Essential Functionalities

Essential Functionalities

- Reindexing
- Indexing & Selection
- Integer Indexing
- Arithmetic & Data Alignment
- Function Application & Mapping
- Sorting & Ranking
- Axis Indexing With Duplicate Values
- Dropping Entries

Essential Functionalities

Reindexing

“reindex”

series

`reindex:`

Which introduces new index object with values of the original index object remains same

If the new index object in reindex method has new index values then then those missed objects values are filled with NaN or NA

In order to fill the missing values while “reindex”ing we can use ‘method’ option which provides ‘ffill’ and ‘bfill’

Essential Functionalities

Reindexing

“reindex”

DataFrame

reindex:

Which can change either row index, column index, or both

By default if we pass only the sequence, it will alter only the row index.

To change the column index, we can use ‘columns’ keyword. We can also use ‘loc’ to reindex both rows and column index simultaneously.

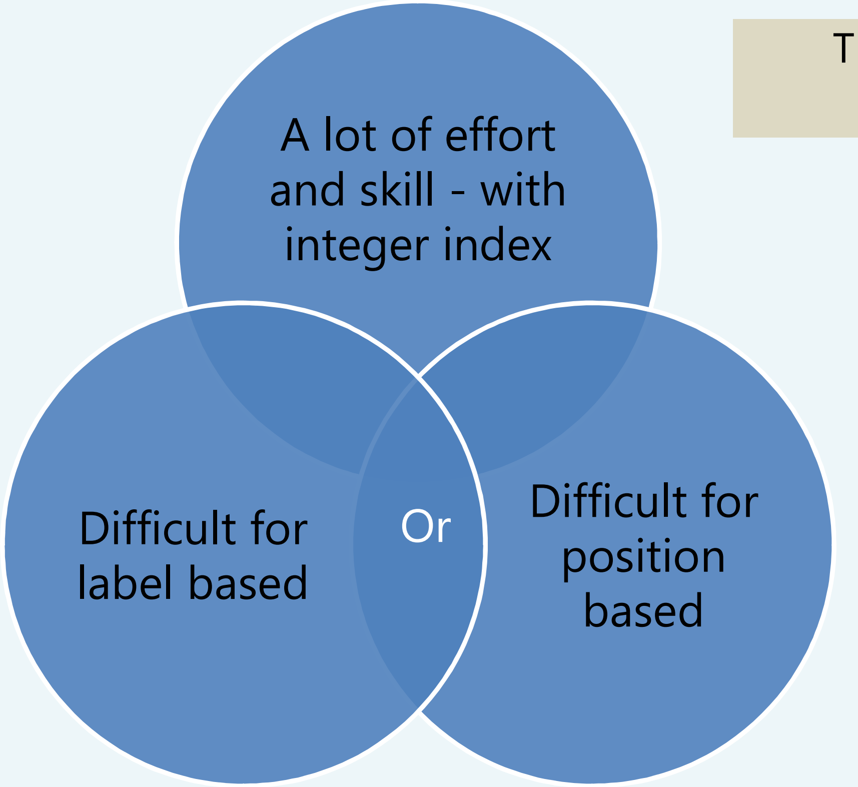
Essential Functionalities

Summary of ‘reindex’ function arguments

index	It is a New sequence to use as index. It Can be Index instance or any other sequence-like Python data structure
method	Interpolation (fill) method; 'ffill' fills forward, while 'bfill' fills backward.
fill_value	Substitute value to use when introducing missing data by reindexing.
limit	When forward- or backfilling, maximum size gap (in number of elements) to fill
tolerance	When forward- or backfilling, maximum size gap (in absolute numeric distance) to fill for inexact matches.
level	Match simple Index on level of MultiIndex; otherwise select subset of.

Essential Functionalities

Integer Indexing



A lot of effort and skill - with integer index

Difficult for label based

Or

Difficult for position based

There Is A Potential For Ambiguity

Essential Functionalities

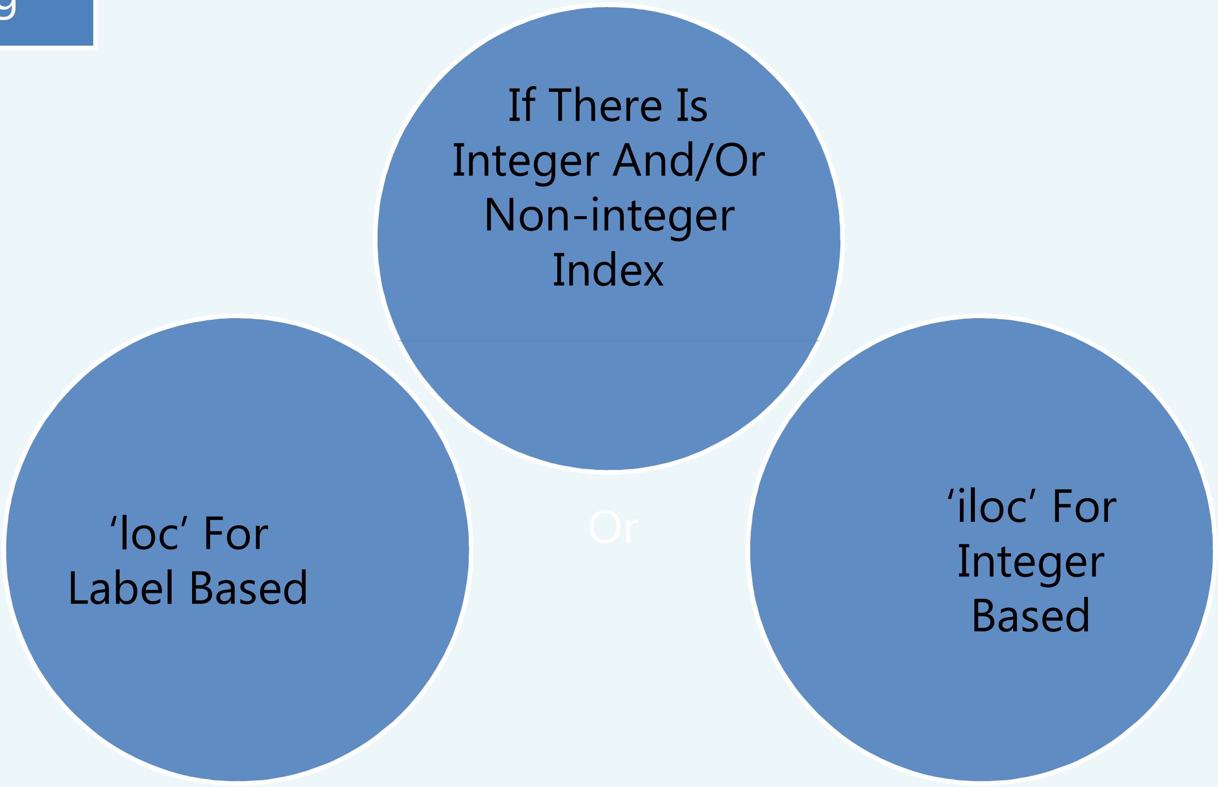
Integer Indexing

If There Is A non-integer Index

There Is No Potential For Ambiguity

Essential Functionalities

