

MAIN FEATURES

1. One-dimensional labeled array capable of holding homogenous data
2. The data labels called indexes start from 0 by default.
3. The data in series is mutable but the size of series is immutable.

PANDAS SERIES

0	Rajan	Data
1	Rajat	
2	Amit	
3	Sunita	
4	Lekha	

Made by: Vineeta Garg

SERIES CREATION

SYNTAX

```
pandas.Series(data, index, name)
```

data

Using Array

EXAMPLE

```
import pandas as pd
import numpy as np
data = np.array(['W','E','T','e','o','m','e'])
series1 = pd.Series(data)
print(series1)
```

Using Scalar Values

EXAMPLE

```
import pandas as pd
# giving a scalar value with index
ser = pd.Series(10, index=[0, 1, 2, 3, 4, 5])
print(ser)
```

Using Lists

EXAMPLE

```
import pandas as pd
list = ['W','e','T','e','o','m','e']
series2 = pd.Series(list)
print(series2)
```

Using Mathematical Expressions

EXAMPLE

```
import pandas as pd
import numpy as np
num = np.arange(10,60,10)
s1 = pd.Series(index=num, data=num**2)
print(s1)
```

Using Dictionary

EXAMPLE

```
import pandas as pd
dict = {'Ankit': 1, 'Ranil': 2, 'Geetika': 3}
# create series from dictionary
ser = pd.Series(dict)
print(ser)
```

index: unique and hashable with same length as data. Default is np.arange(n) if no index is passed. Used for adding customized indexes.

EXAMPLE

```
import pandas as pd
import numpy as np
data = np.arange(5400,2500,7634,8725)
# providing an index
ser = pd.Series(data, index=['North', 'East', 'South', 'West'])
print(ser)
```

name: allows you to give a name to a Series object

EXAMPLE

```
import pandas as pd
import numpy as np
data = np.array(['W','e','T','e','o','m','e'])
series1 = pd.Series(data, name='SERIES-1')
print(series1)
```

SERIES OPERATIONS

- The add() function or + sign is used to perform addition operation on series.
- The sub()/subtract() function or - sign is used to perform subtraction operation on series.
- The mul()/multiply() function or * sign is used to perform multiplication operation on series.
- The div()/divide() function or / sign is used to perform division operation on series.

SERIES FUNCTIONS

- sum(): To add all values present in a series
- prod(): To multiply all values present in a series
- mean(): To find the mean of all values present in a series
- max(): To find the maximum of all values present in a series
- count(): To count the number of values, present in a series
- min(): To find the minimum of all values present in a series
- isnull(): checks for missing data on Series object
- sort_values(): To return the sorted series (ascending=True/False)

ACCESS SERIES ELEMENTS

INDEXING

- Positional indexes are used to extract a data element present at a particular index location from a series.
- Label indexes are used to extract a data element present at a particular index label from a series.
- Boolean indexing is used to filter data by applying certain condition on data using relational operators like ==, >, <, <=, >= and logical operators like ~(not), &(and) and |(or).

SLICING

Slicing is used to extract a subset of a series. You can specify beginning parameter (beg) and end parameter (end) to indicate the size of the slice to be extracted from the series.

HEAD() & TAIL()

- The head function is used to return a specified number of rows from the beginning of a Series. If no parameter is passed it returns top 5 rows.
- The tail function is used to return a specified number of rows from the end of a Series. If no parameter is passed it returns top 5 rows.