# Comprehensive List of PANDAS-SERIES

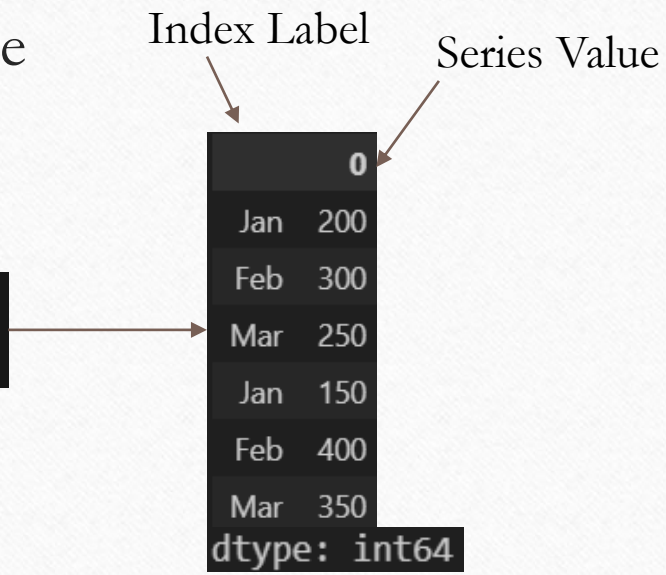
## Functions and Methods

Rajendra Prasad JM
https://chlorinexxe.github.io/portfolio

# Series ? What's That ?

- One Dimensional ndarray with index labels.
- It is Homogenous , Labeled , Size Immutable
- Labels need not to be unique but hashable

```
data = {'Month': ['Jan', 'Feb', 'Mar', 'Jan', 'Feb', 'Mar'],
        'Sales': [200, 300, 250, 150, 400, 350]}
sales = pd.Series(data['Sales'], index=data['Month'])
```



# Creating and Inspecting

- pandas.Series(data,name,index) : Creates a new Series object
- .head(n=5) : Returns first n rows
- .tail(n=5) : Returns last n rows
- .sample(n=5) : Return random n rows
- .describe(n=5) : Generates descriptive statistics

name in Series or DataFrame creation is not necessary and used for displaying the series using the interpreter

# Manipulations and Transformations

- .astype() : Converts data type of the Series
- .copy() : Creates a deep copy of the Series
- .append(to_append) : Appends another Series
- .apply(func) : Applies a function to each element
- .map(dict or func) : Applies mapping or function
- .str.*(string methods) : String operations

# Aggregation and Statistical Operations

- .sum() : Computes sum of elements
- .mean() : Computes mean of elements
- .median() : Computes median of elements
- .std() : Computes standard deviation
- .min() : Returns minimum value
- .max() : Returns maximum value
- .value_counts() : Returns counts of unique values

# Sorting and Indexing

- .sort_values()  : Sorts by values
- .sort_index()  : Sorts by index labels

# Selection, Filtering, and Handling Missing Data

- .isnull() : Detects missing values(NaN) in a Series
- .notnull() : Detects non-missing values in a Series
- .fillna(value) : Fills missing values (used for filling median or means)
- .dropna() : Drops missing values
- .isin(values) : Checks if values are in Series

# Serialization and Output

- .tolist() : Converts the Series to a Python list
- .to_csv(path) : Writes to CSV file

# Visualization

- .plot(*arg,**kwargs) : Plots data
- .hist(*arg,**kwargs) : Plots Histogram

With a series you can plot with the following .plot(kind=“ ”)

| | | | | |
|---|---|---|---|---|
| Line Plot | - “line” (Default) | | Density Plot | - “density” |
| Bar Plot | - “bar” or “barh” | | Area Plot | - “area” |
| Hist plot | - “hist” | | Pie Plot | - “pie” |
| Box Plot | - “box” | | | |

# Bonus Functions and Methods

- Advanced Indexing and Selection : Multi-indexing, .loc[ ] , .iloc[ ]
- Handling Datetime Data : Date and time manipulation pd.to_datetime , Series.dt.
- Grouping and Aggregation : .groupby() operations, custom aggregation functions with .agg()
- Merging and Joining : Combining Series with .merge() or .join()
- Time Series Analysis
  - Resampling : Series.resample()
  - Rolling windows : Series.rolling()
- Performance Optimization : Efficient memory usage with pd.Series.astype() as 'category' dtype, vectorized operations

“ I hope you found this information helpful!
Feel free to save this post for future reference.
Let's continue to learn and grow together! 🚀

Thank You for Your Support

Rajendra Prasad JM ”

https://chlorinexxe.github.io/portfolio