Q1. Explain Pandas for Data Processing

Pandas is a Python library used for working with data sets.
It has functions for analyzing, cleaning, exploring, and manipulating data.
Pandas is a powerful open-source data manipulation and analysis library for Python. It provides easy-to-use data structures, such as DataFrame and Series, along with a variety of functions to manipulate and analyze structured data.
Pandas is useful for various uses:

1. Data Manipulation : Various operations can be performed on Pandas DataFrame such as filtering,etc.
2. Data Cleaning and preProcessing : Pandas offers numerous functions for cleaning and preprocessing data, such as handling missing values, removing duplicates, and transforming data types.

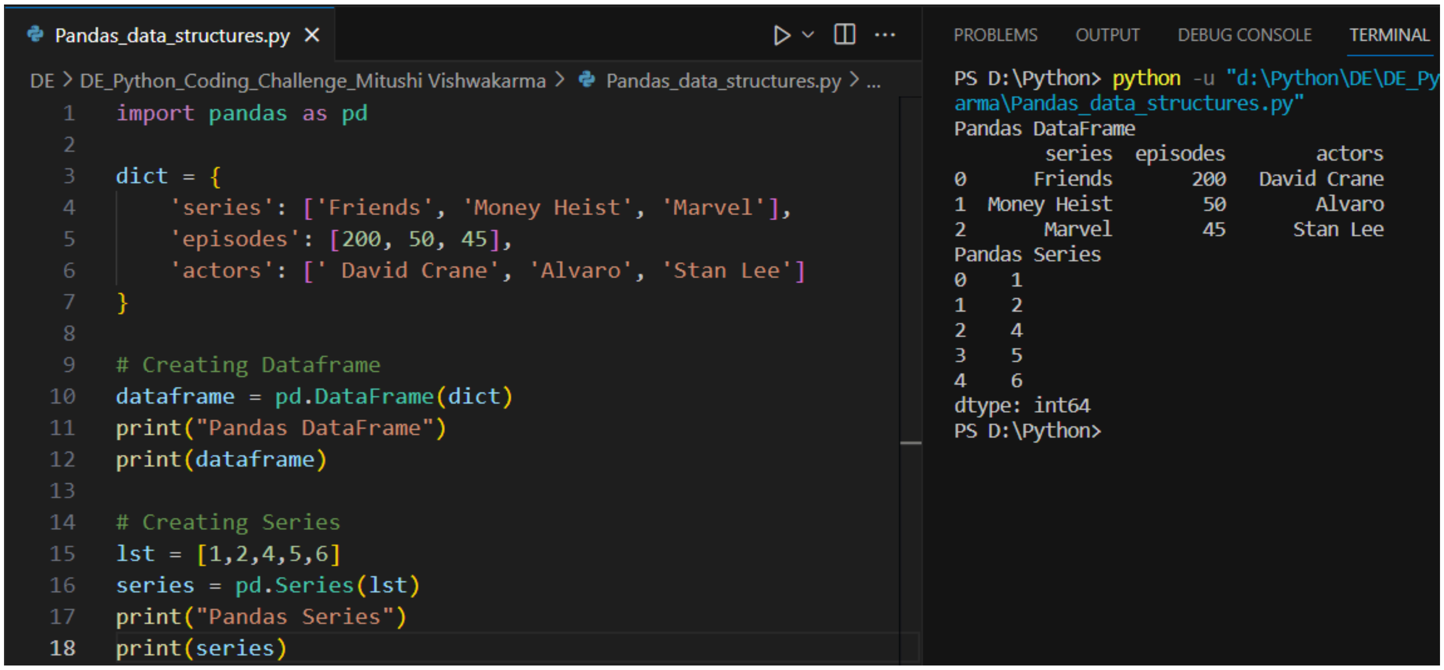
Pandas Data Structures are explained below :

Series :

- A Series is a one-dimensional labeled array that can hold any data type. It is similar to a column in a DataFrame.
- Each column in a DataFrame is essentially a Series.

DataFrame :

- The DataFrame is the primary data structure in Pandas. It is a two-dimensional, labeled data structure with columns that can be of different types (integers, floats, strings, etc.).
- DataFrame can be considered similar to excel sheet with rows and columns and their indexing.


