In [1]:
```
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
```

In [13]:
```
# Making a DataFrame
dframe= DataFrame(np.arange(12).reshape((3, 4)),
                  index=['NY', 'LA', 'SF'],
                  columns=['A', 'B', 'C', 'D'])

#Show
dframe
```

Out[13]:

	A	B	C	D
NY	0	1	2	3
LA	4	5	6	7
SF	8	9	10	11

In [14]:
```
# Just like a Series, axis indexes can also use map

#Let's use map to lowercase the city initials
dframe.index.map(str.lower)
```

Out[14]:
```
array(['ny', 'la', 'sf'], dtype=object)
```

In [25]:
```
# If you want to assign this to the actual index, you can use index
dframe.index = dframe.index.map(str.lower)
#Show
dframe
```

Out[25]:

	A	B	C	D
ny	0	1	2	3
la	4	5	6	7
sf	8	9	10	11

In [28]:
```
# Use rename if you want to create a transformed version withour modifying the or

#str.title will capitalize the first letter, lowercasing the columns
dframe.rename(index=str.title, columns=str.lower)
```

Out[28]:

	a	b	c	d
Ny	0	1	2	3
La	4	5	6	7
Sf	8	9	10	11

In [34]:
```
# We can also use rename to insert dictionaries providing new values for indexes
dframe.rename(index={'ny': 'NEW YORK'},
              columns={'A': 'ALPHA'})
```

Out[34]:

	ALPHA	B	C	D
NEW YORK	0	1	2	3
la	4	5	6	7
sf	8	9	10	11

In [38]:
```
# If you would like to actually edit the data set in place, set inplace=True
dframe.rename(index={'ny': 'NEW YORK'}, inplace=True)
dframe
```

Out[38]:

	A	B	C	D
NEW YORK	0	1	2	3
la	4	5	6	7
sf	8	9	10	11

In [1]:
```
#Up next: Binning!
```

In [ ]:
```

```