Series:

--> A Pandas Series is a ***one-dimensional labeled array in Python*** that is capable of ***holding any data type***—such as integers, floats, strings, Python objects, etc.

--> A ***Series Combines*** the best features of a ***list and a dictionary***

--> A ***series maintains*** a single collection if ordered value.

***One-dimensional:***

--> Like a ***column in a table*** or a single list of values.

***Labeled index:***

--> Each item in the ***series has a label (index) by default***, starting from 0 unless specified.

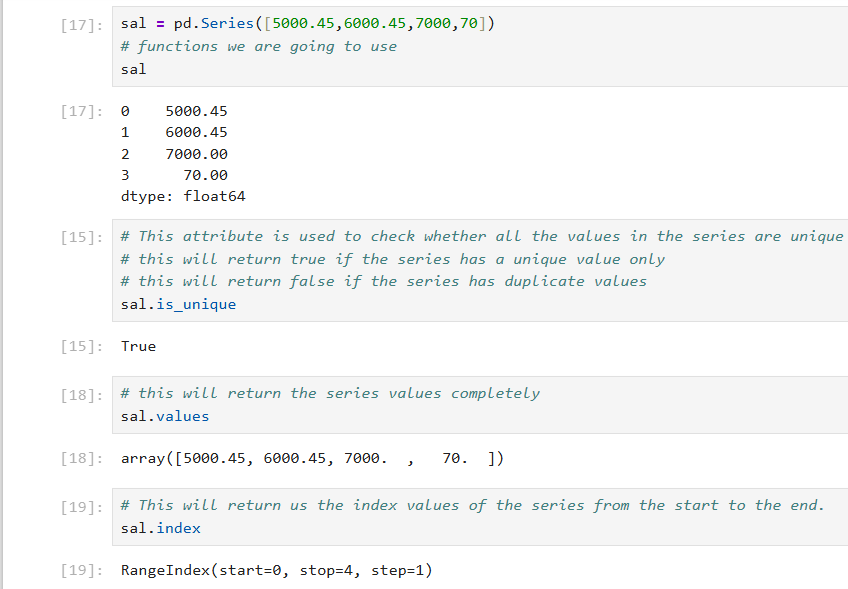
***Methods and Attributes:***

***In pandas series:***

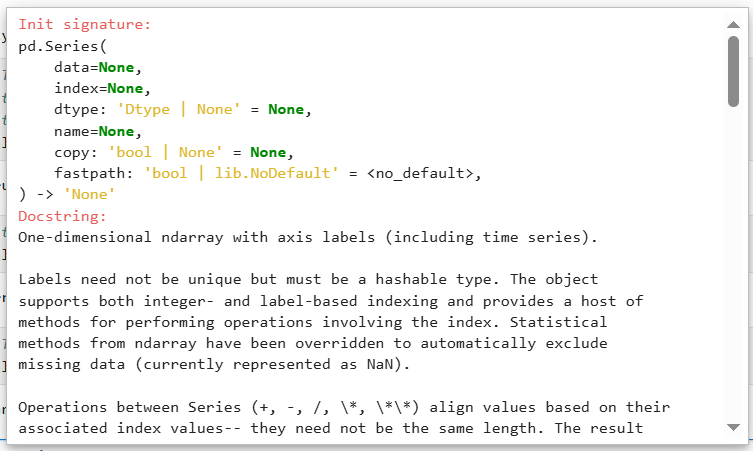
--> ***attributes*** tells about the object and does not required any paraenthesis

