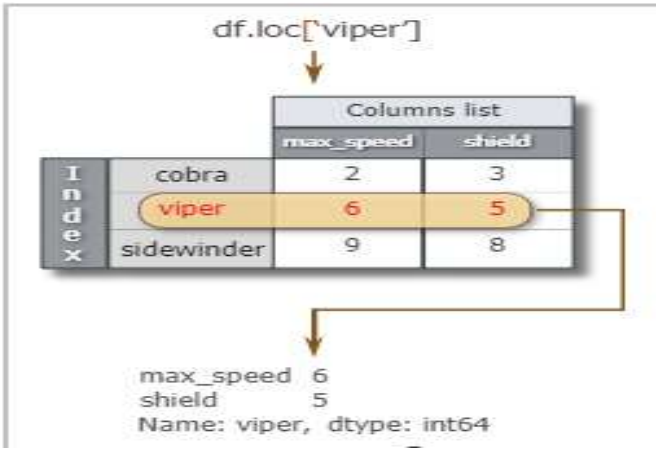
Pandas Series: loc() function

The loc() function is used to access a group of rows and columns by label(s) or a boolean array.

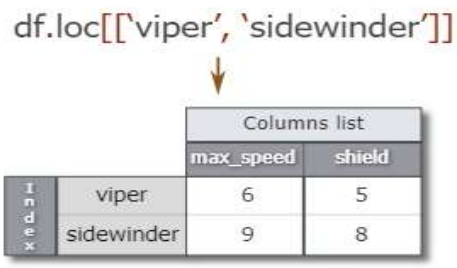
.loc[] is primarily label based, but may also be used with a boolean array.

SYNTAX: Series.loc




```
import numpy as np
import pandas as pd
df = pd.DataFrame([[2, 3], [6, 5], [9, 8]],
     index=['cobra', 'viper', 'sidewinder'],
     columns=['max_speed', 'shield'])
PRINT(df)
>>>df.loc['viper']
output
max_speed    6
shield       5
Name: viper, dtype: int64
>>>df.loc[['viper', 'sidewinder']]
```

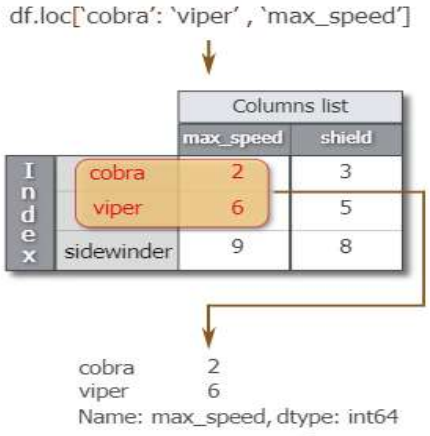
	max_speed	shield
viper	6	5
sidewinder	9	8




```
>>>df.loc['cobra', 'shield']
```

3


```
>>>df.loc['cobra':'viper', 'max_speed']
cobra    2
viper    6
Name: max_speed, dtype: int64
```



```
>>>df.loc[df['shield'] > 6]
```

```
In [8]: df.loc[df['shield'] > 6]
```

Out[8]:

	max_speed	shield
sidewinder	9	8

Conditional that returns a boolean Series with column labels specified

```
In [9]: df.loc[df['shield'] > 6, ['max_speed']]
```

Out[9]:

	max_speed
sidewinder	9

Callable that returns a boolean Series

```
In [10]: df.loc[lambda df: df['shield'] == 8]
```

Out[10]:

	max_speed	shield
sidewinder	9	8

Setting values

Set value for all items matching the list of labels

```
In [11]: df.loc[['viper', 'sidewinder'], ['shield']] = 50
         df
```

Out[11]:

	max_speed	shield
cobra	2	3
viper	6	50
sidewinder	9	50

Set value for an entire row

```
In [12]: df.loc['cobra'] = 10
         df
```

Out[12]:

	max_speed	shield
cobra	10	10
viper	6	50
sidewinder	9	50

Set value for an entire column

```
In [13]: df.loc[:, 'max_speed'] = 40
         df
```

Out[13]:

	max_speed	shield
cobra	40	10
viper	40	50
sidewinder	40	50

Set value for rows matching callable condition

```
In [14]: df.loc[df['shield'] > 25] = 0
         df
```

