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

In [6]:
```
#Lets make some Series to work with

#First Series
ser1 = Series([2,np.nan,4,np.nan,6,np.nan],
           index=['Q','R','S','T','U','V'])

#Second Series (based off length of ser1)
ser2 = Series(np.arange(len(ser1), dtype=np.float64),
           index=['Q','R','S','T','U','V'])

ser2[-1] = np.nan
```

In [7]:
```
ser1
```

Out[7]:
```
Q    2
R  NaN
S    4
T  NaN
U    6
V  NaN
dtype: float64
```

In [8]:
```
ser2
```

Out[8]:
```
Q    0
R    1
S    2
T    3
U    4
V  NaN
dtype: float64
```

In [14]:
```
# Now let's get a series where the value of ser1 is chosen if ser2 is NAN,otherwi
Series(np.where(pd.isnull(ser1),ser2,ser1),index=ser1.index)
```

Out[14]:
```
Q    2
R    1
S    4
T    3
U    6
V  NaN
dtype: float64
```

In [11]:
```
#Take a moment to really understand how the above worked
```

In [21]:
```
#Now we can do the same thing simply by using combine_first with pandas
ser1.combine_first(ser2)

#This combines the Series values, choosing the values of the calling Series first
```

Out[21]:
```
Q     2
R     1
S     4
T     3
U     6
V   NaN
dtype: float64
```

In [22]:
```
#Now lets how this works on a DataFrame!
```

In [34]:
```
#Lets make some
dframe_odds = DataFrame({'X': [1., np.nan, 3., np.nan],
                         'Y': [np.nan, 5., np.nan, 7.],
                         'Z': [np.nan, 9., np.nan, 11.]})
dframe_evens = DataFrame({'X': [2., 4., np.nan, 6., 8.],
                         'Y': [np.nan, 10., 12., 14., 16.]})
```

In [35]:
```
#Show
dframe_odds
```

Out[35]:

	X	Y	Z
0	1	NaN	NaN
1	NaN	5	9
2	3	NaN	NaN
3	NaN	7	11

In [36]:
```
#Show
dframe_evens
```

Out[36]:

	X	Y
0	2	NaN
1	4	10
2	NaN	12
3	6	14
4	8	16

In [38]:
```
#Now lets combine using odds values first, unless theres a NAN, then put the even
dframe_odds.combine_first(dframe_evens)
```

Out[38]:

	X	Y	Z
0	1	NaN	NaN
1	4	5	9
2	3	12	NaN
3	6	7	11
4	8	16	NaN

In [ ]:
```
#Next up: Reshaping DataFrames!
```