Pandas Series exercises

In [1]:

```
# 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__)
```

```
0.25.3
```

Series creation

Create an empty pandas Series

In [ ]:

```
# your code goes here
```

In [ ]:

```
pd.Series()
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = ['A','B','C']
print(X, type(X))

Y = pd.Series(X)
print(Y, type(Y)) # different type
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C'])

X.name = 'My letters'
X
```

Given the X pandas Series, show its values

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C'])

X.values
```

Series indexation

Assign index names to the given X pandas Series

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C'])
index_names = ['first', 'second', 'third']

X.index = index_names
X
```

Given the X pandas Series, show its first element

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C'], index=['first', 'second', 'third'])

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

Given the X pandas Series, show its last element

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C'], index=['first', 'second', 'third'])

#X[-1] # by position
#X.iloc[-1] # by position
X['third'] # by index
```

Given the X pandas Series, show all middle elements

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C','D','E'],
              index=['first','second','third','forth','fifth'])

#X[['second', 'third', 'forth']]
#X.iloc[1:-1] # by position
X[1:-1] # by position
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C','D','E'],
              index=['first','second','third','forth','fifth'])

#X.iloc[::-1]
X[::-1]
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series(['A','B','C','D','E'],
              index=['first','second','third','forth','fifth'])

#X[['first', 'fifth']]
#X.iloc[[0, -1]]
X[[0, -1]]
```

Series manipulation

Convert the given integer pandas Series to float

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([1,2,3,4,5],
              index=['first','second','third','forth','fifth'])

pd.Series(X, dtype=np.float)
```

Reverse the given pandas Series (first element becomes last)

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([1,2,3,4,5],
              index=['first','second','third','forth','fifth'])

X[::-1]
```

Order (sort) the given pandas Series

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([4,2,5,1,3],
              index=['forth','second','fifth','first','third'])

X = X.sort_values()
X
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([1,2,3,4,5],
              index=['A','B','C','D','E'])

X[4] = 10
X
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([1,2,3,4,5],
              index=['A','B','C','D','E'])

X[1:-1] = 0
X
```

Given the X pandas Series, add 5 to every element

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([1,2,3,4,5])

X + 5
```

Series boolean arrays (also called masks)

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10])

mask = X <= 0
mask
```

Given the X pandas Series, get the negative elements

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10])

mask = X <= 0
X[mask]
```

Given the X pandas Series, get numbers higher than 5

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10])

mask = X > 5
X[mask]
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10])

mask = X > X.mean()
X[mask]
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10])

mask = (X == 2) | (X == 10)
X[mask]
```

Logic functions

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10])

X.all()
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([-1,2,0,-4,5,6,0,0,-9,10])

X.any()
```

Summary statistics

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([3,5,6,7,2,3,4,9,4])

#np.sum(X)
X.sum()
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([1,2,0,4,5,6,0,0,9,10])

#np.mean(X)
X.mean()
```

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

In [ ]:

```
# your code goes here
```

In [ ]:

```
X = pd.Series([1,2,0,4,5,6,0,0,9,10])

#np.max(X)
X.max()
```