Day_5 (Pandas Series)

Pandas

Series

Definition

A series in Python is a kind of one-dimensional array of any data type that we specified in the Numpy module. The only difference you can find was, each value in a Python series is associated with the index. The default index value of Python Series is from 0 to n-1, or you can specify your own indexes.

Pandas Series is nothing but a column in an excel sheet. As depicted in the picture below, columns with Name, Age and Designation representing a Series

Purpose

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.

Importance

Series is the very important data structure in the python it is like a excel datasheet or a database that is used to saved our data in the form of tables so that data will remain categorized and readable.

They are used in many ML Algorithms to hold different type of data and to perform different functions.

Strengths

It helps us in many functions to save the data in tabular form to use it and it makes the data more readable and the series datatype is used in many algorithms.

There are some algorithms nuild in the libraries fo python that require the data in the series format as it is easy to manipulate the data in series.

It is very easy to index and geet the relevent subset of information from the series containg large amount of data.

It is easy to replicate the series and to make changes in them and to update it

Weakness

It does not has a weakness as i say but there are some functions that allow th manipulation on the series data and other functions they

Example 38 :

Task:

create a pandas series use index,values and use slicing to get some data out of it.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```python
np.random.randn(5)
```

```python
s = pd.Series(np.random.randn(5), index=['a', 'b', 'c', 'd', 'e'])
```

```python
s
```

```python
s.index
```

```python
s.values
```

```python
pd.Series(np.random.randn(5))
```

```python
studentsSeries=pd.Series(studentsDict)
```

```python
studentsSeries
```

```python
studensMarks=pd.Series(studentsResult)
```

```python
studensMarks
```

```python
pd.Series({3.2,3.06,0.29,0.36,1},index=['s1','s2','s3','s4','s5'])
```

```python
studensMarks[0][0]
```

```python
studensMarks[0:3]
```

Example 39 :

Task:

Get conditional Data out of series

```python
s[s > s.median()]
```

```python
s[[4, 3, 1]]
```

Example 40 :

Task:

Get exponent of the data using exp function and get the data out of it.

```python
np.exp(s)
```

```python
s['a']
```

```python
s['e'] = 12.
```

```python
s
```

```python
'e' in s
```

```python
s['f']
```

Example 41 :

Task:

Get Data out of series using get function and apply +, * ,'**'

```python
s.get('f')
```

```python
s.get('f', np.nan)
```

```python
s+s
```

```python
s*2
```

```python
s**2
```

```python
s = pd.Series(np.random.randn(5), name='something')
s
```

Example 42 :

Task:

use name function on the series and use rename funchange its name

```python
s.name
```

```python
s2 = s.rename("different")
```

```python
s2.name
```

```python

```