In [1]:

```
#Now we'll elarn about reindexing
```

In [23]:

```
import numpy as np
from pandas import Series, DataFrame
import pandas as pd
from numpy.random import randn
```

In [4]:

```
#Lets create a new series
ser1 = Series([1,2,3,4],index=['A','B','C','D'])
```

In [5]:

```
#Show
ser1
```

Out[5]:

```
A    1
B    2
C    3
D    4
dtype: int64
```

In [6]:

```
#Call reindex to rearrange the data to a new index
ser2 = ser1.reindex(['A','B','C','D','E','F'])
```

In [7]:

```
#Show
ser2
```

Out[7]:

```
A      1
B      2
C      3
D      4
E    NaN
F    NaN
dtype: float64
```

In [13]:

```
# We can alos fill in values for new indexes
ser2.reindex(['A','B','C','D','E','F','G'],fill_value=0)
```

Out[13]:

```
A      1
B      2
C      3
D      4
E    NaN
F    NaN
G      0
dtype: float64
```

In [15]:
```
#Using a particular method for filling values
ser3 = Series(['USA','Mexico','Canada'],index=[0,5,10])

#Show
ser3
```

Out[15]:
```
0        USA
5     Mexico
10    Canada
dtype: object
```

In [20]:
```
#Can use a forward fill for interploating values vetween indices
ser3.reindex(range(15),method='ffill')
```

Out[20]:
```
0        USA
1        USA
2        USA
3        USA
4        USA
5     Mexico
6     Mexico
7     Mexico
8     Mexico
9     Mexico
10    Canada
11    Canada
12    Canada
13    Canada
14    Canada
dtype: object
```

In [29]:
```
#Reindexing rows, columns or both

#Lets make a datafram ewith some random values
dframe = DataFrame(randn(25).reshape((5,5)),index=['A','B','D','E','F'],columns=[

#Show
dframe
```

In [37]:
```
#Notice we forgot 'C' , lets reindex it into dframe
dframe2 = dframe.reindex(['A','B','C','D','E','F'])
```

In [38]:

```
#Can also explicitly reindex columns
new_columns = ['col1','col2','col3','col4','col5','col6']

dframe2.reindex(columns=new_columns)
```

Out[38]:

	col1	col2	col3	col4	col5	col6
A	0.442886	0.128326	0.091837	0.018814	-1.419262	NaN
B	0.650227	0.438677	-0.208091	1.263344	0.147084	NaN
C	NaN	NaN	NaN	NaN	NaN	NaN
D	-1.089677	1.616700	1.856270	0.553392	1.046027	NaN
E	0.608385	-0.316224	0.341510	-0.085305	1.015604	NaN
F	1.054495	-0.274723	1.251613	0.160329	-0.527855	NaN

In [40]:

```
#Reindex quickly using the label-indexing with ix (we'll see this more in the fut

#Show original
dframe
```

Out[40]:

	col1	col2	col3	col4	col5
A	0.442886	0.128326	0.091837	0.018814	-1.419262
B	0.650227	0.438677	-0.208091	1.263344	0.147084
D	-1.089677	1.616700	1.856270	0.553392	1.046027
E	0.608385	-0.316224	0.341510	-0.085305	1.015604
F	1.054495	-0.274723	1.251613	0.160329	-0.527855

In [41]:

```
dframe.ix[['A','B','C','D','E','F'],new_columns]
```

Out[41]:

	col1	col2	col3	col4	col5	col6
A	0.442886	0.128326	0.091837	0.018814	-1.419262	NaN
B	0.650227	0.438677	-0.208091	1.263344	0.147084	NaN
C	NaN	NaN	NaN	NaN	NaN	NaN
D	-1.089677	1.616700	1.856270	0.553392	1.046027	NaN
E	0.608385	-0.316224	0.341510	-0.085305	1.015604	NaN
F	1.054495	-0.274723	1.251613	0.160329	-0.527855	NaN

In [ ]: