

Q1. Create a Pandas Series that contains the following data: 4, 8, 15, 16, 23, and 42. Then, print the series.

```
In [2]: import pandas as pd

data = [4, 8, 15, 16, 23, 42]
series = pd.Series(data)
print(series)

0      4
1      8
2     15
3     16
4     23
5     42
dtype: int64
```

Q2. Create a variable of list type containing 10 elements in it, and apply pandas.Series function on the variable print

```
In [3]: data_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
series_from_list = pd.Series(data_list)
print(series_from_list)

0      1
1      2
2      3
3      4
4      5
5      6
6      7
7      8
8      9
9     10
dtype: int64
```

Q3. Create a Pandas DataFrame that contains the following data: Name Alice Bob Claire

Age 25 30 27

Gender Female Male Female

Then, print the DataFrame.

```
In [4]: import pandas as pd

# Create three Series
names = pd.Series(['Alice', 'Bob', 'Claire'], name='Name')
ages = pd.Series([25, 30, 27], name='Age')
genders = pd.Series(['Female', 'Male', 'Female'], name='Gender')

# Combine the Series into a DataFrame
df = pd.DataFrame({'Name': names, 'Age': ages, 'Gender': genders})

print(df)
```

| | Name | Age | Gender |
|---|--------|-----|--------|
| 0 | Alice | 25 | Female |
| 1 | Bob | 30 | Male |
| 2 | Claire | 27 | Female |

Q4. What is 'DataFrame' in pandas and how is it different from pandas.series? Explain with an example.

A DataFrame in pandas is a two-dimensional, size-mutable, and potentially heterogeneous tabular data structure with labeled axes (rows and columns). It is like a spreadsheet or SQL table where you can store and manipulate data. A DataFrame can contain multiple columns, and each column can have a different data type.

A Series, on the other hand, is a one-dimensional labeled array. It's like a single column in a DataFrame. Series is used to store and manipulate one-dimensional data.

Here's an example that illustrates the difference:

```
In [5]: import pandas as pd

# Creating a Series
series = pd.Series([1, 2, 3, 4, 5], name='Numbers')

# Creating a DataFrame with the same data
df = pd.DataFrame({'Numbers': [1, 2, 3, 4, 5]})

print("Series:")
print(series)

print("\nDataFrame:")
print(df)
```

```
Series:
0    1
1    2
2    3
3    4
4    5
Name: Numbers, dtype: int64
```

```
DataFrame:
   Numbers
0        1
1        2
2        3
3        4
4        5
```

Q5. What are some common functions you can use to manipulate data in a Pandas DataFrame? Can you give an example of when you might use one of these functions?

Some common functions to manipulate data in a Pandas DataFrame include:

`head()` and `tail()`: Used to display the first or last few rows of the DataFrame. Useful for quick data inspection.

Example:

```
In [ ]: df.head(3) # Display the first 3 rows of the DataFrame
```

describe(): Provides summary statistics of the numeric columns in the DataFrame.

Example:

```
In [ ]: df.describe()
```

sort_values(): Sorts the DataFrame by one or more columns.

Example:

```
In [ ]: df.sort_values(by='Age', ascending=False) # Sort by the 'Age' column in descending order
```

drop(): Removes specified columns or rows from the DataFrame.

Example:

```
In [ ]: df.drop('Name', axis=1) # Remove the 'Name' column
```

apply(): Applies a custom function to the elements in the DataFrame.

Example:

```
In [ ]: df['Age'] = df['Age'].apply(lambda x: x * 2) # Double the 'Age' values for all rows
```

Q6. Which of the following is mutable in nature Series, DataFrame, Panel?

Both Series and DataFrame are mutable in nature. This means you can modify their contents (add, update, or delete data) after they have been created. Panel is an older data structure in pandas, but it's less commonly used and has been removed in more recent versions of pandas (after version 0.25.0). You should primarily work with Series and DataFrame in modern pandas.

Q7. Create a DataFrame using multiple Series. Explain with an example.

```
In [6]: import pandas as pd

# Create two Series
names = pd.Series(['Alice', 'Bob', 'Charlie'])
ages = pd.Series([25, 30, 35])

# Combine the Series into a DataFrame
df = pd.DataFrame({'Name': names, 'Age': ages})

print(df)
```

| | Name | Age |
|---|---------|-----|
| 0 | Alice | 25 |
| 1 | Bob | 30 |
| 2 | Charlie | 35 |