Pandas_data_structures.py
DE > DE_Python_Coding_Challenge_Mitushi Vishwakarma > Pandas_data_structures.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL

```python
import pandas as pd

dict = {
    'series': ['Friends', 'Money Heist', 'Marvel'],
    'episodes': [200, 50, 45],
    'actors': [' David Crane', 'Alvaro', 'Stan Lee']
}

# Creating Dataframe
dataframe = pd.DataFrame(dict)
print("Pandas DataFrame")
print(dataframe)

# Creating Series
lst = [1,2,4,5,6]
series = pd.Series(lst)
print("Pandas Series")
print(series)
```

```
PS D:\Python> python -u "d:\Python\DE\DE_Py
arma\Pandas_data_structures.py"
Pandas DataFrame
        series  episodes        actors
0      Friends       200   David Crane
1  Money Heist        50        Alvaro
2       Marvel        45      Stan Lee
Pandas Series
0    1
1    2
2    4
3    5
4    6
dtype: int64
PS D:\Python>
```

Execute Reading CSV Data using Pandas

Pandas can be used to read data from CSV files. It has a read_csv() method which takes file path as an argument and converts the csv data to pandas dataframe.

Using csv_read() :

1. Import Pandas library
2. Create csv file path
3. Use the pandas.read_csv() method and pass the file path in it.
4. Print the variable used to store the data.

There are various methods to read csv data using pandas :

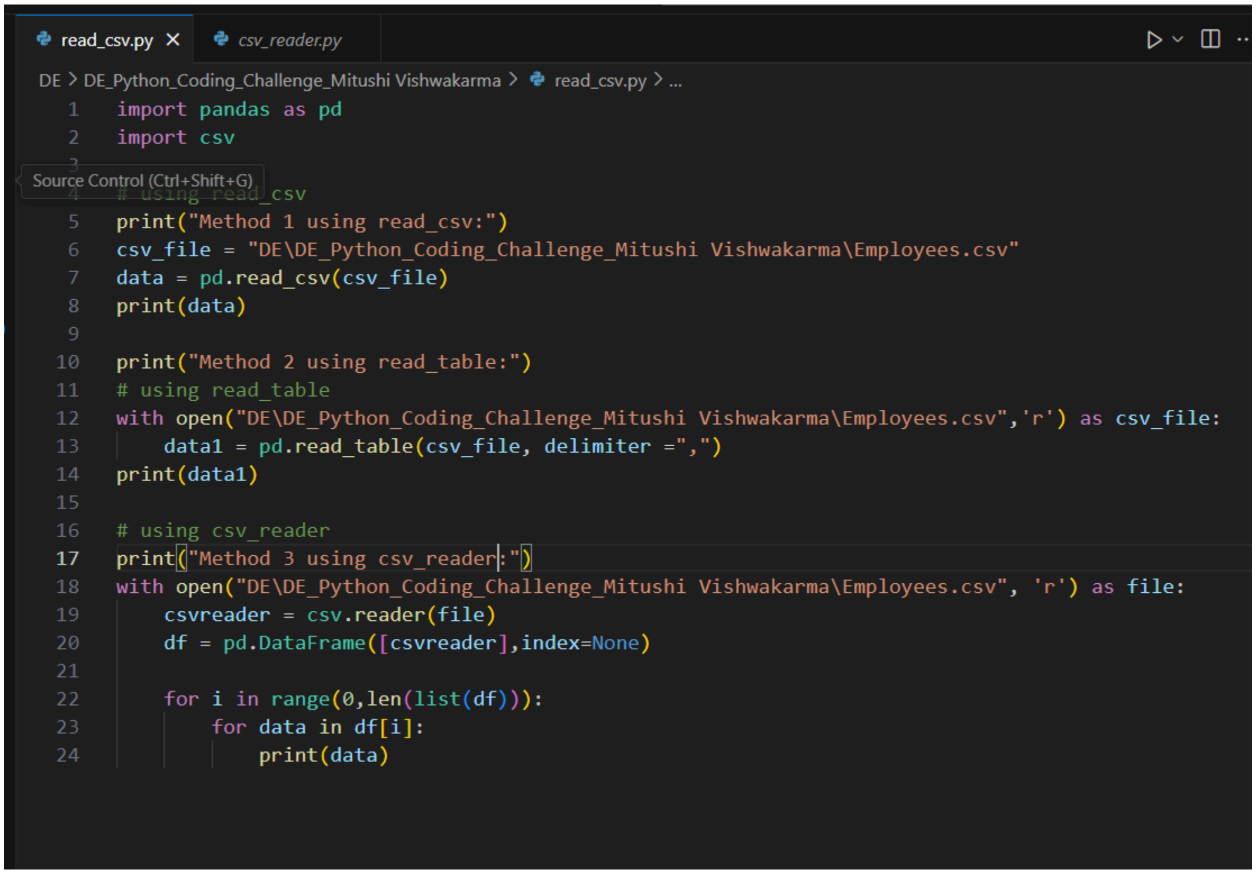
Using read_table() :

