

### Creating Series

|                     |   |
|---------------------|---|
| From a list:        | <code>series = pd.Series(my_list)</code>  |
| From a dictionary:  | <code>series = pd.Series(my_dict)</code>  |
| From a NumPy array: | <code>series = pd.Series(my_array)</code> |

### Constructor

|          |   |
|----------|---|
| data     | used to pass the data for the Series.                     |
| index    | used to specify the index labels for the Series           |
| dtype    | used to specify the data type of the Series               |
| copy     | used to specify whether to make a copy of the data or not |
| name     | used to give a name to the Series                         |
| example: | <code>pd.Series(data=my_data, index=my_index)</code>      |

### Series Attributes

|           |  |
|-----------|--|
| values    | Returns a NumPy array containing the values of the Series                        |
| index     | Returns the index labels of the Series   |
| shape     | Returns the dimensions of the Series in the form of a tuple.                     |
| size      | Returns the number of elements in the Series.                                    |
| name      | Returns the name of the Series.  |
| dtype     | Returns the data type of the values in the Series.                               |
| is_unique | Boolean value indicating whether all the values in the Series are unique or not. |

### Data Selection

|        |  |
|--------|--|
| head() | Return the first n rows of the Series (default value of n is 5). |
| tail() | Return the last n rows of the Series (default value of n is 5).  |

### Data Selection (cont)

|                |   |
|----------------|---|
| sample()       | Return a random sample of the Series.                               |
| value_counts() | Count the number of occurrences of each unique value in the Series. |

### Sorting

|               |                           |
|---------------|---------------------------|
| sort_values() | Sort the Series by value. |
| sort_index()  | Sort the Series by index. |

### Math

|            |  |
|------------|--|
| count()    | Return the number of non-missing values in the Series. |
| sum()      | Return the sum of the values in the Series.            |
| mean()     | Return the mean of the values in the Series.           |
| min()      | Return the minimum value in the Series.                |
| max()      | Return the maximum value in the Series.                |
| describe() | Generate descriptive statistics for the Series.        |

### Python Functions

```
# len/type/dir/sorted/max/min
type(student_series)
# type conversion
list(marks_series)
# Arithmetic Operators(Broadcasting)
100 + marks_series
```

### Indexing

|                  |   |
|------------------|---|
| Fancy Indexing   | <code>students[[1,3,4]]</code> , <code>students[['A','B']]</code>                 |
| Boolean Indexing | <code>student_mark[student_mark &gt;= 50]</code>                                  |
| Normal Indexing  | <code>student[1]</code> , <code>students[1:2]</code> , <code>students[:,2]</code> |
| Error            | <code>students[[1:2]]</code> , <code>student[-1]</code>                           |

### Plotting

|           |  |
|-----------|--|
| plot()    | Creates a line plot of the Series.         |
| bar()     | Creates a vertical bar plot of the Series. |
| hist()    | Creates a histogram plot of the Series.    |
| scatter() | Creates a scatter plot of the Series.      |

### Plot()

|         |  |
|---------|--|
| kind    | The type of plot to create, e.g., 'line', 'bar', 'hist', 'kde', 'pie', etc. Defaults to 'line' |
| title   | The title of the plot.   |
| xlabel  | The label for the x-axis   |
| ylabel  | The label for the y-axis.  |
| color   | The color of the plot.   |
| legend  | Whether or not to show a legend. Defaults to True.   |
| figsize | The size of the plot figure in inches.   |
| grid    | Whether or not to show a grid. Defaults to True.   |
| xticks  | the positions of the x-axis ticks.   |
| yticks  | Additional parameters for the specific plot  |

### Data Manipulation:

|                   |  |
|-------------------|--|
| astype()          | Convert the data type of the values in the Series to a specified data type.                |
| between()         | Return a Boolean Series indicating whether each element is between two values (inclusive). |
| clip()            | Limit the values in the Series to a specified range.                                       |
| drop_duplicates() | Remove duplicate values from the Series.   |
| dropna()          | Remove missing values from the Series.   |



### Data Manipulation: (cont)

`fillna()` Fill missing values in the Series with a specified value or method.

`isin()` Determine whether each element is in a specified set of values.

`apply()` Apply a function to each element of the Series.

`map()` Map values of the Series to a new set of values.

`copy()` Create a copy of the Series.

`hasnans()` Return a Boolean value indicating whether the Series contains any missing values.

### DataFrame Attributes

`shape` returns a tuple representing the dimensions of the DataFrame

`index` returns the row index labels of the DataFrame.

`columns` returns the column index labels of the DataFrame.

`values` returns a NumPy array containing the data of the DataFrame.

`dtypes` returns a Series containing the data types of each column in the DataFrame.

`size` returns the total number of elements in the DataFrame.

`ndim` returns the number of dimensions of the DataFrame.

`empty` returns a Boolean value indicating whether the DataFrame is empty or not.

`axes` returns a list containing the row and column axis labels of the DataFrame.

### DataFrame Indexing and selecting data

`head()` returns the first n rows of a DataFrame

`tail()` returns the last n rows of a DataFrame

`sample()` returns a random sample of a DataFrame

`loc[]` label-based indexing for selecting rows and columns

`iloc[]` integer-based indexing for selecting rows and columns

### Descriptive statistics:

`info()` prints information about a DataFrame, including data types and non-null values

`describe()` provides summary statistics for numerical columns

`value_counts()` counts the unique values in a column

`unique()` returns an array of unique values in a column

`nunique()` returns the number of unique values in a column

`rank()` returns the rank of each value in a column

`corr()` calculates the correlation between columns

`nlargest()` returns the n largest values in a column

`nsmallest()` returns the n smallest values in a column

`max()` returns the maximum value in a column

`min()` returns the minimum value in a column

`mean()` returns the mean of values in a column

`var()` returns the variance of values in a column

### Data manipulation:

`rename()` renames columns or index labels of a DataFrame

`.astype()` converts the data type of a column

`sort_values()` sorts a DataFrame by values

`set_index()` sets a column as the DataFrame's index

`reset_index()` resets the DataFrame's index to a default range index

`sort_index()` sorts the DataFrame by index values

`drop()` removes rows or columns from a DataFrame

`apply()` applies a function to each element of a DataFrame

`isin()` checks whether each element of a DataFrame is contained in a list of values

### Missing data

`isnull()` returns a boolean DataFrame indicating missing values

`notnull()` returns a boolean DataFrame indicating non-missing values

`dropna()` removes rows or columns containing missing values

`fillna()` fills missing values with a specified value or method

`duplicated()` returns a boolean Series indicating duplicate rows

`drop_duplicates()` removes duplicate rows

### Other

`size()` returns the number of elements in a DataFrame

`insert()` inserts a column into a DataFrame at a specified location

`copy()` creates a copy of a DataFrame