Python Pandas Tutorial

Python Pandas is defined as an open-source library that provides high-performance data manipulation in Python. This tutorial is designed for both beginners and professionals.

It is used for data analysis in Python and developed by **Wes McKinney** in 2008. Our Tutorial provides all the basic and advanced concepts of Python Pandas, such as Numpy, Data operation and Time Series

Python Pandas Introduction

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word **Panel Data**, which means **an Econometrics from Multidimensional data**. It is used for data analysis in Python and developed by **Wes McKinney** in **2008**.

Data analysis requires lots of processing, such as **restructuring, cleaning** or **merging**, etc. There are different tools are available for fast data processing, such as **Numpy, Scipy, Cython**, and **Panda**. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools.

Pandas is built on top of the **Numpy** package, means **Numpy** is required for operating the Pandas.

Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., **load, manipulate, prepare, model, and analyze**.

Key Features of Pandas

* It has a fast and efficient DataFrame object with the default and customized indexing.
* Used for reshaping and pivoting of the data sets.
* Group by data for aggregations and transformations.
* It is used for data alignment and integration of the missing data.
* Provide the functionality of Time Series.
* Process a variety of data sets in different formats like matrix data, tabular heterogeneous, time series.
* Handle multiple operations of the data sets such as subsetting, slicing, filtering, groupBy, re-ordering, and re-shaping.
* It integrates with the other libraries such as SciPy, and scikit-learn.
* Provides fast performance, and If you want to speed it, even more, you can use the **Cython**.

Benefits of Pandas

The benefits of pandas over using other language are as follows:

* **Data Representation:** It represents the data in a form that is suited for data analysis through its DataFrame and Series.
* **Clear code:** The clear API of the Pandas allows you to focus on the core part of the code. So, it provides clear and concise code for the user.

Python Pandas Data Structure

The Pandas provides two data structures for processing the data, i.e., **Series** and **DataFrame**, which are discussed below:

1) Series

It is defined as a one-dimensional array that is capable of storing various data types. The row labels of series are called the **index**. We can easily convert the list, tuple, and dictionary into series using "series' method. A Series cannot contain multiple columns. It has one parameter:

**Data:** It can be any list, dictionary, or scalar value.

**Creating Series from Array:**

Before creating a Series, Firstly, we have to import the numpy module and then use array() function in the program.

1. **import** pandas as pd
2. **import** numpy as np
3. info = np.array(['P','a','n','d','a','s'])
4. a = pd.Series(info)
5. **print**(a)

**Output**

0 P

1 a

2 n

3 d

4 a

5 s

dtype: object

**Explanation:** In this code, firstly, we have imported the **pandas** and **numpy** library with the **pd** and **np** alias. Then, we have taken a variable named "info" that consist of an array of some values. We have called the **info** variable through a **Series** method and defined it in an "**a**" variable. The Series has printed by calling the **print(a)** method.

Python Pandas DataFrame

It is a widely used data structure of pandas and works with a two-dimensional array with labeled axes (rows and columns). DataFrame is defined as a standard way to store data and has two different indexes, i.e., row index and column index. It consists of the following properties:

* The columns can be heterogeneous types like int, bool, and so on.
* It can be seen as a dictionary of Series structure where both the rows and columns are indexed. It is denoted as "columns" in case of columns and "index" in case of rows.

**Create a DataFrame using List:**

We can easily create a DataFrame in Pandas using list.

1. **import** pandas as pd
2. # a list of strings
3. x = ['Python', 'Pandas']
5. # Calling DataFrame constructor on list
6. df = pd.DataFrame(x)
7. **print**(df)

**Output**

0

0 Python

1 Pandas