Here we use read_table(file_path,delimiter=",") # default

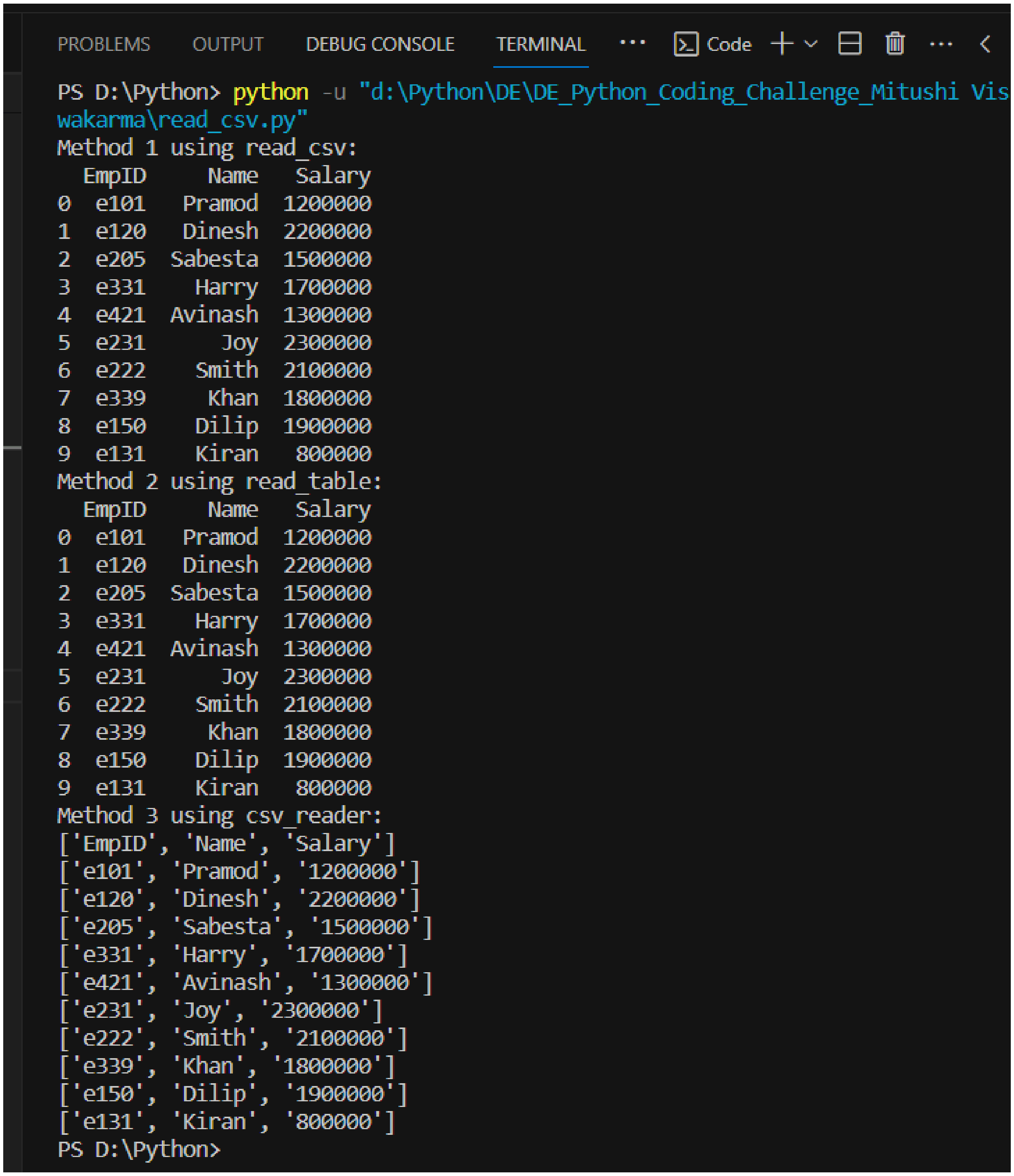
and

Using csv.reader() :

Here we import csv module and create a csv.reader(filepath) object then we convert it to pandas.DataFrame.


read_csv.py
csv_reader.py
DE > DE_Python_Coding_Challenge_Mitushi Vishwakarma > read_csv.py > ...
Source Control (Ctrl+Shift+G)