--> ***methods*** is a behavior or action available in the series





***Arguments in the pd.Series() syntax.***



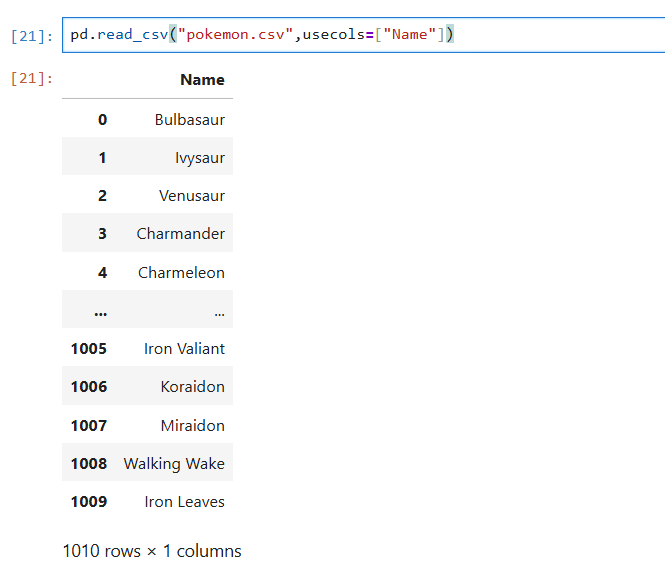
***data*** which will take the input source , ***index*** which will take custom index value which we map for the data if ***do not give index*** by default ***from 0 index position*** will get assigned to each values in the data.

Import Series with the ***pd.read\_csv*** Function

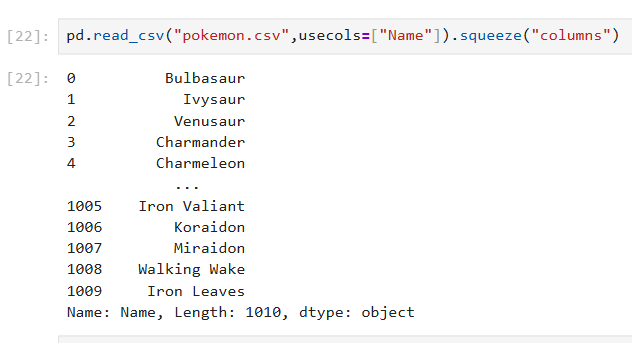
A ***CSV is a plain text file*** that uses line breaks to ***separate rows and commas to separate row values***.

1. Pandas ships with many ***different read\_ functions for different types of files***.
2. The ***read\_csv function*** accepts many different parameters. The first one specifies the file ***name/path***.
3. The ***read\_csv function*** will import the ***dataset as a DataFrame, a 2-dimensional table***.
4. The ***usecols parameter*** accepts a list of the ***column(s) to import***.
5. The ***squeeze method*** converts a ***DataFrame to a Series***.
6. By using the ***squeeze method*** we are going to convert the ***dataframe to series*** where we need to have ***only one column*** in the dataframe
7. ***Pandas Series method*** can support ***only one column*** and provide ***one dimesional array view***

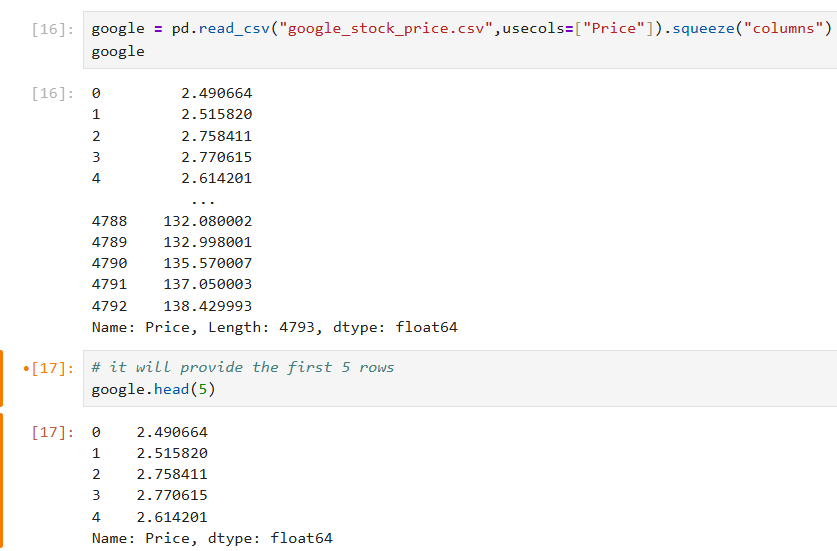
Two dimesional View in DataFrames:



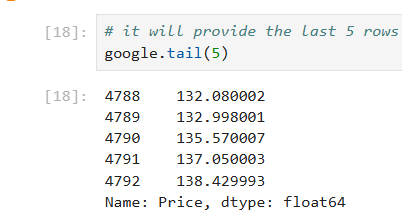
After converting the dataFrames to Series by using the squeeze method we can view the one dimesional view:



The ***head method*** returns a number of rows from the beginning of the series.



