```
In [12]:  import pandas as pd
          from pandas import Series, DataFrame
          import numpy as np
          #Now we'll learn how to merge on an index
```

```
In [4]:  # Lets get two dframes

         df_left = DataFrame({'key': ['X','Y','Z','X','Y'],
                              'data': range(5)})
         df_right = DataFrame({'group_data': [10, 20]}, index=['X', 'Y'])
```

```
In [5]:  #Show
         df_left
```

Out[5]:

	data	key
0	0	X
1	1	Y
2	2	Z
3	3	X
4	4	Y

```
In [7]:  #Show
         df_right
```

Out[7]:

	group_data
X	10
Y	20

```
In [8]:  #Now merge, we'll use the key for the left Dframe, and the index for the right
         pd.merge(df_left,df_right,left_on='key',right_index=True)
```

Out[8]:

	data	key	group_data
0	0	X	10
3	3	X	10
1	1	Y	20
4	4	Y	20

In [10]: ```python
# We can also get a union by using outer
pd.merge(df_left,df_right,left_on='key',right_index=True,how='outer')
```

Out[10]:

| | data | key | group_data |
|---|---|---|---|
| **0** | 0 | X | 10 |
| **3** | 3 | X | 10 |
| **1** | 1 | Y | 20 |
| **4** | 4 | Y | 20 |
| **2** | 2 | Z | NaN |

In [13]: ```python
#Now let's try something a little more complicated, remember hierarchal index?
df_left_hr = DataFrame({'key1': ['SF','SF','SF','LA','LA'],
                        'key2': [10, 20, 30, 20, 30],
                        'data_set': np.arange(5.)})
df_right_hr = DataFrame(np.arange(10).reshape((5, 2)),
                        index=[['LA','LA','SF','SF','SF'],
                               [20, 10, 10, 10, 20]],
                        columns=['col_1', 'col_2'])
```

In [14]: ```python
#SHOW
df_left_hr
```

Out[14]:

| | data_set | key1 | key2 |
|---|---|---|---|
| **0** | 0 | SF | 10 |
| **1** | 1 | SF | 20 |
| **2** | 2 | SF | 30 |
| **3** | 3 | LA | 20 |
| **4** | 4 | LA | 30 |

In [15]: ```python
#Show, this has a index hierarchy
df_right_hr
```

Out[15]:

		col_1	col_2
LA	**20**	0	1
	10	2	3
SF	**10**	4	5
	10	6	7
	20	8	9

```
In [16]:   # Now we can merge the left by using keys and the right by its index
           pd.merge(df_left_hr,df_right_hr,left_on=['key1','key2'],right_index=True)
```

Out[16]:

	data_set	key1	key2	col_1	col_2
0	0	SF	10	4	5
0	0	SF	10	6	7
1	1	SF	20	8	9
3	3	LA	20	0	1

```
In [17]:   # We can alo keep a union by choosing 'outer' method
           pd.merge(df_left_hr,df_right_hr,left_on=['key1','key2'],right_index=True,how='out
```

Out[17]:

	data_set	key1	key2	col_1	col_2
0	0	SF	10	4	5
0	0	SF	10	6	7
1	1	SF	20	8	9
2	2	SF	30	NaN	NaN
3	3	LA	20	0	1
4	4	LA	30	NaN	NaN
4	NaN	LA	10	2	3

```
In [23]:   # WE can also you .join()

           # Shown on our first two DataFrames
           df_left.join(df_right)
```

Out[23]:

	data	key	group_data
0	0	X	NaN
1	1	Y	NaN
2	2	Z	NaN
3	3	X	NaN
4	4	Y	NaN

```
In [ ]:   # Next we'll learn about the concatenate function!
```