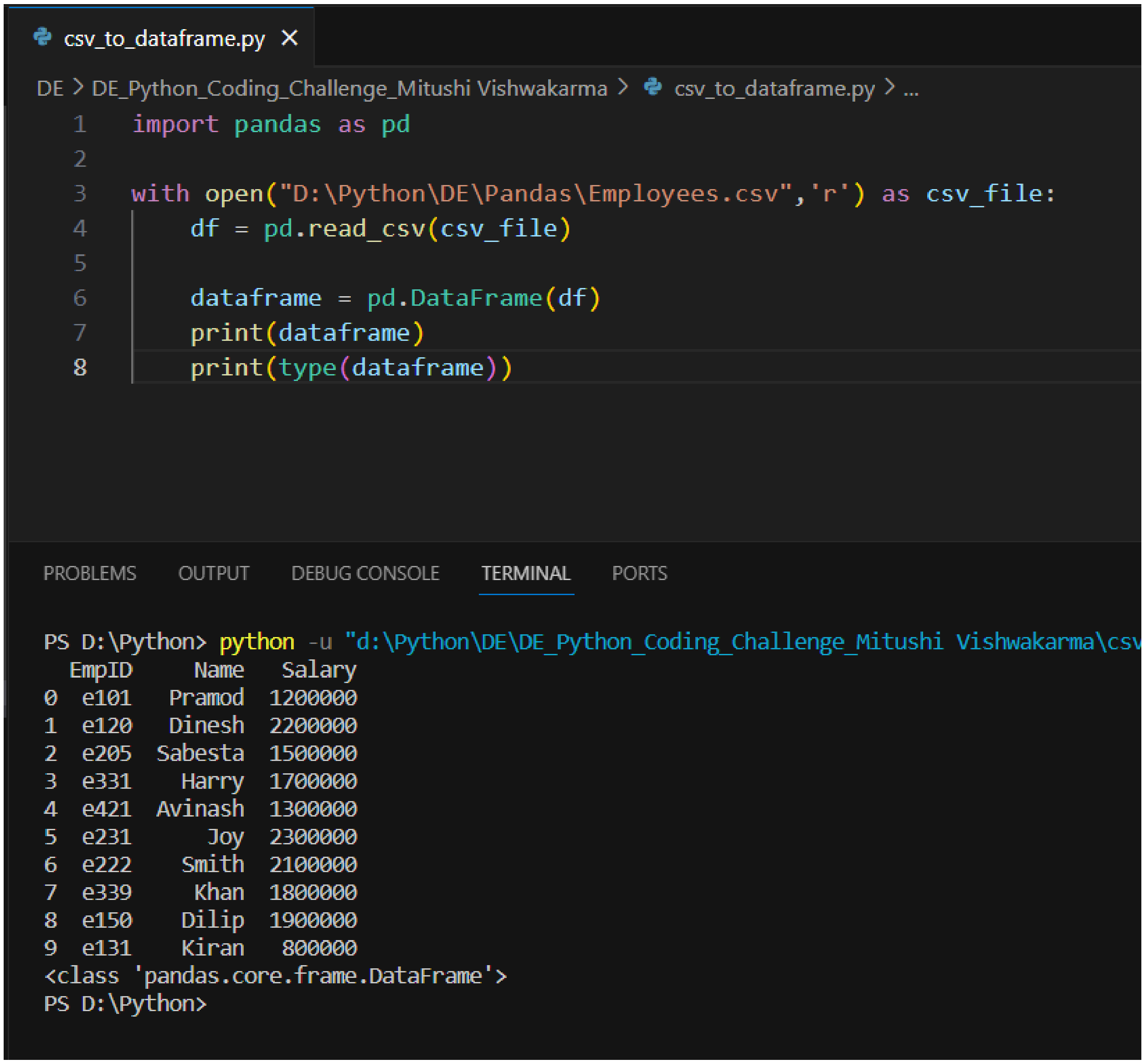
```python
import pandas as pd
import csv

# using read_csv
print("Method 1 using read_csv:")
csv_file = "DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\Employees.csv"
data = pd.read_csv(csv_file)
print(data)

print("Method 2 using read_table:")
# using read_table
with open("DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\Employees.csv",'r') as csv_file:
    data1 = pd.read_table(csv_file, delimiter =",")
print(data1)

# using csv_reader
print("Method 3 using csv_reader:")
with open("DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\Employees.csv", 'r') as file:
    csvreader = csv.reader(file)
    df = pd.DataFrame([csvreader],index=None)

    for i in range(0,len(list(df))):
        for data in df[i]:
            print(data)
```



```
PS D:\Python> python -u "d:\Python\DE\DE_Python_Coding_Challenge_Mitushi Vis
wakarma\read_csv.py"
Method 1 using read_csv:
  EmpID     Name   Salary
0  e101   Pramod  1200000
1  e120   Dinesh  2200000
2  e205  Sabesta  1500000
3  e331    Harry  1700000
4  e421  Avinash  1300000
5  e231      Joy  2300000
6  e222    Smith  2100000
7  e339     Khan  1800000
8  e150    Dilip  1900000
9  e131    Kiran   800000
Method 2 using read_table:
  EmpID     Name   Salary
0  e101   Pramod  1200000
1  e120   Dinesh  2200000
2  e205  Sabesta  1500000
3  e331    Harry  1700000
4  e421  Avinash  1300000
5  e231      Joy  2300000
6  e222    Smith  2100000
7  e339     Khan  1800000
8  e150    Dilip  1900000
9  e131    Kiran   800000
Method 3 using csv_reader:
['EmpID', 'Name', 'Salary']
['e101', 'Pramod', '1200000']
['e120', 'Dinesh', '2200000']
['e205', 'Sabesta', '1500000']
['e331', 'Harry', '1700000']
['e421', 'Avinash', '1300000']
['e231', 'Joy', '2300000']
['e222', 'Smith', '2100000']
['e339', 'Khan', '1800000']
['e150', 'Dilip', '1900000']
['e131', 'Kiran', '800000']
PS D:\Python>
```

Read Data from CSV Files to Pandas Dataframes

- To read the data from csv file to pandas dataframes we need to use read_csv method
- And then using the dataframes method we can convert it into dataframes.



csv_to_dataframe.py

DE > DE_Python_Coding_Challenge_Mitushi Vishwakarma > csv_to_dataframe.py > ...

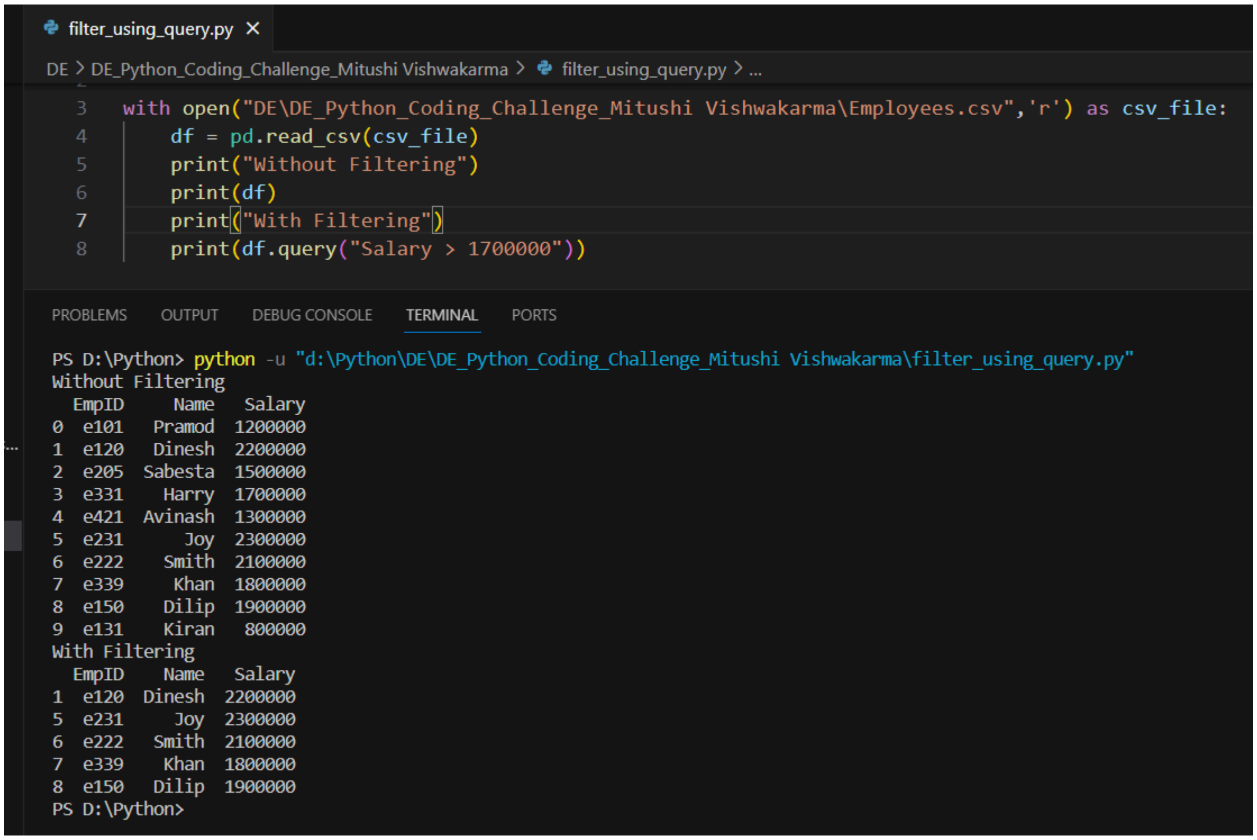
```python
import pandas as pd

with open("D:\Python\DE\Pandas\Employees.csv",'r') as csv_file:
    df = pd.read_csv(csv_file)

    dataframe = pd.DataFrame(df)
    print(dataframe)
    print(type(dataframe))
```

```
PS D:\Python> python -u "d:\Python\DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\csv
  EmpID     Name   Salary
0  e101   Pramod  1200000
1  e120   Dinesh  2200000
2  e205  Sabesta  1500000
3  e331    Harry  1700000
4  e421  Avinash  1300000
5  e231      Joy  2300000
6  e222    Smith  2100000
7  e339     Khan  1800000
8  e150    Dilip  1900000
9  e131    Kiran   800000
<class 'pandas.core.frame.DataFrame'>
PS D:\Python>
```

Filter Data in Pandas Dataframe using query.

Using query() to filter the data in Dataframe where a condition which returns boolean expression is specified in the query(). Query() allows us to write SQL-like queries to select rows based on specified conditions.
The query syntax supports various comparison operators, logical operators (and, or, not), and parentheses to create complex conditions.

