

In [1]:

```
import numpy as np

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

In [2]:

```
#Lets create a Series (array of data and data labels, its index)
obj = Series([3,6,9,12])

#Show
obj
```

Out[2]:

```
0    3
1    6
2    9
3   12
dtype: int64
```

In [3]:

```
obj[:2]
```

Out[3]:

```
0    3
1    6
dtype: int64
```

In [4]:

```
#Lets show the values
obj.values
```

Out[4]:

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

In [5]:

```
#Lets show the index
obj.index
```

Out[5]:

```
RangeIndex(start=0, stop=4, step=1)
```

In [6]:

```
#Now Lets create a Series with an index
#WW2 casualties
ww2_cas = Series([8700000,4300000,3000000,2100000,400000],index=['USSR','Germany','China',
'a','Japan','USA'])

#Show
ww2_cas
```

Out[6]:

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

In [7]:

```
ww2_cas.index
```

Out[7]:

```
Index(['USSR', 'Germany', 'China', 'Japan', 'USA'], dtype='object')
```

In [8]:

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

Out[8]:

```
400000
```

In [9]:

```
#Can also check with array operations

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

Out[9]:

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

In [10]:

```
#Can treat Series as ordered dictionary

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

Out[10]:

```
True
```

In [11]:

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

#Show
ww2_dict
```

Out[11]:

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

In [12]:

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

In [13]:

```
#Show
WW2_Series
```

Out[13]:

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

In [14]:

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

In [15]:

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

In [16]:

```
#Show
obj2
```

Out[16]:

```
China      3000000.0
Germany    4300000.0
Japan      2100000.0
USA         400000.0
USSR       8700000.0
Argentina      NaN
dtype: float64
```

In [17]:

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

Out[17]:

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

In [18]:

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

Out[18]:

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

In [19]:

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

Out[19]:

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

In [20]:

```
#Lets check our Series with Argentina again  
obj2
```

Out[20]:

```
China      3000000.0  
Germany    4300000.0  
Japan      2100000.0  
USA        400000.0  
USSR       8700000.0  
Argentina      NaN  
dtype: float64
```

In [21]:

```
#Now we can add and pandas automatically aligns data by index  
WW2_Series + obj2
```

Out[21]:

```
Argentina      NaN  
China          6000000.0  
Germany        8600000.0  
Japan          4200000.0  
USA            800000.0  
USSR          17400000.0  
dtype: float64
```

In [22]:

```
#We can give Series names  
obj2.name = "World War 2 Casualties"
```

In [23]:

```
#Show  
obj2
```

Out[23]:

```
China      3000000.0  
Germany    4300000.0  
Japan      2100000.0  
USA        400000.0  
USSR       8700000.0  
Argentina      NaN  
Name: World War 2 Casualties, dtype: float64
```

In [24]:

```
#We can also name index  
obj2.index.name = 'Countries'
```

In [25]:

```
#Show  
obj2
```

Out[25]:

```
Countries  
China      3000000.0  
Germany    4300000.0  
Japan      2100000.0  
USA         400000.0  
USSR       8700000.0  
Argentina      NaN  
Name: World War 2 Casualties, dtype: float64
```

In [26]:

```
#Next we'll learn DataFrames!
```

In [ ]: