Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index.

The basic method to create a series object is:

S = pd.Series(data, index=index)

Here, data can be any of the follwing:

- a Python dict
- an ndarray
- a scalar value (like 2)

The **index** is a list of axis labels.

Working with Series:

Creating an empty Series:

```
import pandas as pd

import numpy as np

s = pd.Series()
print(s)
```

o/p: Series([], dtype: float64)

Creating a Series from ndarray object:

1. **Creating a series without index**

```
data = np.array(['Apple', 'Mango', 'Orange'])

s = pd.Series(data)

print(s)
```

o/p:

```
0     Apple

1     Mango

2    Orange

dtype: object
```

Note: when you don’t pass an index, by default it assigns the index ranging from 0 to len(data)-1

2. **Creating a series with index**

```
data1 = np.array(['Aditya', 'Vamshi', 'Kranthi'])

s1 = pd.Series(data1, index=[101, 102, 103])

print(s1)
```

o/p:

```
101     Aditya
102     Vamshi
103    Kranthi
dtype: object
```

Creating a series from a dictionary object:

Without index:

```
data2 = {'a':0, 'b':1, 'c':2, 'd':3, 'e':4}

s2 = pd.Series(data2)

print(s2)
```

o/p:

```
a    0
b    1
c    2
d    3
e    4
dtype: int64
```

Note:

When a **dict** object is passed as input and if no index is specified, then the dictionary keys are taken in a sorted order to construct index. If **index** is passed, the values in data corresponding to the labels in the index will be pulled out.

with index:

```
data3 = {'a':0, 'b':1, 'c':2, 'd':3, 'e':4}

s3 = pd.Series(data3, index=['x', 'y', 'z', 'a', 'b'])

print(s3)
```

o/p:

```
x    NaN
y    NaN
z    NaN
a    0.0
b    1.0
dtype: float64
```

Note:

Pandas fills the missing elements with NaN, also called Not any Number.

Creating a series from a scalar value/object:

If the data is a scalar value, an index must be provided. The value will be repeated to match the length of index.

```
s = pd.Series(2, index=[0,1,2,3,4])

print(s)
```

o/p:

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

Accessing the data from a Series object:

Data in a series can be accessed similar to that in an ndarray object.

```
s2 = pd.Series([1,2,3,4,5], index=['a','b','c','d','e'])
print(s2)
```

o/p:

```
a    1
b    2
c    3
d    4
e    5
dtype: int64
```

retrieving first element:

```
print(s2[0])
```

o/p: 1

retrieving first 3 elements:

```
print(s2[:3])
```

o/p:

```
a    1
b    2
c    3
dtype: int64
```

retrieving last element:

```
print(s2[-1])
```

o/p: 5

retrieving data using index:

```
print(s2['a'])
```

o/p: 1

retrieving multiple elements:

```
print(s2[['b','d']])
```

o/p:

```
b    2
d    4
dtype: int64
```

```
print(s2[['c','e']])
```

o/p:

```
c    3
e    5
dtype: int64
```

If a label is not contained, an exception is raised:

```
print(s2['f'])
```

o/p:

```
KeyError: 'f'
```