Series

A Series is very similar to a NumPy array. The difference between two is that a Series can have axis labels, meaning it can be indexed by a label, instead of just a number location. It also doesn't need to hold numeric data, it can hold any arbitrary Python Object.

```
import numpy as np
import pandas as pd
```

Creating a Series

We can convert a list,numpy array, or dictionary to a Series.

** Using Lists**

```
X = [10,20,30]
Y = np.array(['a','b','c'])
type(Y)
```

```
numpy.ndarray
```

```
Y[0]
```

```
'a'
```

```
# A = pd.Series(data=X)
A = pd.Series([1,2,3])
type(A)
```

```
pandas.core.series.Series
```

```
A
```

```
0    1
1    2
2    3
dtype: int64
```

```
B=pd.Series([10,20],['dog','cats'])
print(B)
print(B)

     dog     10
     cats    20
     dtype: int64
```

```
B['a']

     1
```

** Using NumPy Arrays **

```
X1 = np.array([10,20,30])
Y1 = ['a', 'b', 'c']
```

```
pd.Series(X1)

     0    10
     1    20
     2    30
     dtype: int32
```

```
pd.Series(X1, Y1)

     a    10
     b    20
     c    30
     dtype: int32
```

** Using Dictionary**

```
X3 = {'x':10,'y':20,'z':30}
```

```
pd.Series(X3)

     x    10
     y    20
     z    30
     dtype: int64
```

Using an Index

The key to using a Series is understanding its index. Pandas makes use of these index names or numbers by allowing for fast look ups of information.

```
a = pd.Series([1,2,3,4],index = ['USA', 'Germany','USSR', 'Japan'])
```

```
a = pd.Series([1,2,3,4],index = ['USA', 'Germany','USSR', 'Japan'])
```

```
a
```

```
USA        1
Germany    2
USSR       3
Japan      4
dtype: int64
```

```
b = pd.Series([1,2,5,4],index = ['USA', 'Germany','Italy', 'Japan'])
```

```
b
```

```
USA        1
Germany    2
Italy      5
Japan      4
dtype: int64
```

```
a['USA']
```

```
1
```

```
a + b
```

```
Germany    4.0
Italy      NaN
Japan      8.0
USA        2.0
USSR       NaN
dtype: float64
```