In the below code, we are filtering the Employees.csv data having column Salary where Salary > 1700000.


filter_using_query.py
DE > DE_Python_Coding_Challenge_Mitushi Vishwakarma > filter_using_query.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```python
with open("DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\Employees.csv",'r') as csv_file:
    df = pd.read_csv(csv_file)
    print("Without Filtering")
    print(df)
    print("With Filtering")
    print(df.query("Salary > 1700000"))
```

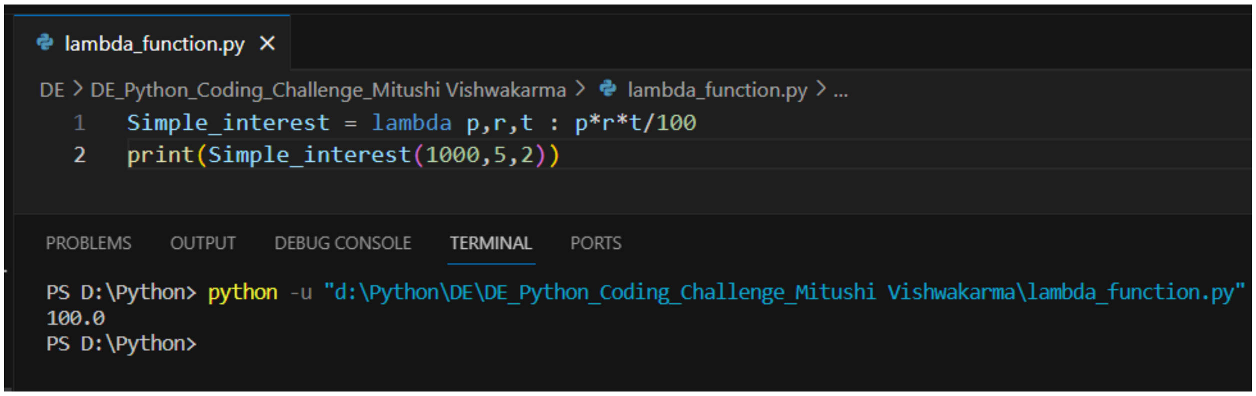
```
PS D:\Python> python -u "d:\Python\DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\filter_using_query.py"
Without Filtering
  EmpID     Name   Salary
0  e101   Pramod  1200000
1  e120   Dinesh  2200000
2  e205  Sabesta  1500000
3  e331    Harry  1700000
4  e421  Avinash  1300000
