```
#1
import pandas as pd

data = pd.Series([10, 20, 30, 40, 50])

print(data)
```

```
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

```
#2
import pandas as pd

data = pd.Series([10, 20, 30, 40, 50])

data_list = data.tolist()

print("Converted list:", data_list)
print("Type of the converted list:", type(data_list))
```

```
Converted list: [10, 20, 30, 40, 50]
Type of the converted list: <class 'list'>
```

```
#3
import pandas as pd

series1 = pd.Series([2, 4, 6, 8, 10])
series2 = pd.Series([1, 3, 5, 7, 9])

addition_result = series1 + series2

subtraction_result = series1 - series2

multiplication_result = series1 * series2

division_result = series1 / series2

print("Addition:")
print(addition_result)
print("\nSubtraction:")
print(subtraction_result)
print("\nMultiplication:")
print(multiplication_result)
print("\nDivision:")
print(division_result)
```

```
Addition:
0     3
1     7
2    11
3    15
4    19
dtype: int64

Subtraction:
0    1
1    1
2    1
3    1
4    1
dtype: int64

Multiplication:
0     2
1    12
2    30
3    56
4    90
dtype: int64

Division:
```

```
0    2.000000
1    1.333333
2    1.200000
3    1.142857
4    1.111111
dtype: float64
```

```
#4
import pandas as pd

series1 = pd.Series([2, 4, 6, 8, 10])
series2 = pd.Series([1, 3, 5, 7, 10])

comparison_result = series1 == series2

print("Comparison Result:")
print(comparison_result)
```

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

```
#5
import pandas as pd

original_dict = {'a': 100, 'b': 200, 'c': 300, 'd': 400, 'e': 800}

converted_series = pd.Series(original_dict)

print("Converted series:")
print(converted_series)
```

```
Converted series:
a    100
b    200
c    300
d    400
e    800
dtype: int64
```

```
#6
import pandas as pd
import numpy as np

numpy_array = np.array([10, 20, 30, 40, 50])

converted_series = pd.Series(numpy_array)

print("Converted Pandas series:")
print(converted_series)
```

```
Converted Pandas series:
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

```
#7
import pandas as pd

original_series = pd.Series(['100', '200', 'python', '300.12', '400'])

converted_series = pd.to_numeric(original_series, errors='coerce')

print("Original Data Series:")
print(original_series)
print("\nChange the said data type to numeric:")
print(converted_series)
```

```
Original Data Series:
0       100
1       200
2    python
3    300.12
4       400
dtype: object

Change the said data type to numeric:
0    100.00
1    200.00
2       NaN
3    300.12
4    400.00
dtype: float64
```

```
#8
import pandas as pd

data = {
    'col1': [1, 2, 3, 4, 7, 11],
    'col2': [4, 5, 6, 9, 5, 0],
    'col3': [7, 5, 8, 12, 1, 11]
}

df = pd.DataFrame(data)

first_column_series = df.iloc[:, 0]

print("Original DataFrame:")
print(df)
print("\n1st column as a Series:")
print(first_column_series)
```

```
Original DataFrame:
   col1  col2  col3
0     1     4     7
1     2     5     5
2     3     6     8
3     4     9    12
4     7     5     1
5    11     0    11

1st column as a Series:
0     1
1     2
2     3
3     4
4     7
5    11
Name: col1, dtype: int64
```

```
#9
import pandas as pd

original_series = pd.Series(['100', '200', 'python', '300.12', '400'])

array_from_series = original_series.values

print("Original Data Series:")
print(original_series)
print("\nSeries to an array:")
print(array_from_series)
print(type(array_from_series))
```

```
Original Data Series:
0       100
1       200
2    python
3    300.12
4       400
dtype: object

Series to an array:
['100' '200' 'python' '300.12' '400']
<class 'numpy.ndarray'>
```

```
#10
import pandas as pd

original_series = pd.Series([['Red', 'Green', 'White'], ['Red', 'Black'], ['Yellow']])

one_series = original_series.explode()

print("Original Series of list:")
print(original_series)
print("\nOne Series:")
print(one_series)
```

```
Original Series of list:
0    [Red, Green, White]
1           [Red, Black]
2              [Yellow]
dtype: object

One Series:
0       Red
0     Green
0     White
1       Red
1     Black
2    Yellow
dtype: object
```