

Pandas Series

- **Type Conversion - .astype()**

for e.g.

Code:

```
import numpy as np
import pandas as pd

num_array = pd.Series(range(5))

num_array
```

Output:

```
0    0
1    1
2    2
3    3
4    4
dtype: int64
```

Code:

```
num_array.astype("float")
```

Output:

```
0    0.0
1    1.0
2    2.0
3    3.0
4    4.0
dtype: float64
```

Code:

```
num_array.astype("object")
```

Output:

```
0    0
1    1
2    2
3    3
4    4
dtype: object
```

Code:

```
num_array.astype("string")
```

Output:

```
0    0
1    1
2    2
3    3
4    4
dtype: string
```

Code:

```
num_array = pd.Series(["a","b","c"])
num_array
```

Output:

```
0    a
1    b
2    c
dtype: object
```

Note: We cannot convert string to integer, if the value provided is not numeric

Code:

```
num_array.astype("int")
```

Output:

```
ValueError: invalid literal for int() with base 10: 'a'
```

- Pandas index series and custom indices

Code:

```
import numpy as np
import pandas as pd
my_series = pd.Series(range(5))

my_series
```

```
>>
0    0
1    1
2    2
3    3
4    4
dtype: int64
```

```
my_series[3]
```

```
>> 3
```

```
my_series[0]
```

```
>> 0
```

```
my_series[1:3]    # 1 to 2
```

```
>>
1    1
2    2
dtype: int64
```

```
my_series[1::2]    # step size of 2
```

```
>>
1    1
3    3
dtype: int64
```

```
my_series = pd.Series(range(5), index=["Day 0","Day 1","Day 2","Day 3","Day 4"])
my_series
```

```
>>
```

```
Day 0    0
Day 1    1
Day 2    2
Day 3    3
Day 4    4
dtype: int64
```

```
my_series["Day 1"]
```

```
>> 1
```

```
my_series["Day 3"]
```

```
>> 3
```

```
my_series["Day 1":"Day 3"]
```

```
>>
```

```
Day 1    1
Day 2    2
Day 3    3
dtype: int64
```

Note: Here stop point is inclusive, because we have index in a non-numerical format.

```
my_series[::2] # Step size of two
```

```
>>
```

```
Day 0    0
Day 2    2
Day 4    4
dtype: int64
```

- **iloc method**

```
my_series.iloc[1]
```

```
>> 1
```

```
my_series.iloc[3]
```

```
>> 3
```

```
my_series.iloc[-1]
```

```
>> 4
```

```
my_series.iloc[[1,3,4]] # multiple row selection
```

```
>>
```

```
Day 1    1
Day 3    3
Day 4    4
dtype: int64
```

```
my_series.iloc[1:4]
```

```
>>
```

```
Day 1    1
Day 2    2
Day 3    3
dtype: int64
```

```
my_series.iloc[1:]
```

```
>>
```

```
Day 1    1
Day 2    2
Day 3    3
Day 4    4
dtype: int64
```

- **loc Method**

```
import pandas as pd
import numpy as np
my_series = pd.Series(range(5), index=["Day 0","Day 1","Day 2","Day 3","Day 4"])
my_series
```

```
>>
```

```
Day 0    0
Day 1    1
Day 2    2
Day 3    3
Day 4    4
dtype: int64
```

```
my_series.loc["Day 3"]
```

```
>> 3
```

```
my_series.loc["Day 1":"Day 3"]
```

```
>>
```

```
Day 1    1
Day 2    2
Day 3    3
dtype: int64
```

```
my_series.index = [0,2,44,100,5]
```

```
my_series
```

```
>>
```

```
0      0
2      1
44     2
100    3
5      4
```

```
dtype: int64
```

```
my_series.loc[5]
```

```
>> 4
```

```
my_series.loc[[2,44,5]] # multiple row
```

```
>>
```

```
2      1
44     2
5      4
dtype: int64
```

```
my_series.loc[0:5]
```

```
>>
```

```
0      0
2      1
44     2
100    3
5      4
dtype: int64
```

```
my_series.reset_index(drop=True) # resets the index to default type
```

```
>>
```

```
0      0
1      1
2      2
3      3
4      4
dtype: int64
```

```
my_series.reset_index(drop=True)
```

```
0      0
2      1
44     2
100    3
5      4
dtype: int64
```



```
0      0
1      1
2      2
3      3
4      4
dtype: int64
```