More notes about Pandas Series

In [ ]:
```
import random as random
import pandas as pd
```

quick note about using the index labels

In [ ]:
```
# made some data and some labels
x = [1,6,3,7]
labels = ['D1', 'D2', 'D3', 'D4']

# the first time let pandas assign the labels (0...N-1)
s = pd.Series(x)
```

In [ ]:
```
# now you can index into the labels like this
# it looks just like you're indexing based on position
# but really you're indexing based on label!
# (the labels just happen to be numbers)
s[0:3]
```

In [ ]:
```
# now do it with the labels specified in "1"
x = [1,6,3,7]
labels = ['D1', 'D2', 'D3', 'D4']

s = pd.Series(x, index = labels)
```

In [ ]:
```
# same idea (start, stop, step), but use the index labels!
s['D1':'D5':2]
```

now lets do a more complex example to make a few more points...

In [ ]:
```
# set the 'pseudo' random number generator
random.seed(2)

N = 10

data1=[]
data2=[]

# fill up arrays
for i in range(0,N):
  data1.append(random.randint(0,10))
  data2.append(random.randint(2,12))

# make some labels
labels = []
for i in range(0,N):
  labels.append('Samp' + str(i))
```

make two series

- lets shuffle the labels before making each series - so they'll have all the same labels but in a different order
- we're doing this to illustrate the "shuffle" method, but also to make a point about how series keep track of data

In [ ]:
```
# shuffle labels, make the series
random.shuffle(labels)
s1 = pd.Series(data1, index = labels)

# shuffle labels, make the series
random.shuffle(labels)
s2 = pd.Series(data2, index = labels)
```

In [ ]:
```
s1
```

In [ ]:
```
# do simple operations
s1+s2
# s1['Samp0']+s2['Samp0']
```

In [ ]:
```
# use booleans to index into series
s1==9
```

In [ ]:
```
ind = (s1==9)
ind = ((s1==9) | (s1==8))
s1[ind]
```

test for an item "in" the series

- tricky, cause it will default to operating on the index (labels)
- but can directly access the values

In [ ]:
```
9 in s1      #no - will operate on labels
# 9 in s1.values  # yes! operates on the values
```

make a new series from an old one

- or reassign to modify the current series
- handy if you want to filter the data

In [ ]:
```
s3 = s1[s1!=9]

#s3 = s1[s1>=4]
s3
# can even reassign to itself to create a modified series
# s1 = s1[s1!=3]
```