

# Pandas Core Components - The DataFrame Object

## We'll cover the following

- The DataFrame Object
  - 1. Constructing a DataFrame From a Series Object
  - 2. Constructing a DataFrame From a Dictionary
  - 3. Constructing a DataFrame by Importing Data From a File

## The DataFrame Object #

In the previous lesson, we learned about Series. The next fundamental structure in Pandas that we will learn about is the DataFrame. **While a Series is essentially a column, a DataFrame is a multi-dimensional table made up of a collection of Series.** Dataframes allow us to store and manipulate tabular data where rows consist of observations and columns represent variables.

There are several ways to create a DataFrame using `pd.DataFrame()`. For example, we can create a DataFrame by passing multiple Series into the DataFrame object, we can convert a dictionary to a DataFrame or we can import data from a csv file. Let's look at each of these in detail.

### 1. Constructing a DataFrame From a Series Object #

We can create a DataFrame from a single Series by passing the Series object as input to the DataFrame creation method, along with an optional input parameter, *column*, which allows us to name the columns:

```
1 import pandas as pd
2
3 data_s1 = pd.Series([12, 24, 33, 15],
4                     index=['apples', 'bananas', 'strawberries', 'oranges'])
5
6 # 'quantity' is the name for our column
7 dataframe1 = pd.DataFrame(data_s1, columns=['quantity'])
```



```
8 print(dataframe1)
9
```



## 2. Constructing a DataFrame From a Dictionary #

We can construct a DataFrame from any list of dictionaries. Say we have a dictionary with countries, their capitals and some other variable (population, size of that country, number of schools, etc.):

```
1 dict = {"country": ["Norway", "Sweden", "Spain", "France"],
2         "capital": ["Oslo", "Stockholm", "Madrid", "Paris"],
3         "SomeColumn": ["100", "200", "300", "400"]}
4
5 data = pd.DataFrame(dict)
6 print(data)
```



We can also construct a DataFrame from a dictionary of Series objects. Say we have two different Series; one for the price of fruits and one for their quantity. We want to put all the fruits related data together into a single table. We can do this like so:

```
import pandas as pd

quantity = pd.Series([12, 24, 33, 15],
                     index=['apples', 'bananas', 'strawberries', 'oranges'])

price = pd.Series([4, 4.5, 8, 7.5],
                  index=['apples', 'bananas', 'strawberries', 'oranges'])

df = pd.DataFrame({'quantity': quantity,
                   'price': price})
print(df)
```



## 3. Constructing a Dataframe by Importing Data From a File #

It's quite simple to load data from various file formats, e.g., *CSV*, *Excel*, *json* into a DataFrame. We will be importing actual data for analyzing the IMDB-

movies dataset in the next lesson. Here is what loading data from different file formats looks like in code:

```
import pandas as pd

# Given we have a file called data1.csv in our working directory:
df = pd.read_csv('data1.csv')

#given json data
df = pd.read_json('data2.json')
```



We have only just scratched the surface and learned how to construct DataFrames. In the next lessons ***we will go deeper and learn-by-doing the many methods that we can call on these powerful objects.***