

## Q1. Explain Pandas for Data Processing

- Pandas is a Python library for data manipulation and analysis.
- It provides easy-to-use data structures like DataFrame and Series, along with tools for reading and writing data in various formats.
- Think of it as a table or spreadsheet. It's a 2D structure to hold data, where each column can be of a different data type.

## Q2. Execute Reading CSV Data using Pandas

```
In [3]: import pandas as pd
# Reading the csv file by using read_csv and specifying the file path
temp = pd.read_csv('sales_data.csv')
print(temp)
```

	Date	Status	Amount	Month
0	2022-01-01	Active	100	January
1	2022-01-01	Inactive	150	January
2	2022-02-01	Active	80	February
3	2022-02-01	Active	120	February
4	2022-02-01	Inactive	90	February

we can see different details of the csv file by using different functions like tail, head, describe, etc.

```
In [8]: print(temp.head(2))
print()
print(temp.tail(2))
print()
print(temp.describe())
```

	Date	Status	Amount	Month
0	2022-01-01	Active	100	January
1	2022-01-01	Inactive	150	January

	Date	Status	Amount	Month
3	2022-02-01	Active	120	February
4	2022-02-01	Inactive	90	February

	Amount
count	5.000000
mean	108.000000
std	27.748874
min	80.000000
25%	90.000000
50%	100.000000
75%	120.000000
max	150.000000

### Read Data from CSV Files to Pandas Dataframes

- we can read csv file and create a pandas dataframe using pd.DataFrame() functions which returns a dataframe.

```
In [9]: import pandas as pd
# using pd.DataFrame to initialize the dataframe from the csv file
df1 = pd.DataFrame(temp)
print(df1)
```

	Date	Status	Amount	Month
0	2022-01-01	Active	100	January
1	2022-01-01	Inactive	150	January
2	2022-02-01	Active	80	February
3	2022-02-01	Active	120	February
4	2022-02-01	Inactive	90	February

### Filter Data in Pandas Dataframe using query.

- Now we will filter the dataframe by passing different conditions to the dataframe and get the desired output by using .query() method.

```
In [12]: # Getting all those records where Amount is equal to 100
print(df1.query('Amount==100'))
```

	Date	Status	Amount	Month
0	2022-01-01	Active	100	January

```
In [13]: # Getting all those records where Status is equal to "Active"
print(df1.query('Status=="Active"'))
```

	Date	Status	Amount	Month
0	2022-01-01	Active	100	January
2	2022-02-01	Active	80	February
3	2022-02-01	Active	120	February

```
In [14]: # Getting all those records where Month is equal "February"
print(df1.query('Month=="February"'))
```

	Date	Status	Amount	Month
2	2022-02-01	Active	80	February
3	2022-02-01	Active	120	February
4	2022-02-01	Inactive	90	February

```
In [21]: # Getting query results with a variable
threshold = 100
print(df1.query('Amount > @threshold'))
```

	Date	Status	Amount	Month
1	2022-01-01	Inactive	150	January
3	2022-02-01	Active	120	February

### Ques. Execute with one example Lambda Functions in Python.

- Lambda functions are those functions which don't have a name, and they are mainly one-liner functions. They are not for any complicated functionality but rather simple tasks which are performed repetitively

```
In [16]: is_even = lambda x: x%2==0
is_odd = lambda x: x%2==1
get_sum = lambda x,y: x+y
get_abs_diff = lambda x,y: abs(x-y)

print(is_even(2))
print(is_odd(10))
print(is_even(11))
print(get_sum(10, 11))
print(get_abs_diff(1,10))
```

```
True
False
False
21
9
```

### Read JSON Strings to Python dicts or lists

- we can read JSON String to Python dicts or list based on our requirements or according to the data.
- we will json module in python to read JSON String and convert them to python list or dicts.
- we will use .loads to converting json string to python dict or list.

```
In [18]: # importing the json module
import json
# this is our example json string
temp_str = '{"name": "John", "age": 25, "city": "New York"}'
obj1 = json.loads(temp_str)
print(obj1)
print(obj1['name'])
```

```
{'name': 'John', 'age': 25, 'city': 'New York'}
John
```

```
In [20]: # This is our example python json list string
json_array_string = '[{"name": "John", "age": 25}, {"name": "Alice", "age": 30}]'
python_list = json.loads(json_array_string)
print(python_list)
```

```
print(python_list[0])
print(python_list[1]['name'])
```

```
[{'name': 'John', 'age': 25}, {'name': 'Alice', 'age': 30}]
{'name': 'John', 'age': 25}
Alice
```