```
import pandas as pd
```

Series

```
s = pd.Series([3, -5, 7, 4],  index=['a',  'b',  'c',  'd'])
```

DataFrame

```
data = {'Country': ['Belgium',  'India',  'Brazil'],

'Capital': ['Brussels',  'New Delhi',  'Brasilia'],

'Population': [11190846, 1303171035, 207847528]}

df = pd.DataFrame(data,columns=['Country',  'Capital',  'Population'])
```

Asking For Help

+ Code + Text

```
help(pd.Series.loc)
```

I/O

Read and Write to CSV

```
pd.read_csv('file.csv', header=None, nrows=5)
df.to_csv('myDataFrame.csv')
```

Read multiple sheets from the same file

```
xlsx = pd.ExcelFile('file.xls')
df = pd.read_excel(xlsx,  'Sheet1')
```

Read and Write to Excel

```
pd.read_excel('file.xlsx')
df.to_excel('dir/myDataFrame.xlsx',  sheet_name='Sheet1')
```

Read and Write to SQL Query or Database Table

```
from sqlalchemy import create_engine
engine = create_engine('sqlite:///:memory:')
pd.read_sql(SELECT * FROM my_table;, engine)
pd.read_sql_table('my_table', engine)
pd.read_sql_query(SELECT * FROM my_table;', engine)
df.to_sql('myDf', engine)
```

```
       File "<ipython-input-14-ecd41690a416>", line 3
         pd.read_sql(SELECT * FROM my_table;, engine)
                                           ^
     SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

Selection

Getting

```
s['b']
-5
```

Get subset of a DataFrame

```
df[1:]
Country     Capital   Population
1  India    New Delhi 1303171035
2  Brazil   Brasilia  207847528
```

Selecting', Boolean Indexing and Setting

By Position

```
df.iloc([0], [0])
'Belgium'
df.iat([0], [0])
'Belgium'
```

By Label

```
df.loc([0],  ['Country'])
'Belgium'
```

```
df.at([0],  ['Country'])
'Belgium'
```

By Label/Position

```
df.ix[2]
Country      Brazil
Capital    Brasilia
Population  207847528
```

```
df.ix[:, 'Capital']
0     Brussels
1    New Delhi
2     Brasilia
```

```
df.ix[1, 'Capital']
'New Delhi'
```

Boolean Indexing

```
s[~(s > 1)]
```

```
s[(s < -1) | (s > 2)]
```

```
df[df['Population']>1200000000]
```

Setting

```
s['a'] = 6
```

Dropping

```
s.drop(['a',  'c'])
```

```
df.drop('Country', axis=1)
```

Sort and Rank

```
df.sort_index()
```

```
df.sort_values(by='Country')
```

```
df.rank()
```

Retrieving Series/DataFrame Information

Basic Information

```
df.shape
```

```
df.index
```

```
df.columns
```

```
df.info()
```

```
df.count()
```

Summary

```
df.sum()
```

```
df.cumsum()
```

```
df.min()/df.max()
```

```
df.idxmin()/df.idxmax()
```

```
df.describe()
```

```
df.mean()
```

```
df.median()
```

Applying Functions

```
f = lambda x: x*2
```

```
df.apply(f)
```

```
df.applynap(f)
```

Internal Data Alignment

```
s3 = pd.Series([7, -2, 3],  index=['a',  'c',  'd'])
s + s3
a     10.0
b     NaN
c     5.0
d     7.0
```

Arithmetic Operations with Fill Methods

```
s.add(s3, fill_value=0)
a    10.0
b    -5.0
c    5.0
d    7.0
s.sub(s3, fill_value=2)
s.div(s3, fill_value=4)
s.mul(s3, fill_value=3)
```

0s completed at 12:56 AM