The ***tail method*** returns a number of rows from the end of the series.



***Passing Series to Python's Built-In Functions***

The ***len function*** returns the ***length of the Series***.

The ***type function*** returns the ***type of an object***.

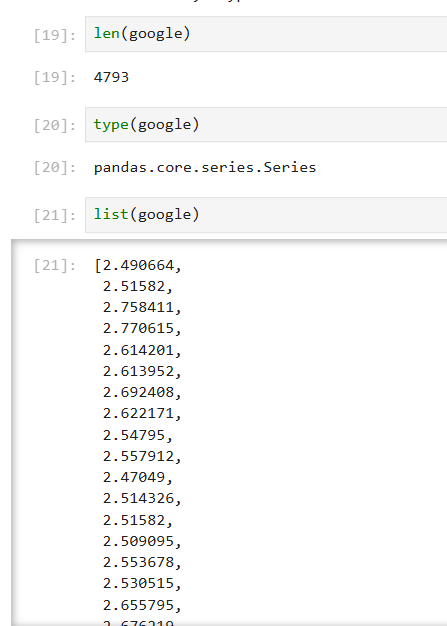
The ***list function*** converts the ***Series to a list***.

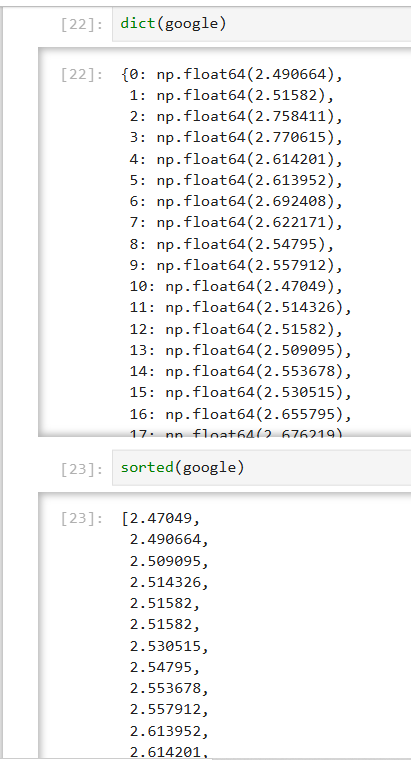
The ***dict function*** converts the ***Series to a dictionary***.

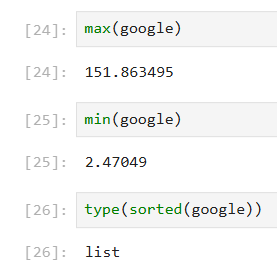
The ***sorted function*** converts the ***Series to a sorted list***.(ascending to descending order)

The ***max function*** returns the ***largest value*** in the Series.

The ***min function*** returns the ***smallest value*** in the Series.







***Check for Inclusion with Python's in Keyword***

The ***in*** keyword checks ***if a value exists within an object***.

The ***in*** keyword will look ***for a value in the Series's index***.

Use the ***index and values*** attributes to ***access "nested" objects*** within the Series.

Combine the ***in keyword*** with values to search within the ***Series's values.***

***The sort\_values Method***

The ***sort\_values*** method sorts a ***Series values in order***.

By default, pandas **applies** an ***ascending sort (smallest to largest)***.

Customize the ***sort order*** with the ***ascending parameter***.

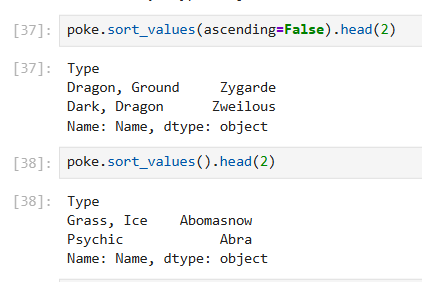
***The sort\_index Method***

The ***sort\_index*** method sorts a ***Series by its index***.

The ***sort\_index method*** also accepts an ascending parameter to ***set sort order***.

In the eaxmple ***pokemon*** we can see we have the ***index\_col as Type*** and value as ***Name***





For the ***string sorting based on the alphabtes*** only sort will be happening.