Out[14]:

	max_speed	shield
cobra	40	10
viper	0	0
sidewinder	0	0

```
>>>df = pd.DataFrame([[2, 3], [6, 5], [9, 8]],
     index=[3, 4, 5], columns=['max_speed', 'shield'])
PRINT(df)

>>>df.loc[3:5]
```

	max_speed	shield
3	2	3
4	6	5
5	9	8

Getting values with a MultiIndex

A number of examples using a DataFrame with a MultiIndex

Slice with integer labels for rows. As mentioned above, note that both the start and stop of the slice are included.

```
>>>df.loc[3:5]
```

The iloc() function

Access a group of rows and columns in Pandas

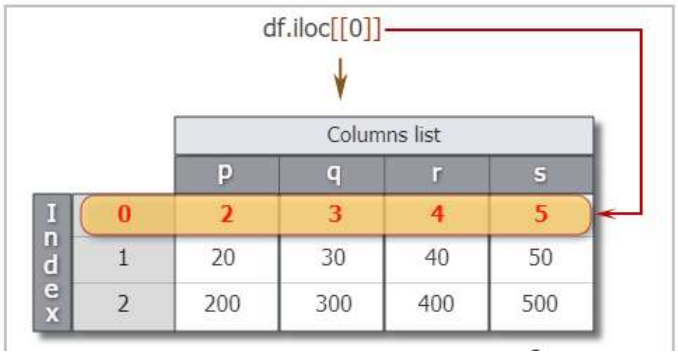
The iloc() function is used to access a group of rows and columns by label(s) or a boolean array.

.iloc[] is primarily integer position based (from 0 to length-1 of the axis), but may also be used with a boolean array.

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

```
mydict = [{'p': 2, 'q': 3, 'r': 4, 's': 5},
          {'p': 20, 'q': 30, 'r': 40, 's': 50},
          {'p': 200, 'q': 300, 'r': 400, 's': 500 }]
```

```
df = pd.DataFrame(mydict)
print(df)
```



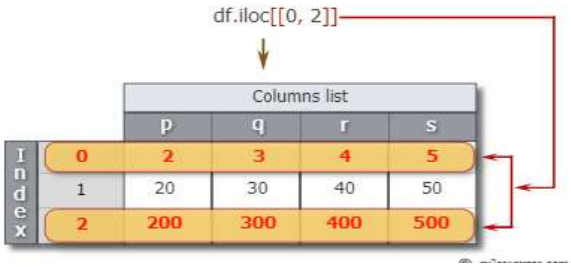

```
>>>type(df.iloc[0])
```

```
pandas.core.series.Series
>>>df.iloc[0]
```

```
p    2
q    3
r    4
s    5
Name: 0, dtype: int64
```

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

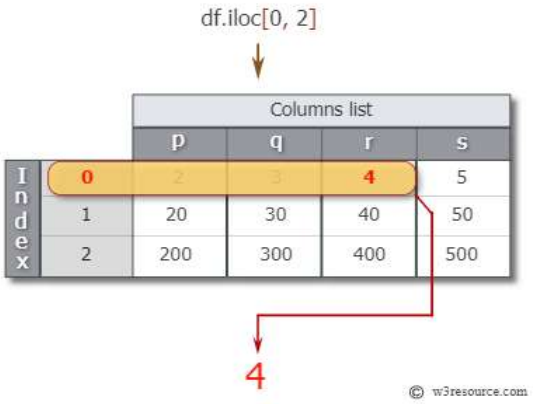
	p	q	r	s
0	2	3	4	5
2	200	300	400	500




```
>>>df.iloc[:3]
```

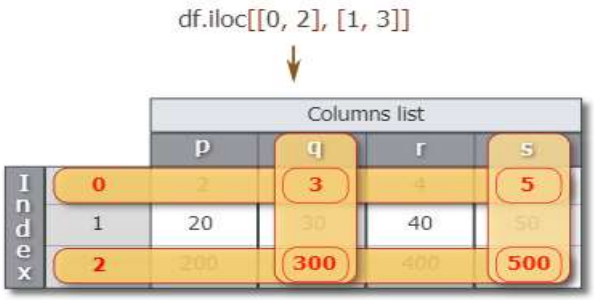
	p	q	r	s
0	2	3	4	5
1	20	30	40	50
2	200	300	400	500

```
>>>df.iloc[0, 2]
    4
```



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

	q	s
0	3	5
2	300	500




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

	p	q	r
1	20	30	40
2	200	300	400

```
>>>df.iloc[:, lambda df: [0, 2]]
```

	p	r
0	2	4
1	20	40
2	200	400