

## Filter Data in Pandas DataFrame using query:

Filtering Data with query:

Pandas provides the query() method to filter data based on a query expression.

The screenshot displays a Jupyter Notebook titled "Python-Coding-Challenge" running on a local host. The notebook contains two code cells. The first cell, labeled "In [5]:", filters a DataFrame based on the 'Age' attribute being greater than 25. It uses the following code:

```
# Filter Data using query
Filtered_df = new_df.query('Age > 25')

# Display the filtered DataFrame
print("\nFiltered DataFrame:")
print(Filtered_df)
```

The output of this cell shows a filtered DataFrame with the following data:

ID	Name	Age	Status	Date
1	Alice	25	Active	2022-01-01
2	Bob	30	Inactive	2022-01-02
3	Charlie	22	Active	2022-01-03
4	David	35	Active	2022-01-04
5	Eva	28	Inactive	2022-01-05

The second cell, labeled "In [6]:", further filters the DataFrame by adding a condition for 'Status' being "Active". It uses the following code:

```
Filtered_df = df.query('Age > 25 and Status == "Active"')
print(filtered_df)
```

The output of this cell shows the final filtered DataFrame:

ID	Name	Age	Status	Date
3	David	35	Active	2022-01-04

In the above example, we filtered the dataframe based on attribute Age i.e. where age is greater than 30 and in other query we added an addition query i.e to include age over 30 and status as active. This filters the DataFrame to include only the rows that meet the specified criteria.