

###Series

```
1: import pandas as pd
import numpy as np

# a) Operations on two pandas Series
series1 = pd.Series([2, 4, 6, 8, 10])
series2 = pd.Series([1, 3, 5, 7, 10])

addition = series1 + series2
subtraction = series1 - series2
multiplication = series1 * series2
division = series1 / series2

print("Addition:")
print(addition)
print("Subtraction:")
print(subtraction)
print("Multiplication:")
print(multiplication)
print("Division:")
print(division)
```

```
1: Addition:
0    3
1    7
2   11
3   15
4   20
dtype: int64
Subtraction:
0    1
1    1
2    1
3    1
4    0
dtype: int64
Multiplication:
0     2
1    12
2    30
3    56
4   100
dtype: int64
Division:
0    2.000000
1    1.333333
2    1.200000
3    1.142857
4    1.000000
dtype: float64
```

```
1: # b) String operations on a given pandas series
string_series = pd.Series(['Hello', 'World', 'Python', 'Pandas'])
upper_case = string_series.str.upper()
lower_case = string_series.str.lower()
string_length = string_series.str.len()
print("Uppercase:")
print(upper_case)
print("Lowercase:")
print(lower_case)
print("String Length:")
print(string_length)
```

```
1: Uppercase:
0    HELLO
1    WORLD
2    PYTHON
3    PANDAS
dtype: object
Lowercase:
0    hello
1    world
2    python
3    pandas
dtype: object
String Length:
0     5
1     5
2     6
3     6
dtype: int64
```

```
1: # c) Remove whitespaces from string values
whitespace_series = pd.Series([' Data', 'Science ', ' Machine Learning '])
```

```

remove_whitespace = whitespace_series.str.strip()
remove_left_whitespace = whitespace_series.str.lstrip()
remove_right_whitespace = whitespace_series.str.rstrip()

print("Remove Whitespace:")
print(remove_whitespace)
print("Remove Left Whitespace:")
print(remove_left_whitespace)
print("Remove Right Whitespace:")
print(remove_right_whitespace)

```

```

1: Remove Whitespace:
0      Data
1      Science
2  Machine Learning
dtype: object
Remove Left Whitespace:
0      Data
1      Science
2  Machine Learning
dtype: object
Remove Right Whitespace:
0      Data
1      Science
2  Machine Learning
dtype: object

```

```

1: # d) Create a series from a list, numpy array, and dictionary
list_series = pd.Series([1, 2, 3, 4, 5])
array_series = pd.Series(np.array([10, 20, 30, 40, 50]))
dict_series = pd.Series({'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 500})
print("List Series:", list_series)
print("Array Series:", array_series)
print("Dictionary Series:", dict_series)

```

```

1: List Series: 0      1
1      2
2      3
3      4
4      5
dtype: int64
Array Series: 0      10
1      20
2      30
3      40
4      50
dtype: int64
Dictionary Series: a      100
b      200
c      300
d      400
e      500
dtype: int64

```

```

1: # e) Calculate the number of characters in each word in a series
word_series = pd.Series(['apple', 'banana', 'cherry', 'date'])
word_length = word_series.str.len()
print("Word Length:")
print(word_length)

```

```

1: Word Length:
0      5
1      6
2      6
3      4
dtype: int64

```

```

1: # f) Compare two Pandas Series
comparison_result = (series1 == series2)
print("Comparison Result:")
print(comparison_result)

```

```

1: Comparison Result:
0      False
1      False
2      False
3      False
4       True
dtype: bool

```

```

1: # g) Convert a Pandas Series to a Python list and its type
list_from_series = series1.tolist()
type_of_series = type(series1)

```

```
print("List from Series:")
print(list_from_series)
print(type_of_series)
```

```
1: List from Series:
[2, 4, 6, 8, 10]
<class 'pandas.core.series.Series'>
```

```
1: # i) Combine many series to form a dataframe
data = {'Series1': series1, 'Series2': series2, 'ListSeries': list_series}
df = pd.DataFrame(data)
print("DataFrame:")
print(df)
```

```
1: DataFrame:
   Series1  Series2  ListSeries
0         2         1           1
1         4         3           2
2         6         5           3
3         8         7           4
4        10        10           5
```

```
1: # j) Stack two series vertically and horizontally
stacked_vertical = pd.concat([series1, series2], axis=0)
stacked_horizontal = pd.concat([series1, series2], axis=1)
print("Stacked Horizontally:")
print(stacked_horizontal)
print("Stacked Vertically:")
print(stacked_vertical)
```

```
1: Stacked Horizontally:
   0  1
0  2  1
1  4  3
2  6  5
3  8  7
4 10 10
Stacked Vertically:
   0  2
1   4
2   6
3   8
4  10
0   1
1   3
2   5
3   7
4  10
dtype: int64
```

```
1: # k) Create and display a DataFrame from a specified dictionary data with index labels
data_dict = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 35]}
index_labels = ['a', 'b', 'c']
df_with_index = pd.DataFrame(data_dict, index=index_labels)
print("DataFrame with Index:")
print(df_with_index)
```

```
1: DataFrame with Index:
   Name  Age
a  Alice  25
b   Bob   30
c  Charlie 35
```

```
1: # l) Identify frequency counts of unique items in a series
fruit_series = pd.Series(['apple', 'banana', 'apple', 'cherry', 'banana', 'apple'])
item_counts = fruit_series.value_counts()
print("Item Counts:")
print(item_counts)
```

```
1: Item Counts:
apple      3
banana     2
cherry     1
dtype: int64
```

```
1: # m) Get the items of series A not present in series B
series_a = pd.Series([1, 2, 3, 4, 5])
series_b = pd.Series([3, 4, 5, 6, 7])
items_not_in_b = series_a[~series_a.isin(series_b)]
print("Items not in Series B:")
print(items_not_in_b)
```

```
] Items not in Series B:
```

```
0    1
1    2
dtype: int64
```

```
] :
```

```
# n) Convert a numpy array to a DataFrame of a given shape
import numpy as np

numpy_array = np.array([[1, 2], [3, 4], [5, 6]])
shape_df = pd.DataFrame(numpy_array, columns=['A', 'B'], index=['A', 'B', 'C'])
print("DataFrame with Shape:")
print(shape_df)
```

```
] :
```

```
DataFrame with Shape:
```

```
   A  B
A   1  2
B   3  4
C   5  6
```