5  e231      Joy  2300000
6  e222    Smith  2100000
7  e339     Khan  1800000
8  e150    Dilip  1900000
9  e131    Kiran   800000
With Filtering
  EmpID    Name   Salary
1  e120  Dinesh  2200000
5  e231     Joy  2300000
6  e222   Smith  2100000
7  e339    Khan  1800000
8  e150   Dilip  1900000
PS D:\Python>
```

Q2 . Execute with one example Lambda Functions in Python

Lambda functions are anonymous or nameless functions written in a single expression.
The syntax is like:

    lambda arguments: expression.

Lambda works for only one expression.
In the below code, we defined a lambda function Simple_interest with three arguments p,r,t.

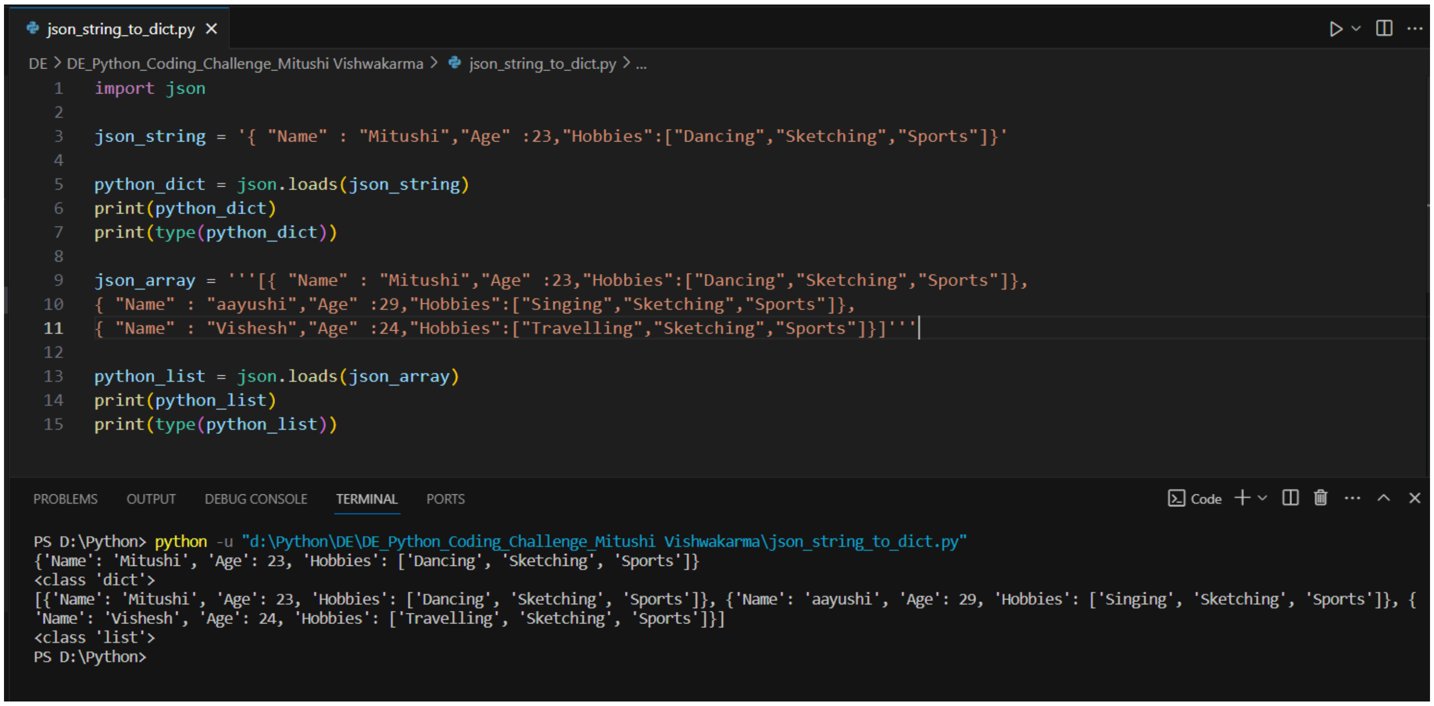

lambda_function.py
DE > DE_Python_Coding_Challenge_Mitushi Vishwakarma > lambda_function.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS

