

```
In [3]: import numpy as np
import pandas as pd
from pandas import DataFrame, Series
```

```
In [4]: #Let's make a dframe
dframe = DataFrame({'k1':['X','X','Y','Y','Z'],
                    'k2':['alpha','beta','alpha','beta','alpha'],
                    'dataset1':np.random.randn(5),
                    'dataset2':np.random.randn(5)})

#Show
dframe
```

```
Out[4]:
```

	dataset1	dataset2	k1	k2
0	-0.123544	1.924614	X	alpha
1	-1.448666	0.477115	X	beta
2	-1.139759	-1.378362	Y	alpha
3	-0.617664	-0.105714	Y	beta
4	-0.573748	0.409242	Z	alpha

```
In [5]: #Now let's see how to use groupby

#Lets grab the dataset1 column and group it by the k1 key
group1 = dframe['dataset1'].groupby(dframe['k1'])

#Show the groupby object
group1
```

```
Out[5]: <pandas.core.groupby.SeriesGroupBy object at 0x00000000AF84780>
```

```
In [6]: #Now we can perform operations on this particular group
group1.mean()
```

```
Out[6]: k1
X      -0.786105
Y      -0.878712
Z      -0.573748
Name: dataset1, dtype: float64
```

In [7]: *# We can use group keys that are series as well*

#For example:

#We'll make some arrays for use as keys

```
cities = np.array(['NY','LA','LA','NY','NY'])
```

```
month = np.array(['JAN','FEB','JAN','FEB','JAN'])
```

#Now using the data from dataset1, group the means by city and month

```
dframe['dataset1'].groupby([cities,month]).mean()
```

Out[7]:

LA	FEB	-1.448666
	JAN	-1.139759
NY	FEB	-0.617664
	JAN	-0.348646

Name: dataset1, dtype: float64

In [8]: *# Let's see the original dframe again.*
dframe

Out[8]:

	dataset1	dataset2	k1	k2
0	-0.123544	1.924614	X	alpha
1	-1.448666	0.477115	X	beta
2	-1.139759	-1.378362	Y	alpha
3	-0.617664	-0.105714	Y	beta
4	-0.573748	0.409242	Z	alpha

In [9]: *# WE can also pass column names as group keys*
dframe.groupby('k1').mean()

Out[9]:

	dataset1	dataset2
k1		
X	-0.786105	1.200865
Y	-0.878712	-0.742038
Z	-0.573748	0.409242

```
In [10]: # Or multiple column names
dframe.groupby(['k1','k2']).mean()
```

Out[10]:

		dataset1	dataset2
k1	k2		
X	alpha	-0.123544	1.924614
	beta	-1.448666	0.477115
Y	alpha	-1.139759	-1.378362
	beta	-0.617664	-0.105714
Z	alpha	-0.573748	0.409242

```
In [11]: # Another useful groupby method is getting the group sizes
dframe.groupby(['k1']).size()
```

Out[11]:

```
k1
X      2
Y      2
Z      1
dtype: int64
```

```
In [12]: # We can also iterate over groups

#For example:
for name,group in dframe.groupby('k1'):
    print "This is the %s group" %name
    print group
    print '\n'
```

```
This is the X group
  dataset1 dataset2 k1    k2
0 -0.123544  1.924614 X  alpha
1 -1.448666  0.477115 X   beta
```

```
This is the Y group
  dataset1 dataset2 k1    k2
2 -1.139759 -1.378362 Y  alpha
3 -0.617664 -0.105714 Y   beta
```

```
This is the Z group
  dataset1 dataset2 k1    k2
4 -0.573748  0.409242 Z  alpha
```

```
In [13]: # We can also iterate with multiple keys
for (k1,k2) , group in dframe.groupby(['k1','k2']):
    print "Key1 = %s Key2 = %s" %(k1,k2)
    print group
    print '\n'
```

```
Key1 = X Key2 = alpha
  dataset1  dataset2 k1    k2
0 -0.123544  1.924614  X   alpha
```

```
Key1 = X Key2 = beta
  dataset1  dataset2 k1    k2
1 -1.448666  0.477115  X   beta
```

```
Key1 = Y Key2 = alpha
  dataset1  dataset2 k1    k2
2 -1.139759 -1.378362  Y   alpha
```

```
Key1 = Y Key2 = beta
  dataset1  dataset2 k1    k2
3 -0.617664 -0.105714  Y   beta
```

```
Key1 = Z Key2 = alpha
  dataset1  dataset2 k1    k2
4 -0.573748  0.409242  Z   alpha
```

```
In [14]: # A possibly useful tactic is creating a dictionary of the data pieces
group_dict = dict(list(dframe.groupby('k1')))

#Show the group with X
group_dict['X']
```

Out[14]:

	dataset1	dataset2	k1	k2
0	-0.123544	1.924614	X	alpha
1	-1.448666	0.477115	X	beta

```
In [15]: # We could have also chosen to do this with axis = 1

# Let's creat a dictionary for dtypes of objects!
group_dict_axis1 = dict(list(dframe.groupby(dframe.dtypes,axis=1)))

#show
group_dict_axis1
```

```
Out[15]: {dtype('float64'):  dataset1  dataset2
0 -0.123544  1.924614
1 -1.448666  0.477115
2 -1.139759 -1.378362
3 -0.617664 -0.105714
4 -0.573748  0.409242, dtype('O'):  k1      k2
0 X  alpha
1 X  beta
2 Y  alpha
3 Y  beta
4 Z  alpha}
```

```
In [16]: # Next we'll learn how to use groupby with columns
```

```
In [18]: # For example if we only wanted to group the dataset2 column with both sets of keys
dataset2_group = dframe.groupby(['k1','k2'])[['dataset2']]

dataset2_group.mean()
```

```
Out[18]:
```

		dataset2
k1	k2	
X	alpha	1.924614
	beta	0.477115
Y	alpha	-1.378362
	beta	-0.105714
Z	alpha	0.409242

```
In [19]: #Next we'll have a quick lesson on grouping with dictionaries and series!
```

```
Out[19]: <pandas.core.groupby.DataFrameGroupBy object at 0x000000000AFE908>
```

```
In [ ]:
```