

Pandas

Pandas

Series

0	maths	85
1	science	72

index

location

DataFrame

		columns		
	index	student1	student2	student3
0	maths	85	62	45
1	science	72	70	48
2	english	NaN	55	70

location

Series

```
test_list = [100,200,300]
```

```
pd.Series(data=test_list)
```

```
dictionary = {'a':100,'b':200,'c':300}
```

```
pd.Series(data=dictionary)
```

Series

```
In [6]: test_list = [100,200,300]
pd.Series(data=test_list)
```

```
Out[6]: 0    100
        1    200
        2    300
        dtype: int64
```

```
In [5]: dictionary = {'a':100,'b':200,'c':300}
pd.Series(data=dictionary)
```

```
Out[5]: a    100
        b    200
        c    300
        dtype: int64
```

Dataframe

```
data = [['thomas', 100], ['nicholas', 200], ['danson', 300]]  
df = pd.DataFrame(data, columns = ['Name', 'Age'])
```

```
data = {'Name':['thomas', 'nicholas', 'danson', 'jack'], 'Age':  
[100, 200, 300, 400]}  
df = pd.DataFrame(data)
```

Dataframe

```
In [6]: data = [['thomas', 100], ['nicholas', 200], ['danson', 300]]  
df = pd.DataFrame(data, columns = ['Name', 'Age'])  
df
```

Out[6]:

	Name	Age
0	thomas	100
1	nicholas	200
2	danson	300

```
In [7]: data = {'Name':['thomas', 'nicholas', 'danson', 'jack'], 'Age':[100, 200, 300, 400]}  
df = pd.DataFrame(data)  
df
```

Out[7]:

	Name	Age
0	thomas	100
1	nicholas	200
2	danson	300
3	jack	400

IO tools (text, CSV, HDF5, ...)

The pandas I/O API is a set of top level `reader` functions accessed like `pandas.read_csv()` that generally return a pandas object. The corresponding `writer` functions are object methods that are accessed like `DataFrame.to_csv()`. Below is a table containing available `readers` and `writers`.

Format Type	Data Description	Reader	Writer
text	CSV	<code>read_csv</code>	<code>to_csv</code>
text	JSON	<code>read_json</code>	<code>to_json</code>
text	HTML	<code>read_html</code>	<code>to_html</code>
text	Local clipboard	<code>read_clipboard</code>	<code>to_clipboard</code>
binary	MS Excel	<code>read_excel</code>	<code>to_excel</code>
binary	OpenDocument	<code>read_excel</code>	
binary	HDF5 Format	<code>read_hdf</code>	<code>to_hdf</code>
binary	Feather Format	<code>read_feather</code>	<code>to_feather</code>
binary	Parquet Format	<code>read_parquet</code>	<code>to_parquet</code>
binary	Msgpack	<code>read_msgpack</code>	<code>to_msgpack</code>
binary	Stata	<code>read_stata</code>	<code>to_stata</code>
binary	SAS	<code>read_sas</code>	
binary	Python Pickle Format	<code>read_pickle</code>	<code>to_pickle</code>
SQL	SQL	<code>read_sql</code>	<code>to_sql</code>
SQL	Google Big Query	<code>read_gbq</code>	<code>to_gbq</code>

read_csv

```
In [4]: df= pd.read_csv('/Users/nicholas/Downloads/titanic/train.csv')
df
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

read_excel

```
pd.read_excel('file.xlsx')
```

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

- Read multiple sheets from the same file

```
xlsx = pd.ExcelFile('file.xls')
```

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

Basic Information

- `df.shape`
- `df.index`
- `df.columns`
- `df.info()`
- `df.count()`
- `(rows, columns)`
- Describe index
- Describe DataFrame columns
- Info on DataFrame
- Number of non-NA values

Basic stats

- `df.sum()`
- `df.cumsum()`
- `df.min()/df.max()`
- `df.idxmin()/df.idxmax()`
- `df.describe()`
- `df.mean()`
- `df.median()`
- Sum of values
- Cumulative sum of values
- Minimum/maximum values
- Minimum/Maximum index value
- Summary statistics
- Mean of values
- Median of values

Basic plotting

- `data['col'].plot()`
- `data['col'].hist()`

