The Pandas Series object

Challenge 1.

Suppose that you're given these two data structures:

In [1]:
```
superheroes = [
"Batman",
"Superman",
"Spider-Man",
"Iron Man",
"Captain America",
"Wonder Woman"
]
strength_levels = (100, 120, 90, 95, 110, 120)
```

Here are your challenges:

1. Use the list of superheroes to populate a new Series object.
2. Use the tuple of strengths to populate a new Series object.
3. Create a Series with the superheroes as index labels and the strength levels as the values. Assign the Series to a heroes variable.
4. Extract the first two rows of the heroes Series.
5. Extract the last four rows of the heroes Series.
6. Determine the number of unique values in your heroes Series.
7. Calculate the average strength of the superheroes in heroes.
8. Calculate the maximum and minimum strengths in heroes.
9. Calculate what each superhero's strength level would be if it doubled.
10. Convert the heroes Series to a Python dictionary.

Solution

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

Use the list of superheroes to populate a new Series object.

In [3]:
```
#Assigning new series variable "new_superheroes" & Using pd.Series to create our ne
new_superheroes = pd.Series(superheroes)
new_superheroes
```

Out[3]:
```
0            Batman
1          Superman
2        Spider-Man
3          Iron Man
4    Captain America
5       Wonder Woman
dtype: object
```

Use the tuple of strengths to populate a new Series object.

In [4]:
```
#Assigning new series variable "new_strengths" & Using pd.Series to create our new
new_strengths = pd.Series(strength_levels)
print(new_strengths)
```

```
0    100
1    120
2     90
3     95
4    110
5    120
dtype: int64
```

Create a Series with the superheroes as index labels and the strength levels as the values. Assign the Series to a heroes variable.

In [5]:
```
# Assigned a heroes variable. The index parameter passes superheroes list as index
heroes = pd.Series(index = superheroes, data = strength_levels)
heroes
```

Out[5]:
```
Batman            100
Superman          120
Spider-Man         90
Iron Man           95
Captain America   110
Wonder Woman      120
dtype: int64
```

Extract the first two rows of the heroes Series.

In [6]:
```
first2rows = heroes.iloc[:2]
first2rows
```

Out[6]:
```
Batman      100
Superman    120
dtype: int64
```

Extract the last four rows of the heroes Series.

In [7]:
```
last4rows = heroes.iloc[2:]
last4rows
```

Out[7]:
```
Spider-Man         90
Iron Man           95
Captain America   110
Wonder Woman      120
dtype: int64
```

In [8]:
```
last4_rows = heroes.iloc[-4:]
last4_rows
```

Out[8]:
```
Spider-Man         90
Iron Man           95
Captain America   110
Wonder Woman      120
dtype: int64
```

Determine the number of unique values in your heroes Series.

In [9]:
```
heroes.unique()
```

Out[9]:
```
array([100, 120,  90,  95, 110], dtype=int64)
```

In [10]:
```
unique_values = heroes.unique()
unique_values
```

Out[10]:
```
array([100, 120,  90,  95, 110], dtype=int64)
```

Average, Maximum and Minimum Values

In [11]:
```
heroes.mean()
```

Out[11]:
```
105.83333333333333
```

In [12]:
```
average_strength = heroes.mean()
average_strength
```

Out[12]:
```
105.83333333333333
```

In [13]:
```
heroes.min()
```

Out[13]:
```
90
```

In [14]:
```
min_strength = heroes.min()
min_strength
```

Out[14]:
```
90
```

In [15]:
```
heroes.max()
```

Out[15]:
```
120
```

In [16]:
```
max_strength = heroes.max()
max_strength
```

Out[16]:
```
120
```

Calculate what each superhero's strength level would be if it doubled.

In [17]:
```
heroes * 2
```

Out[17]:
```
Batman             200
Superman           240
Spider-Man         180
Iron Man           190
Captain America    220
Wonder Woman       240
dtype: int64
```

Convert the heroes Series to a Python dictionary.

In [18]:
```
heroes_dictionary = heroes.to_dict()
heroes_dictionary
```

Out[18]:
```
{'Batman': 100,
 'Superman': 120,
 'Spider-Man': 90,
 'Iron Man': 95,
 'Captain America': 110,
 'Wonder Woman': 120}
```

In [ ]: