In [15]:
```
import numpy as np

from pandas import Series,DataFrame
import pandas as pd
```

In [16]:
```
#Lets create a Series (array of data and data labels, its index)

obj = Series([3,6,9,12])

#Show
obj
```

Out[16]:
```
0     3
1     6
2     9
3    12
dtype: int64
```

In [17]:
```
#Lets show the values
obj.values
```

Out[17]:
```
array([ 3,  6,  9, 12], dtype=int64)
```

In [18]:
```
#Lets show the index
obj.index
```

Out[18]:
```
Int64Index([0, 1, 2, 3], dtype='int64')
```

In [21]:
```
#Now lets create a Series with an index

#WW2 casualties
ww2_cas = Series([8700000,4300000,3000000,2100000,400000],index=['USSR','Germany'

#Show
ww2_cas
```

Out[21]:
```
USSR       8700000
Germany    4300000
China      3000000
Japan      2100000
USA         400000
dtype: int64
```

In [22]:
```
#Now we can use index values to select Series values
ww2_cas['USA']
```

Out[22]:
```
400000
```

In [26]:

```
#Can also check with array operations

#Check who had casualties greater than 4 million
ww2_cas[ww2_cas>4000000]
```

Out[26]:

```
USSR       8700000
Germany    4300000
dtype: int64
```

In [27]:

```
#Can treat Series as ordered dictionary

#Check if USSR is in Series
'USSR' in ww2_cas
```

Out[27]:

```
True
```

In [31]:

```
#Can convert Series into Python dictionary
ww2_dict = ww2_cas.to_dict()

#Show
ww2_dict
```

Out[31]:

```
{'China': 3000000,
 'Germany': 4300000,
 'Japan': 2100000,
 'USA': 400000,
 'USSR': 8700000}
```

In [34]:

```
#Can convert back into a Series
WW2_Series = Series(ww2_dict)
```

In [35]:

```
#Show
WW2_Series
```

Out[35]:

```
China      3000000
Germany    4300000
Japan      2100000
USA         400000
USSR       8700000
dtype: int64
```

In [36]:

```
#Passing a dictionary the index will have the dict keys in order
countries = ['China','Germany','Japan','USA','USSR','Argentina']
```

In [37]:

```
#Lets redefine a Series
obj2 = Series(ww2_dict,index=countries)
```

In [38]:

```
#Show
obj2
```

Out[38]:

```
China        3000000
Germany      4300000
Japan        2100000
USA           400000
USSR         8700000
Argentina        NaN
dtype: float64
```

In [39]:

```
#We can use isnull and notnull to find missing data
pd.isnull(obj2)

#obj2.isnull()
```

Out[39]:

```
China        False
Germany      False
Japan        False
USA          False
USSR         False
Argentina     True
dtype: bool
```

In [40]:

```
#Same for the opposite
pd.notnull(obj2)

#obj2.notnull()
```

Out[40]:

```
China         True
Germany       True
Japan         True
USA           True
USSR          True
Argentina    False
dtype: bool
```

In [41]:

```
#Lets see the ww2 Series again
WW2_Series
```

Out[41]:

```
China      3000000
Germany    4300000
Japan      2100000
USA         400000
USSR       8700000
dtype: int64
```

```
In [42]: #Lets check our Series with Argentine again
         obj2
```

```
Out[42]: China         3000000
         Germany       4300000
         Japan         2100000
         USA            400000
         USSR          8700000
         Argentina         NaN
         dtype: float64
```

```
In [43]: #Now we can add and pandas automatically aligns data by index
         WW2_Series + obj2
```

```
Out[43]: Argentina          NaN
         China          6000000
         Germany        8600000
         Japan          4200000
         USA             800000
         USSR          17400000
         dtype: float64
```

```
In [45]: #We can give Series names
         obj2.name = "World War 2 Casualties"
```

```
In [46]: #Show
         obj2
```

```
Out[46]: China         3000000
         Germany       4300000
         Japan         2100000
         USA            400000
         USSR          8700000
         Argentina         NaN
         Name: World War 2 Casualties, dtype: float64
```

```
In [47]: #We can also name index
         obj2.index.name = 'Countries'
```

```
In [48]: #Show
         obj2
```

```
Out[48]: Countries
         China         3000000
         Germany       4300000
         Japan         2100000
         USA            400000
         USSR          8700000
         Argentina         NaN
         Name: World War 2 Casualties, dtype: float64
```

```
In [49]: #Next we'll learn DataFrames!
In [ ]:
```