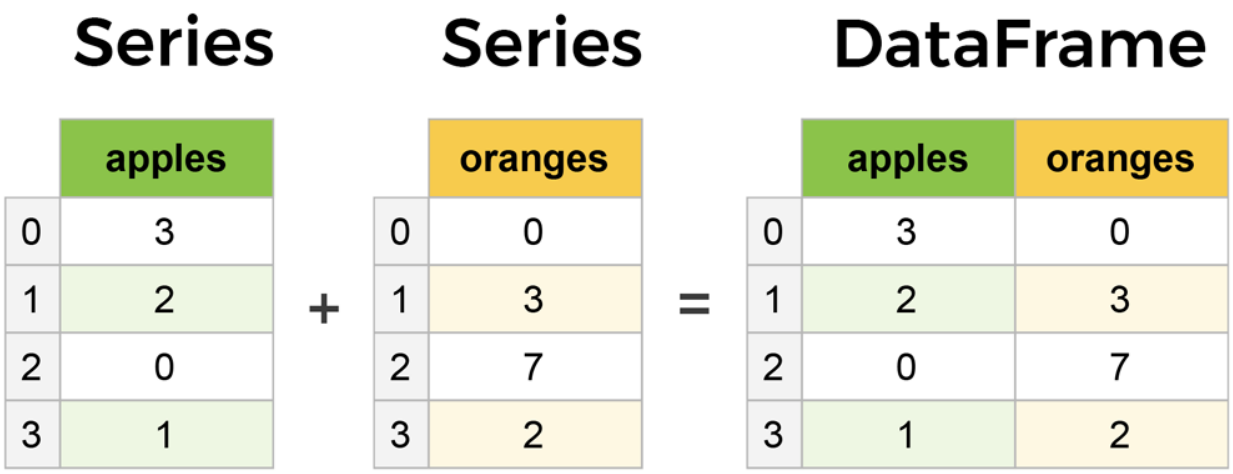
Pandas (Offers data structures and operations for Manipulating Numeric table and series)

[pandas] is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. — Wikipedia

The primary two components of pandas are the *Series* and *DataFrame*.

A Series is essentially a column, and a DataFrame is a multi-dimensional table made up of a collection of Series.

a Pandas Series : a one-dimensional labeled array capable of holding any data type with axis labels or index. An example of a Series object is one column from a DataFrame. a NumPy ndarray , which can be a record or structured. ... dictionaries of one-dimensional ndarray 's, lists, dictionaries or Series.



Series

Series is a one-dimensional array like structure with homogeneous data. For example, the following series is a collection of integers 10, 23, 56.

8	9	4	1	2	1	73	8	26	2

Key Points

1. Homogeneous data
2. Size Immutable
3. Values of Data Mutable

A pandas Series can be created using the following constructor -

pandas.Series(data, index, dtype)

Sr.No	Parameter & Description
1	**data** data takes various forms like ndarray, list, constants
2	**index** Index values must be unique and hashable, same length as data. Default **np.arrange(n)** if no index is passed.
3	**dtype** dtype is for data type. If None, data type will be inferred

Example 1.

```
import pandas as pd

data = pd.Series([10, 20, 45, 50])
print(data, type(data))
```

Output:

0 10

1 20

2 45

3 50

dtype: int64 <class 'pandas.core.series.Series'>

```
data.index = ['a', 'b', 'c', 'd', 'e']
print(data)
```

output:

a 10

b 20

c 45

d 50

dtype: int64

DataFrame

Pandas DataFrame. Pandas DataFrame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns

DataFrame is a two-dimensional array with heterogeneous data. For example,

Name	Age	Gender	Rating
Steve	32	Male	3.45
Lia	28	Female	4.6
Vin	45	Male	3.9
Katie	38	Female	2.78

Key Points

1) Heterogeneous data
2) Size Mutable
3) Data Mutable

A pandas DataFrame can be created using the following constructor −

pandas.DataFrame(data, index, columns, dtype, copy)

The parameters of the constructor are as follows –

Sr.No	Parameter & Description
1	**data** data takes various forms like ndarray, series, map, lists, dict, constants and also another DataFrame.
2	**index** For the row labels, the Index to be used for the resulting frame is Optional Default np.arrange(n) if no index is passed.
3	**columns** For column labels, the optional default syntax is - np.arrange(n). This is only true if no index is passed.

4	**dtype** Data type of each column.

Example 1

```
import pandas as pd
data = [1,2,3,4,5]
df = pd.DataFrame(data)
print (df)
```

Its **output** is as follows −

```
   0
0  1
1  2
2  3
3  4
4  5
```

Example 2

```
import pandas as pd
data = [['Alex',10],['Bob',12],['Clarke',13]]
df = pd.DataFrame(data,columns=['Name','Age'])
print (df)
```

Its **output** is as follows −

```
    Name    Age
0   Alex    10
1   Bob     12
2   Clarke  13
```

Example 3

```
import pandas as pd
data = [['Alex',10],['Bob',12],['Clarke',13]]
df = pd.DataFrame(data,columns=['Name','Age'],dtype=float)
print (df)
```

Its **output** is as follows −

```
    Name     Age
0    Alex    10.0
1    Bob     12.0
2    Clarke  13.0
```

Example 4

```
import pandas as pd
data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],'Age':[28,34,29,42]}
df = pd.DataFrame(data)
print (df)
```

Its **output** is as follows −

```
    Age     Name
0    28      Tom
1    34     Jack
2    29    Steve
3    42    Ricky
```

Example 5

Let us now create an indexed DataFrame using arrays.

```
import pandas as pd
data = {'Name':['Tom', 'Jack', 'Steve', 'Ricky'],'Age':[28,34,29,42]}
df = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])
```

print (df)

Its **output** is as follows −

```
      Age    Name
rank1  28     Tom
rank2  34     Jack
rank3  29    Steve
rank4  42    Ricky
```

Data Wrangling (Data Munging)

It involves the processing of data in various formats like concatenating, grouping, merging, etc. for the purpose of getting them used with another set of data or for analyzing.

More often than not, you find yourself dealing with a lot of data, which is of no use to you in its raw form. The process of cleaning the data enough to input to the analytical algorithm is known as Data Wrangling. It is also referred to as Data Munging

Data Wrangling with Python using Pandas Library

One of the preferred tools for data visualisation in Python is Pandas Library. It used for data manipulation and analysis. It was originally built by Numpy. The data structure offered by Pandas is fast, expressive and flexible

The Goals of Data Wrangling with Python:

- Gathering data from numerous sources to reveal a more profound intelligence within it
- Provide actionable and accurate data in the hands of business/data analysts in a timely matter

- Reduce the time spent collecting and organising, in short cleaning unruly data before it can be used
- Enable data analysts and scientists to focus on the analysis of data, not the wrangling part
- Help senior leaders in an organisation to take better decisions