Pandas Series exercises

```
# Import the numpy package under the name np
import numpy as np

# Import the pandas package under the name pd
import pandas as pd

# Print the pandas version and the configuration
print(pd.__version__)
```

```
1.5.3
```

Series creation

Create an empty pandas Series

```
# your code goes here
pd.Series() #S harus kapital
```

```
<ipython-input-16-c9dfc8139efe>:2: FutureWarning: The default dtype for empty Series will be 'object' instead of 'float64' in a future v
  pd.Series()
Series([], dtype: float64)
```

Given the X python list convert it to an Y pandas Series

```
# your code goes here
X= ['A', 'B', 'C', 'D', 'E', 'F']
print(X, type(X))
Y = pd.Series([4.56, 5.67, 6.78, 7.89, 8.90, 2.34])
print(Y, type(Y))
```

```
['A', 'B', 'C', 'D', 'E', 'F'] <class 'list'>
0    4.56
1    5.67
2    6.78
3    7.89
4    8.90
5    2.34
dtype: float64 <class 'pandas.core.series.Series'>
```

Given the X pandas Series, name it 'My letters'

```
# your code goes here
X =pd.Series (['A', 'B', 'C', 'D', 'E', 'F'])

X.name = 'My Letters'
X
```

```
0    A
1    B
2    C
3    D
4    E
5    F
Name: My Letters, dtype: object
```

Given the X pandas Series, show its values

```
# your code goes here
X = pd.Series(['A', 'B', 'C', 'D', 'E', 'F'])

X.values
```

```
array(['A', 'B', 'C', 'D', 'E', 'F'], dtype=object)
```

Series indexation

Assign index names to the given X pandas Series

```
# your code goes here
X = pd.Series(['A', 'B', 'C', 'D', 'E', 'F'])
index_names = ['one', 'two', 'three', 'four', 'five', 'six']

X.index = index_names
X
```

```
one      A
two      B
three    C
four     D
five     E
six      F
dtype: object
```

Given the X pandas Series, show its first element

```
# your code goes here
X = pd.Series(['A', 'B', 'C', 'D', 'E', 'F'], index=['one', 'two', 'three', 'four', 'five', 'six'])

#by position : X[0]
#by position : X.iloc[0]
X['one'] #by index
```

```
'A'
```

Given the X pandas Series, show its last element

```
# your code goes here
#by position : X[-1]
#by position : X.iloc[-1]
#by index : X.loc['six']
X['six'] #by index
```

```
'F'
```

Given the X pandas Series, show all middle elements

```
# your code goes here
#middle with index : X[['two', 'three', 'four', 'five']]
#middle with position : X.iloc[1:-1]
#middle with index : X.loc['two':'five']
X[1:-1] #middle with position
```

```
two      B
three    C
four     D
five     E
dtype: object
```

Given the X pandas Series, show the elements in reverse position

```
# your code goes here
X[::-1]
```

```
six      F
five     E
four     D
three    C
two      B
one      A
dtype: object
```

Given the X pandas Series, show the first and last elements

```
# your code goes here
#Search 1 and Last : by position ----- X[[0, -1]]
#Search 1 and Last : by index -------- X[['one', 'six']]
#Search 1 and Last : by position ----- X.iloc[[0, -1]]
#Search 1 and Last : by index --------
X.loc[['one', 'six']]
```

```
one    A
six    F
dtype: object
```

Series manipulation

Convert the given integer pandas Series to float

```
# your code goes here
pd.Series(X, dtype=float)
```

```
first     A
second    B
third     C
forth     D
fifth     E
dtype: object
```

Reverse the given pandas Series (first element becomes last)

```
# your code goes here
X[::-1]
```

```
fifth     E
forth     D
third     C
second    B
```

```
first     A
dtype: object
```

- Order (sort) the given pandas Series

```
# your code goes here
X = X.sort_values()
X
```

```
first     A
second    B
third     C
forth     D
fifth     E
dtype: object
```

- Given the X pandas Series, set the fifth element equal to 10

```
# your code goes here
X[4] = 10
X
```

```
first      A
second     B
third      C
forth      D
fifth     10
dtype: object
```

- Given the X pandas Series, change all the middle elements to 0

```
# your code goes here
X[1:-1] = 0
X
```

```
first      A
second     0
third      0
forth      0
fifth     10
dtype: object
```

- Given the X pandas Series, add 5 to every element

```
# your code goes here
X = pd.Series([1, 2, 3, 4, 5, 6]) #make new series without index
X + 5
```

```
0     6
1     7
2     8
3     9
4    10
5    11
dtype: int64
```

- Series boolean arrays (also called masks)

Given the X pandas Series, make a mask showing negative elements

```
# your code goes here
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10]) #new series without index

mask = X <= 0
mask
```

```
0     True
1    False
2     True
3     True
4    False
5    False
6     True
7     True
8     True
9    False
dtype: bool
```

Given the X pandas Series, get the negative elements

```
# your code goes here
mask = X <= 0
X[mask]
```

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

Given the X pandas Series, get numbers higher than 5

```
# your code goes here
mask = X > 5
X[mask]
```

```
5     6
9    10
dtype: int64
```

Given the X pandas Series, get numbers higher than the elements mean

```
# your code goes here
mask = X > X.mean()
X[mask]
```

```
1     2
4     5
5     6
9    10
dtype: int64
```

Given the X pandas Series, get numbers equal to 2 or 10

```
# your code goes here
mask = (X == 2) | (X == 10)
X[mask]
```

```
1     2
9    10
dtype: int64
```

Logic functions

Given the X pandas Series, return True if none of its elements is zero

```
# your code goes here
X.all()
```

```
False
```

Given the X pandas Series, return True if any of its elements is zero

```
# your code goes here
X.any()
```

```
True
```

Summary statistics

Given the X pandas Series, show the sum of its elements

```
# your code goes here
X = pd.Series([3,5,6,7,2,3,4,9,4])
X.sum()
```

```
43
```

Given the X pandas Series, show the mean value of its elements

```
# your code goes here
X = pd.Series([1,2,0,4,5,6,0,0,9,10])
X.mean()
```

```
3.7
```

Given the X pandas Series, show the max value of its elements

```
# your code goes here
X.max()
```

```
10
```