```python
Simple_interest = lambda p,r,t : p*r*t/100
print(Simple_interest(1000,5,2))
```

```
PS D:\Python> python -u "d:\Python\DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\lambda_function.py"
100.0
PS D:\Python>
```

Read JSON Strings to Python dicts or lists

To convert JSON strings to Python dictionaries or lists, you can use the json module, which is part of the Python standard library.The method loads() from json module is used to convert json strings into python dictionaries.
If Json has arrays then loads convert it into python lists.


json_string_to_dict.py
DE > DE_Python_Coding_Challenge_Mitushi Vishwakarma > json_string_to_dict.py > ...
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code

```python
import json

json_string = '{ "Name" : "Mitushi","Age" :23,"Hobbies":["Dancing","Sketching","Sports"]}'

python_dict = json.loads(json_string)
print(python_dict)
print(type(python_dict))

json_array = '''[{ "Name" : "Mitushi","Age" :23,"Hobbies":["Dancing","Sketching","Sports"]},
{ "Name" : "aayushi","Age" :29,"Hobbies":["Singing","Sketching","Sports"]},
{ "Name" : "Vishesh","Age" :24,"Hobbies":["Travelling","Sketching","Sports"]}]'''

python_list = json.loads(json_array)
print(python_list)
print(type(python_list))
```

```
PS D:\Python> python -u "d:\Python\DE\DE_Python_Coding_Challenge_Mitushi Vishwakarma\json_string_to_dict.py"
{'Name': 'Mitushi', 'Age': 23, 'Hobbies': ['Dancing', 'Sketching', 'Sports']}
<class 'dict'>
[{'Name': 'Mitushi', 'Age': 23, 'Hobbies': ['Dancing', 'Sketching', 'Sports']}, {'Name': 'aayushi', 'Age': 29, 'Hobbies': ['Singing', 'Sketching', 'Sports']}, {
'Name': 'Vishesh', 'Age': 24, 'Hobbies': ['Travelling', 'Sketching', 'Sports']}]
<class 'list'>
PS D:\Python>
```