Integer Indexing



If There Is Integer And/Or Non-integer Index

‘loc’ For Label Based

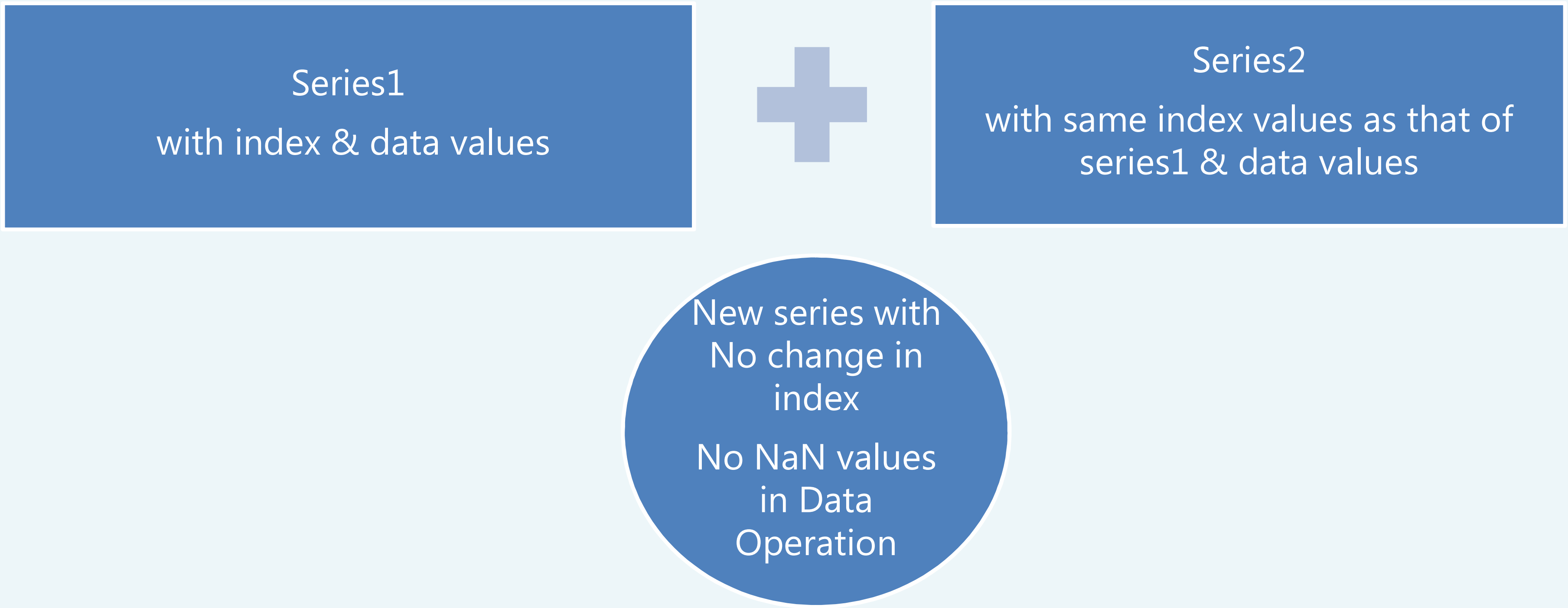
Or

‘iloc’ For Integer Based

Essential Functionalities

Arithmetic & Data Alignment

Series with same index and No missing values



Series1

with index & data values

Series2

with same index values as that of series1 & data values

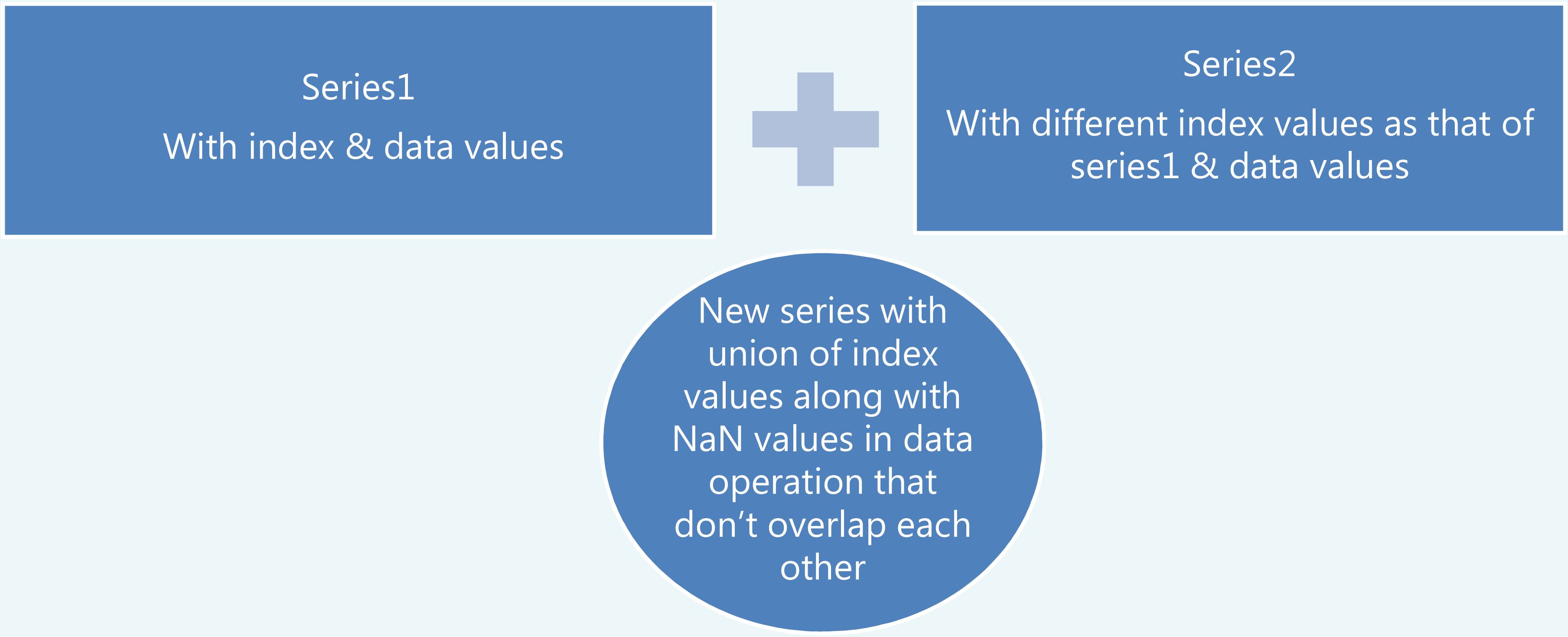
New series with No change in index

No NaN values in Data Operation

Essential Functionalities

Arithmetic & Data Alignment

Series with different and missing index values



Series1

With index & data values

Series2

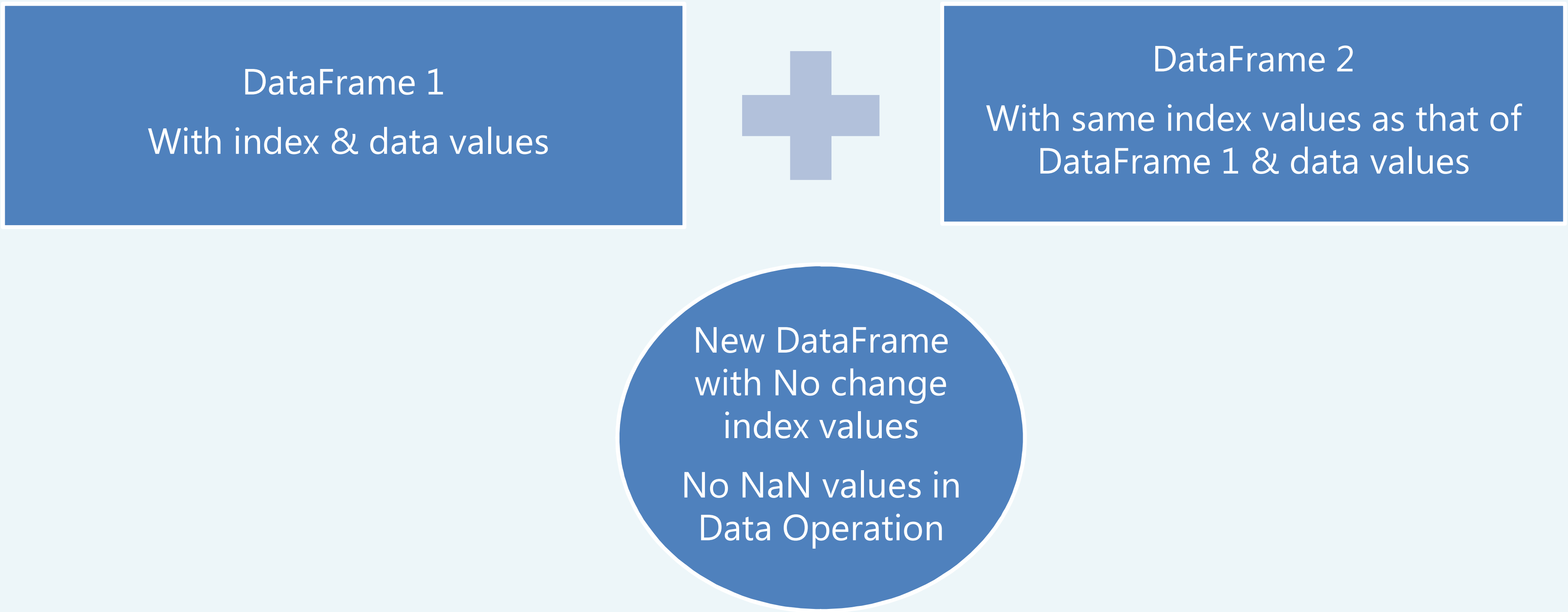
With different index values as that of series1 & data values

