

Sample Data

```
>>> s = df['city']
```

```
>>> df
```

	company	city	state	zip
a	Widgets, Inc.	Jay City	MO	46239
b	ABC, LLC	Wanton	TX	94562
c	ACME	Bland	AL	32329

Pandas Indexing Cheatsheet

<https://wrighters.io>

by Matt Wright

Array Indexing Operator ([])

DataFrame

```
df['company']
```

Series, entire column

```
df[['company']]
```

DataFrame, with one column

```
df[['company', 'city']]
```

DataFrame, multiple columns

Series

```
s[0]
```

scalar, 'Jay City', selecting by location

```
s['a']
```

scalar, 'Jay City', selecting by index label

loc Indexer - selecting by index label

DataFrame

```
df.loc['a']
```

Series, first row matching index label

```
df.loc[['a', 'c']]
```

DataFrame, selecting by multiple labels

```
df.loc['a', 'city']
```

scalar, 'Jay City', selecting by row and column label

```
df.loc[['a', 'c'], 'city']
```

Series, selecting by labels and column

```
df.loc[['a', 'c'], ['city', 'zip']]
```

DataFrame, selecting by labels and columns

Series

```
s.loc['a']
```

scalar, 'Jay City', selecting by index label

```
s.loc[['a', 'c']]
```

Series, selecting by multiple labels

```
s.loc[['a']]
```

Series, selecting by a single label

iloc Indexer - selecting by row and column offset

DataFrame

```
df.iloc[0]
```

Series, first row by row location

```
df.iloc[[0,2]]
```

DataFrame, selecting by multiple rows

```
df.iloc[0, 1]
```

scalar, 'Jay City', selecting by row and column location

```
df.iloc[[0,2], 1]
```

Series, selecting by rows and a single column

```
df.iloc[[0,2], [1,3]]
```

DataFrame, selecting by rows and columns

Series

```
s.iloc[0]
```

scalar, 'Jay City', selecting by offset

```
s.iloc[-1]
```

scalar, 'Bland', selecting by relative offset

```
s.iloc[[0,2]]
```

Series, selecting by multiple offsets

```
s.iloc[[0]]
```

Series, selecting by a single offset

Slicing

DataFrame

```
df[0:2]
```

slice by row location (selects rows a, b only)

```
df.loc['a':'c']
```

slice by label (selects rows a, b, **and** c)

```
df.iloc[0:2]
```

slice by row location (selects rows, b,c only)

```
df.loc['a':'c', 'city':'zip']
```

slice both rows and columns (inclusive for both)

```
df.iloc[0:3, 1:4]
```

returns same as above

Series

```
s[:]
```

full slice

```
s[:2], s[0:2], s[0:2:1],
```

all ways to slice first two elements in Series

```
s[slice(0,2,1)]
```

```
s['b':'c']
```

slice by label, both labels inclusive

```
s.loc['b':'c']
```

same as above

```
s.iloc[1:3]
```

slice by location (non inclusive, returns labels b,c like above)