

```
import pandas as pd
import numpy as np
```

dataframes

```
from numpy.random import randn
np.random.seed(101)
```

```
df = pd.DataFrame(randn(5,4),index='A B C D E'.split(),columns='W X Y Z'.split())
print(df)
```

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

```
DF = pd.DataFrame(randn(4,3),index='A B C D'.split(),columns='a b c'.split())
print(DF)
```

	a	b	c
A	0.302665	1.693723	-1.706086
B	-1.159119	-0.134841	0.390528
C	0.166905	0.184502	0.807706
D	0.072960	0.638787	0.329646

```
print(df['W'])
```

A	2.706850
B	0.651118
C	-2.018168
D	0.188695
E	0.190794

Name: W, dtype: float64

```
print(df['Z'])
```

A	0.503826
B	0.605965
C	-0.589001
D	0.955057
E	0.683509

Name: Z, dtype: float64

```
print(df[['W', 'Z', 'X']])
```

	W	Z	X
A	2.706850	0.503826	0.628133
B	0.651118	0.605965	-0.319318
C	-2.018168	-0.589001	0.740122
D	0.188695	0.955057	-0.758872
E	0.190794	0.683509	1.978757

```
print(df.W)
```

A	2.706850
B	0.651118
C	-2.018168
D	0.188695
E	0.190794

Name: W, dtype: float64

```
print(type(df['W']))
```

```
df['new'] = df['W'] + df['Y']
```

```
<class 'pandas.core.series.Series'>
```

```
print(df)
```

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

dropping col

```
df.drop('new',axis=1,inplace=True)
```

```
print(df)
```

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

dropping row

```
df.drop('E',axis=0,inplace=True)
```

```
print(df)
```

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

selecting row

```
print(df.loc['A'])
```

W	2.706850
X	0.628133
Y	0.907969
Z	0.503826

Name: A, dtype: float64

selecting through index

```
print(df.iloc[2])
```

W	-2.018168
X	0.740122
Y	0.528813
Z	-0.589001

Name: C, dtype: float64

```
print(df.loc['B','Y'])
```

-0.8480769834036315

```
print(df.loc[['A','B'],['W','Y']])
```

	W	Y
A	2.706850	0.907969
B	0.651118	-0.848077

conditional selectiona

```
print(df)
print(df>0)
```

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

	W	X	Y	Z
A	True	True	True	True
B	True	False	False	True
C	False	True	True	False
D	True	False	False	True
E	True	True	True	True

```
print(df[df["W"]>0])
```

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

```
print(df[df['W']>0]['Y'])
```

```
A    0.907969
B   -0.848077
D   -0.933237
E    2.605967
Name: Y, dtype: float64
```

```
print(df[df['W']>0][['Y','X']])
```

	Y	X
A	0.907969	0.628133
B	-0.848077	-0.319318
D	-0.933237	-0.758872
E	2.605967	1.978757

or and and

```
print(df[(df['W']>0) & (df['Y']>1)])
```

	W	X	Y	Z
E	0.190794	1.978757	2.605967	0.683509

```
print(df)
```

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

```
print(df.reset_index())
```

	index	W	X	Y	Z
0	A	2.706850	0.628133	0.907969	0.503826
1	B	0.651118	-0.319318	-0.848077	0.605965
2	C	-2.018168	0.740122	0.528813	-0.589001
3	D	0.188695	-0.758872	-0.933237	0.955057
4	E	0.190794	1.978757	2.605967	0.683509

```
newind = 'CA NY WY OR CO'.split()
```

```
df['States'] = newind
```

```
print(df)
```

	W	X	Y	Z	States
A	2.706850	0.628133	0.907969	0.503826	CA
B	0.651118	-0.319318	-0.848077	0.605965	NY
C	-2.018168	0.740122	0.528813	-0.589001	WY
D	0.188695	-0.758872	-0.933237	0.955057	OR
E	0.190794	1.978757	2.605967	0.683509	CO

```
print(df.set_index('States'))
```

	W	X	Y	Z
States				
CA	2.706850	0.628133	0.907969	0.503826
NY	0.651118	-0.319318	-0.848077	0.605965
WY	-2.018168	0.740122	0.528813	-0.589001
OR	0.188695	-0.758872	-0.933237	0.955057
CO	0.190794	1.978757	2.605967	0.683509

```
print(df)
```

	W	X	Y	Z	States
A	2.706850	0.628133	0.907969	0.503826	CA
B	0.651118	-0.319318	-0.848077	0.605965	NY
C	-2.018168	0.740122	0.528813	-0.589001	WY

```

D  0.188695 -0.758872 -0.933237  0.955057  OR
E  0.190794  1.978757  2.605967  0.683509  CO

```

```
df.set_index('States',inplace=True)
```

```
print(df)
```

```

           W           X           Y           Z
States
CA      2.706850  0.628133  0.907969  0.503826
NY      0.651118 -0.319318 -0.848077  0.605965
WY     -2.018168  0.740122  0.528813 -0.589001
OR      0.188695 -0.758872 -0.933237  0.955057
CO      0.190794  1.978757  2.605967  0.683509

```

multi index and index hierarchy

index lbl

```

outside = ['G1','G1','G1','G2','G2','G2']
inside = [1,2,3,1,2,3]
hier_index = list(zip(outside,inside))
hier_index = pd.MultiIndex.from_tuples(hier_index)

```

```
print(hier_index)
```

```

MultiIndex([( 'G1', 1),
            ( 'G1', 2),
            ( 'G1', 3),
            ( 'G2', 1),
            ( 'G2', 2),
            ( 'G2', 3)],
           )

```

```

df = pd.DataFrame(np.random.randn(6,2),index=hier_index,columns=['A','B'])
print(df)

```

```

           A           B
G1 1 -0.497104 -0.754070
   2 -0.943406  0.484752
   3 -0.116773  1.901755
G2 1  0.238127  1.996652
   2 -0.993263  0.196800
   3 -1.136645  0.000366

```

```
print(df.loc['G1'].loc[1])
print(df.index.names)
```

```
A    -0.497104
B    -0.754070
Name: 1, dtype: float64
[None, None]
```

```
df.index.names = ['Group','Num']
print(df)
```

		A	B
Group	Num		
G1	1	-0.497104	-0.754070
	2	-0.943406	0.484752
	3	-0.116773	1.901755
G2	1	0.238127	1.996652
	2	-0.993263	0.196800
	3	-1.136645	0.000366

```
print(df.xs('G1'))
print(df.xs(('G1',1)))
```

	A	B
Num		
1	-0.497104	-0.754070
2	-0.943406	0.484752
3	-0.116773	1.901755
A	-0.497104	
B	-0.754070	

```
Name: (G1, 1), dtype: float64
```

✓ 0s completed at 22:44

