DATA SCIENCE

It is the process of analyzing a large set of data points to get answers to questions related to that dataset.

⇓

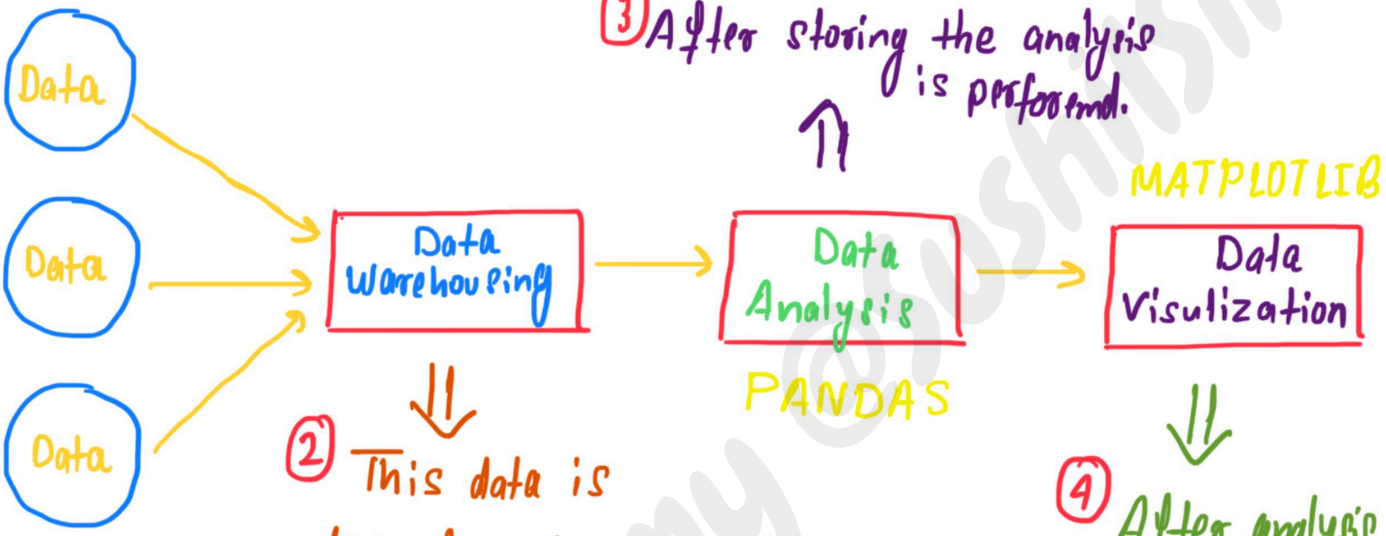
How to handle this big data ?

⇓

It can be handled through

Data Life Cycle (DLC)

Data Life Cycle

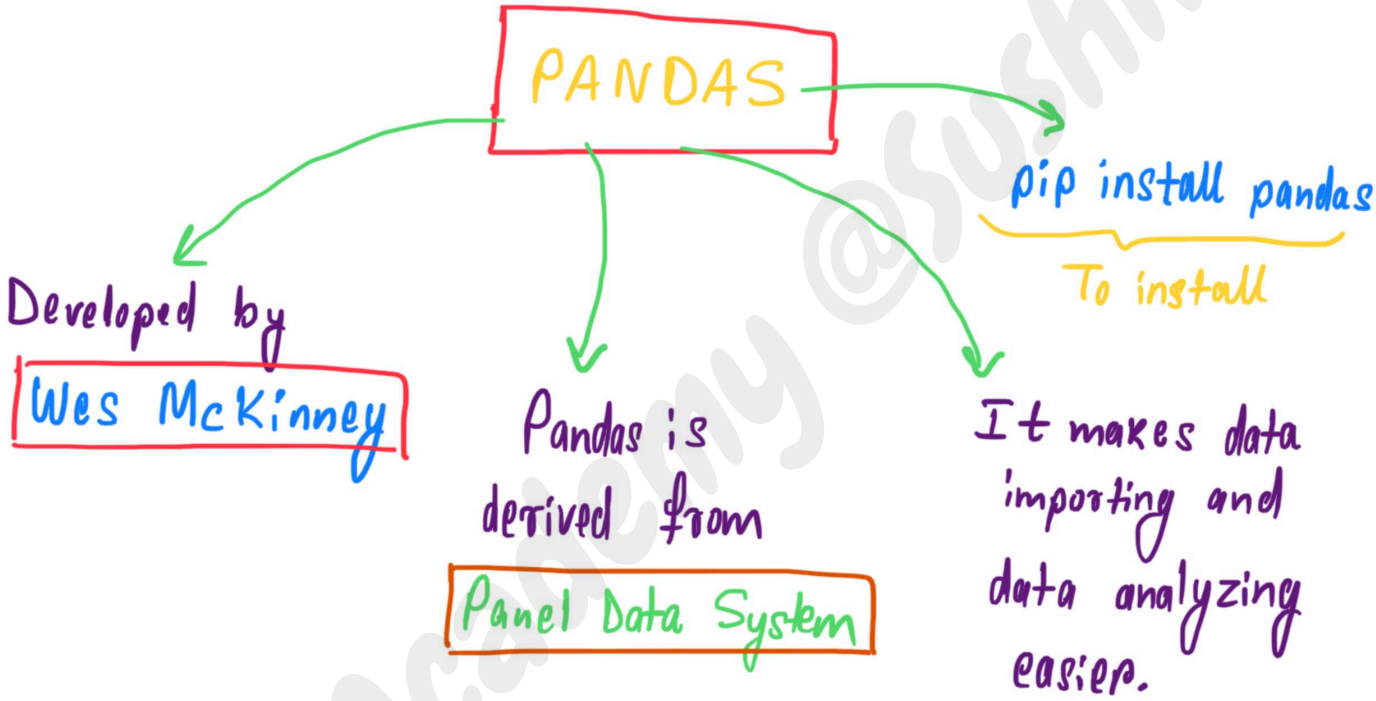


(1) Data is stored in different formats.
csv file, excel, html file etc.

(2) This data is transformed or converted into a single format.
That place is called Data Warehousing.

(3) After storing the analysis is performed.

(4) After analysis it is plotted to gain insight



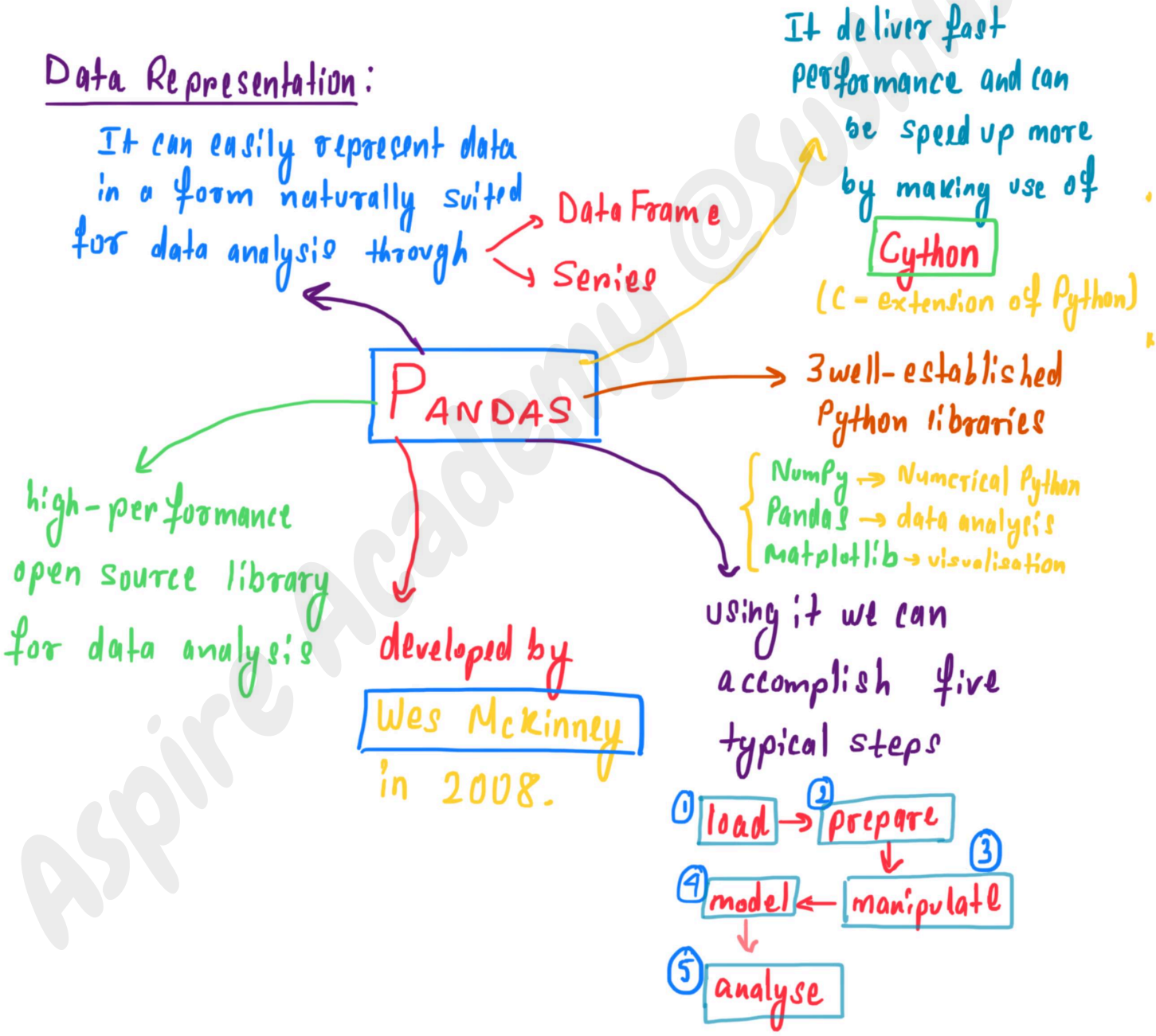
Pandas module doesn't come bundled with Standard Python.

PIP stands for "Preferred Installer Program"

PyPI stands for "Python Package Index".

To use pandas → "import pandas as pd"

Data Representation:



Data Structure In Pandas

Definition of DS

Data structure is defined as the storage and management of data for its efficient and easy access in the future where the data is collected, modified and the various type of operations are performed on data

Pandas provide two data structure

```
import pandas as pd
```

Series

- 1-D (Similar to array, list)
- assign label index to each item
- By default (index) → 0 to N
- N = series length - 1
- pd.Series(data, index, dtype, copy)
- Label of series are called index.
- Stores homogeneous data
- Data mutable (can be modified)
- Size immutable (cannot be modified)

Data Frame

- tabular data structure comprised of row and columns.
- two different indexes (row, column)
- 2-D (excel sheet)
- stores heterogeneous data
- pd.DataFrame(data, index, columns, dtype, copy)
- data & size (both mutable)
- pd.DataFrame()

SERIES

```
import pandas as pd
```

```
pd.Series(data, index = idx)
```

1. List

```
l = [1,2,3,4]
i = ['a','b','c','d']
s = pd.Series(l, index=i)
```

Access value

```
s['a']
s[['a','b','c']]
```

Slicing

```
s[0]
s[:3]
s[-3:]
```

2. ndarray

```
s1 = np.arange(10,15)
s = pd.Series(data = s1*4,
              index = s1)
```

3. Scalar value

```
pd.Series(10, index = range(0,3))
pd.Series(15, index = range(1,6,2))
```

4. dictionary

```
dict = {'Jan': 31, 'Feb': 28, 'Mar': 31}
s = pd.Series(dict)
s.name = "Days"
s.index.name = "Month"
```

5. mathematical function

```
s1 = np.arange(10,15)
s = pd.Series(s1**2,
              index = s)
```

position-based

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

name-based

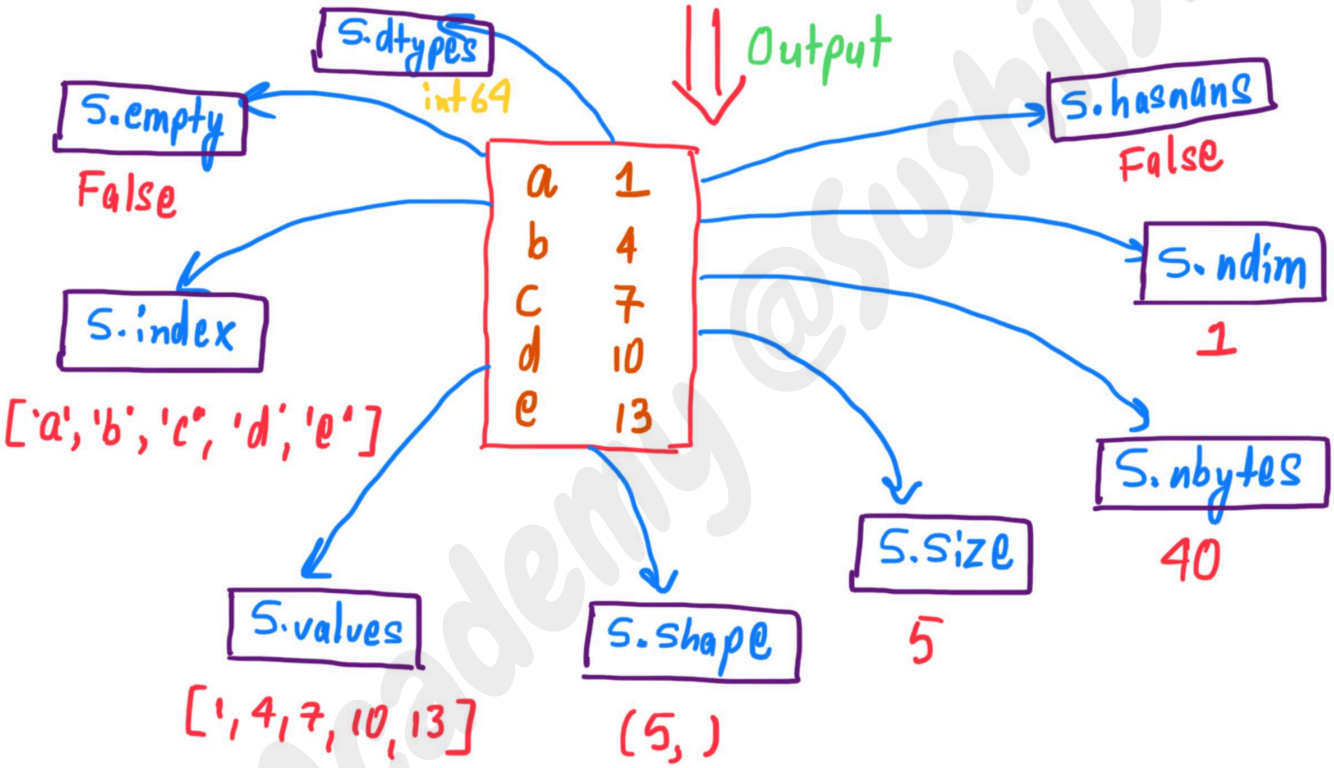
```
s.loc['b':'d']
```

Series Object Attributes

CODE

```
import pandas as pd
S= pd. Series (range (1,15,3), index = [x for x in 'abcde'])
```

Output



head() & tail functions

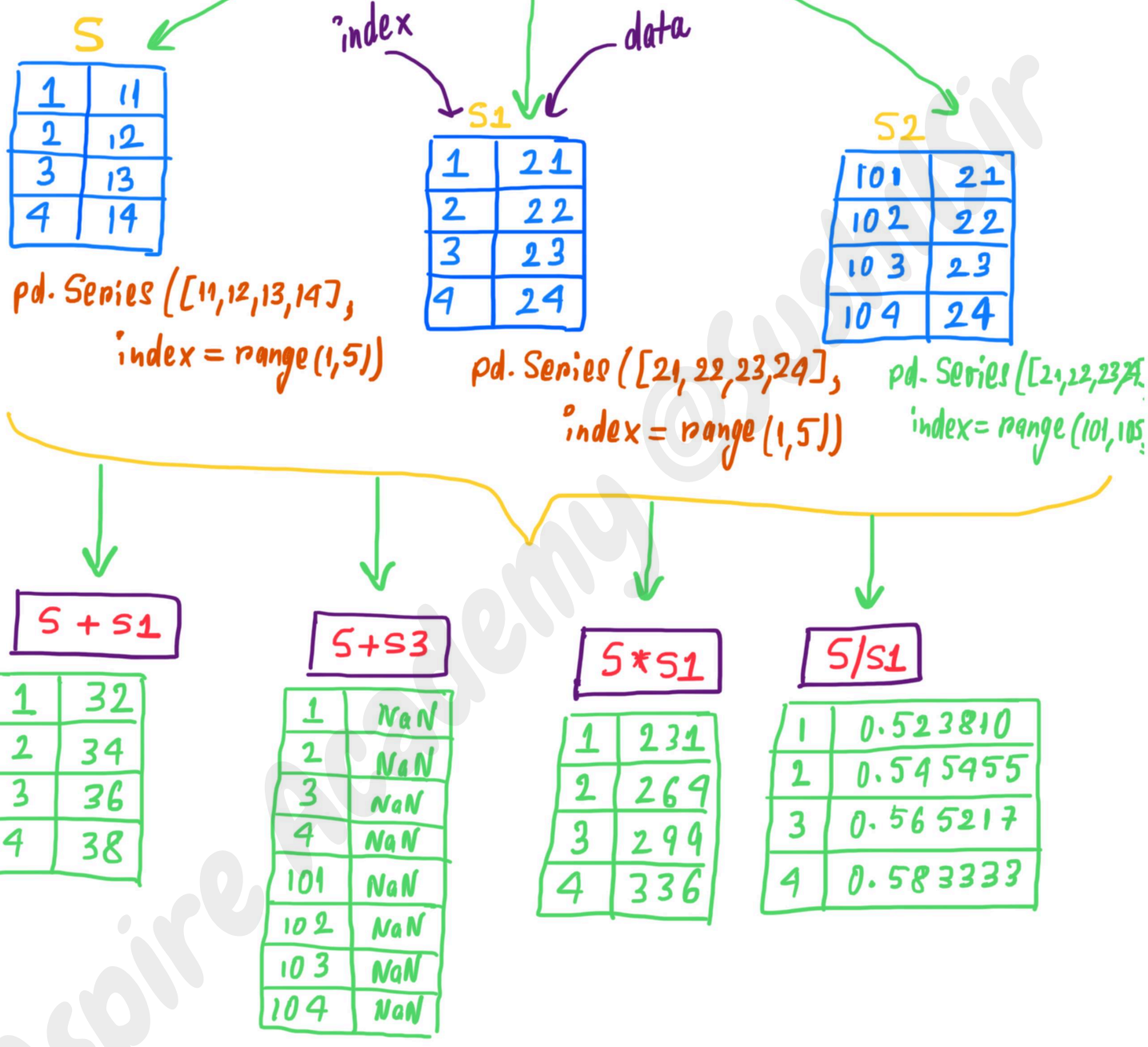
S.head()

It gives first five data by default

S.tail()

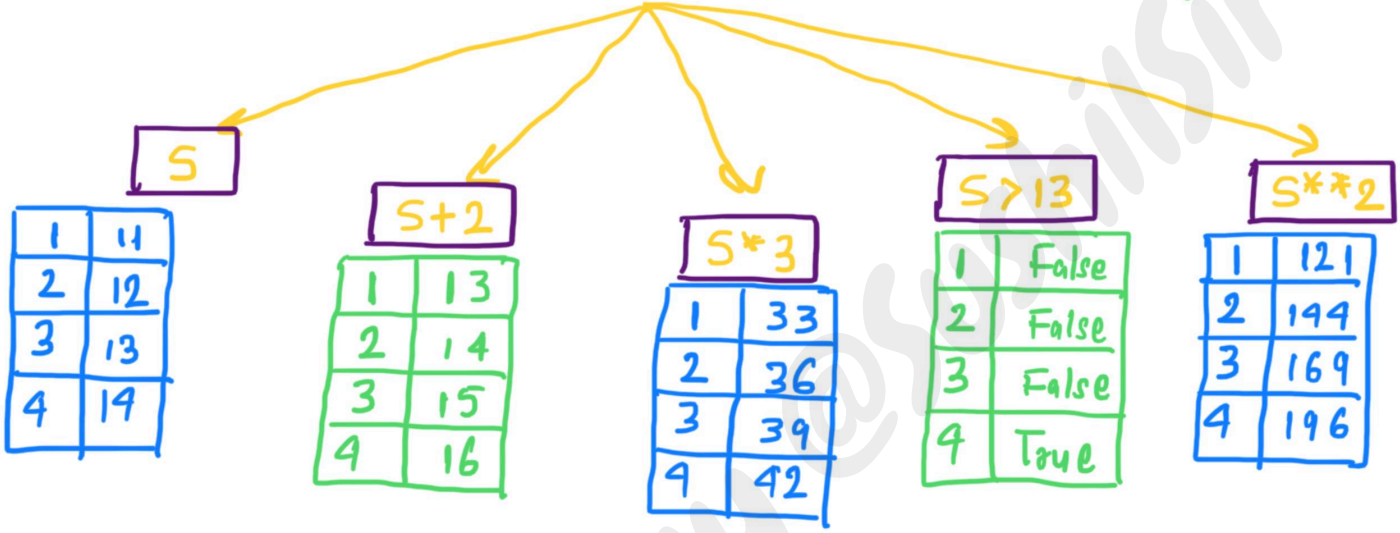
It gives last five data by default

Mathematical Operation On Series



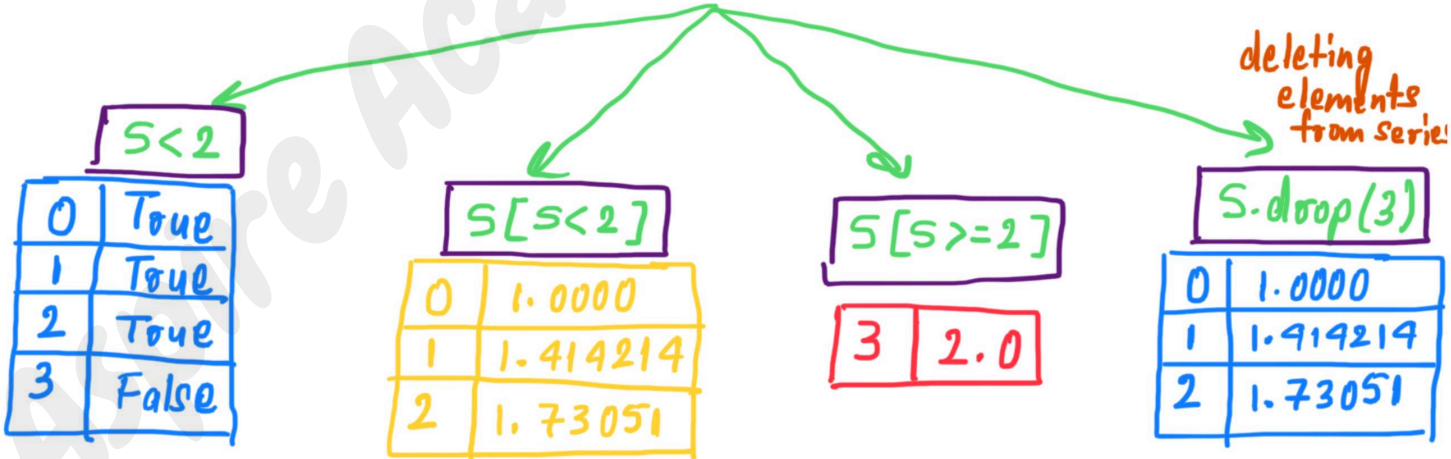
Vector Operations On Series

```
S= pd.Series([11,12,13,14], index=range(1,5))
```



Retrieving Values Using Conditions

```
S= pd.Series([1.0000, 1.414214, 1.73051, 2.0000])
```



x End Of Series DS x