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

In [2]:
```
#Let's see how stack and unstack work

# Create DataFrame
dframe1 = DataFrame(np.arange(8).reshape((2, 4)),
                    index=pd.Index(['LA', 'SF'], name='city'),
                    columns=pd.Index(['A', 'B', 'C','D'], name='letter'))
#Show
dframe1
```

Out[2]:

letter	A	B	C	D
city				
LA	0	1	2	3
SF	4	5	6	7

In [7]:
```
# Use stack to pivot the columns into the rows
dframe_st = dframe1.stack()

#Show
dframe_st
```

Out[7]:
```
city  letter
LA    A         0
      B         1
      C         2
      D         3
SF    A         4
      B         5
      C         6
      D         7
dtype: int32
```

In [8]:
```
#We can always rearrange back into a DataFrame
dframe_st.unstack()
```

Out[8]:

letter	A	B	C	D
city				
LA	0	1	2	3
SF	4	5	6	7

In [10]:
```
#We can choose which level to unstack by
dframe_st.unstack(0)
```

Out[10]:

city	LA	SF
letter		
A	0	4
B	1	5
C	2	6
D	3	7

In [12]:
```
# Also by which name to unstack by
dframe_st.unstack('letter')
```

Out[12]:

letter	A	B	C	D
city				
LA	0	1	2	3
SF	4	5	6	7

In [13]:
```
# Also by which name to unstack by
dframe_st.unstack('city')
```

Out[13]:

city	LA	SF
letter		
A	0	4
B	1	5
C	2	6
D	3	7

In [15]:
```
# Let's see how stack and unstack handle NAN

#Make two series
ser1 = Series([0, 1, 2], index=['Q', 'X', 'Y'])
ser2 = Series([4, 5, 6], index=['X', 'Y', 'Z'])

#Concat to make a dframe
dframe = pd.concat([ser1, ser2], keys=['Alpha', 'Beta'])

# Unstack resulting DataFrame
dframe.unstack()
```

Out[15]:

	Q	X	Y	Z
Alpha	0	1	2	NaN
Beta	NaN	4	5	6

In [16]:
```
# Now stack will filter out NAN by default
dframe.unstack().stack()
```

Out[16]:
```
Alpha  Q    0
       X    1
       Y    2
Beta   X    4
       Y    5
       Z    6
dtype: float64
```

In [17]:
```
# IF we dont want this we can set it to False
dframe.unstack().stack(dropna=False)
```

Out[17]:
```
Alpha  Q      0
       X      1
       Y      2
       Z    NaN
Beta   Q    NaN
       X      4
       Y      5
       Z      6
dtype: float64
```

In [ ]:
```
# Next we'll learn more abot Pivoting DataFrames!
```