

Pandas is a Python library.

Pandas is used to analyze data

Attributes	
shape	Returns a tuple representing the dimensions of the DataFrame (rows, columns).
columns	Returns the column labels of the DataFrame.
index	Returns the row labels of the DataFrame.
Data Retrieval	
head(n)	Returns the first n rows of the DataFrame.
tail(n)	Returns the last n rows of the DataFrame.
Data Information	
info()	Displays a concise summary of the DataFrame, including data types and non-null values.
describe()	Generates descriptive statistics of the DataFrame.
Data Selection	
loc[]	Access a group of rows and columns by labels.
iloc[]	Access a group of rows and columns by integer position.
Data Modification	
copy()	Creates a copy of the DataFrame.
drop(labels, axis=0)	Drops specified labels from rows or columns.
fillna(value)	Fills missing values with a specified value.
replace(to_replace, value)	Replaces values in the DataFrame with another value.
set_index(keys)	Sets the DataFrame index using existing columns.
Aggregation/Grouping	
groupby(by)	Groups DataFrame using a mapper or by a series of columns.
agg(func)	Applies aggregation functions to grouped data.
File I/O	
read_csv()	Reads a comma-separated values (csv) file into DataFrame.
to_csv()	Writes DataFrame to a CSV file.
Statistical Methods	
mean(), median(), sum()	Calculate mean, median, and sum of DataFrame values.
std(), var(), count()	Calculate standard deviation, variance, and count of DataFrame values.
Visualization	
plot()	Plot data.
hist(), boxplot()	Plot histograms and box plots.

How to read	<pre>import pandas as pd pk=pd.read_csv("data_clean1 1.csv") print(pk.to_string())</pre>
-------------	--

file  
using  
Panda  
s

```
In [1]: import pandas as pd

In [5]: pk=pd.read_csv("data_clean11.csv")

In [6]: print(pk.to_string())
```

	Order ID	Product Name	Category	Quantity Sold	Price	Order Date	Customer Location
0	00R00001	NaN	Accessories	10.0	197.425698	29-09-2020 18:05	NaN
1	00R00002	NaN	Electronics	NaN	NaN	21-12-2020 12:03	North
2	00R00003	Monitor	Accessories	3.0	89.857973	NaN	NaN
3	00R00004	Speaker	Accessories	NaN	771.762759	21-12-2020 08:49	North
4	00R00005	Mouse	Accessories	7.0	989.723357	19-06-2020 06:52	West
5	00R00006	Smartphone	Accessories	3.0	768.556563	19-05-2020 04:37	NaN
6	00R00007	Tablet	Electronics	8.0	589.358678	11-02-2020 01:55	East
7	00R00008	Headphones	Electronics	9.0	795.220567	25-05-2020 20:59	West
8	00R00009	Monitor	Accessories	NaN	202.651940	04-06-2020 23:15	North
9	00R00010	Keyboard	Electronics	2.0	269.953723	16-05-2020 18:56	South
10	00R00011	Tablet	Electronics	8.0	NaN	16-03-2020 00:13	NaN
11	00R00012	Keyboard	Electronics	5.0	427.484703	28-04-2020 01:15	South
12	00R00013	Smartphone	Electronics	5.0	76.208596	08-07-2020 23:41	West
13	00R00014	NaN	Accessories	1.0	724.583414	04-12-2020 10:52	South
14	00R00015	NaN	Electronics	5.0	673.025034	18-01-2020 02:36	West
15	00R00016	Keyboard	Electronics	8.0	333.163006	09-12-2020 14:43	South

Show array data in table form	<pre>df = pd.DataFrame([[5,7,3]],columns=['Python','pyspark','Azur'])  print(df)</pre> <table><thead><tr><th></th><th>Python</th><th>pyspark</th><th>Azur</th></tr></thead><tbody><tr><td>0</td><td>5</td><td>7</td><td>3</td></tr></tbody></table>		Python	pyspark	Azur	0	5	7	3
	Python	pyspark	Azur						
0	5	7	3						

Series-column in table  
One-dimensional array holding data

Create series in pandas	<pre>a=[1,2,3,4] pk=pd.Series(a) print(pk)</pre> <table> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>2</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>3</td><td>4</td></tr> </tbody> </table> <pre>dtype: int64</pre>	0	1	1	2	2	3	3	4
0	1								
1	2								
2	3								
3	4								
Add dynamic index	<pre>a=[1,2,3,4] pk=pd.Series(a,index=["ondo","erdo","muro","nalko"]) print(pk)</pre> <table> <tbody> <tr><td>ondo</td><td>1</td></tr> <tr><td>erdo</td><td>2</td></tr> <tr><td>muro</td><td>3</td></tr> <tr><td>nalko</td><td>4</td></tr> </tbody> </table> <pre>dtype: int64</pre>	ondo	1	erdo	2	muro	3	nalko	4
ondo	1								
erdo	2								
muro	3								
nalko	4								
Convert dimension into series	<pre>import pandas as pd  calories = {"day1": 420, "day2": 380, "day3": 390}  myvar = pd.Series(calories)</pre>								

	<pre> print(myvar) day1    420 day2    380 day3    390 dtype: int64 </pre>
Convert dataframe into series and print specified index values	<pre> import pandas as pd  data = {     "calories": [420, 380, 390],     "duration": [50, 40, 45] }  #load data into a DataFrame object: df = pd.DataFrame(data)  print(df.loc[[0, 1]]) 0    [420, 380] 1    [50, 40] dtype: object </pre>
Create Dataframe from json file	<pre> import pandas as pd  data = {     "Duration":{         "0":60,         "1":60,         "2":60,         "3":45,         "4":45,         "5":60     },     "Pulse":{         "0":110,         "1":117,         "2":103,         "3":109,         "4":117,         "5":102     },     "Maxpulse":{         "0":130,         "1":145,         "2":135,         "3":175,         "4":148,         "5":127     },     "Calories":{ </pre>

	<pre> "0":409, "1":479, "2":340, "3":282, "4":406, "5":300 } }  df = pd.DataFrame(data)  print(df) </pre>
--	---

<p>The head() method if the number of rows is not specified, the head() method will return the top 5 rows.</p>	<pre> import pandas as pd  df = pd.read_csv('data_clean1 1.csv')  print(df.head(10)) </pre>
tail() method - returns last 5 rows	<pre> print(df.tail()) </pre>
To get info of csv file	<pre> print(df.info()) </pre> <p>Output=</p> <pre> &lt;class 'pandas.core.frame.DataFrame'&gt; RangeIndex: 1000 entries, 0 to 999 Data columns (total 7 columns): #   Column          Non-Null Count  Dtype ---  --- 0   Order ID        930 non-null  object 1   Product Name    908 non-null  object 2   Category        1000 non-null  object 3   Quantity Sold   908 non-null  float64 4   Price           895 non-null  float64 5   Order Date      942 non-null  object 6   Customer Location 808 non-null  object dtypes: float64(2), object(5) memory usage: 54.8+ KB None </pre>