New series with union of index values along with NaN values in data operation that don’t overlap each other

Essential Functionalities

Arithmetic & Data Alignment

DataFrame with No missing data values



DataFrame 1

With index & data values

DataFrame 2

With same index values as that of DataFrame 1 & data values

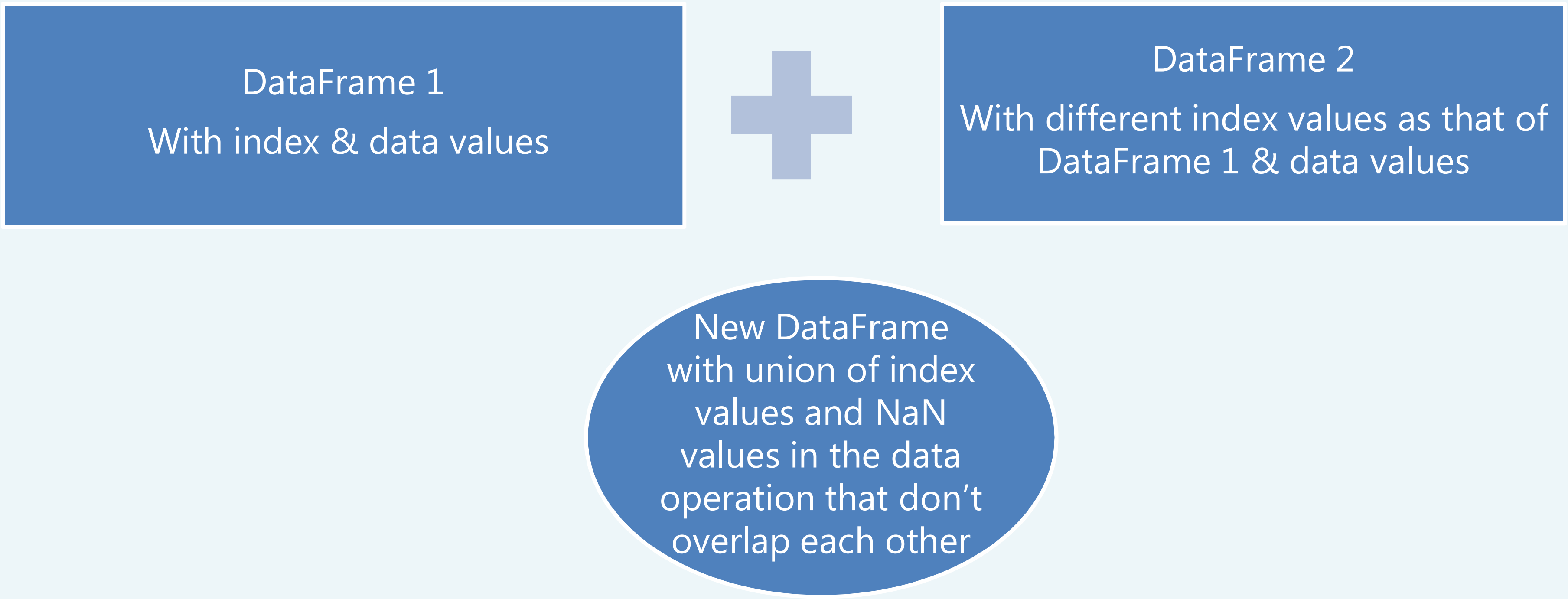
New DataFrame with No change index values

No NaN values in Data Operation

Essential Functionalities

Arithmetic & Data Alignment

DataFrame with different index values



DataFrame 1

With index & data values

DataFrame 2

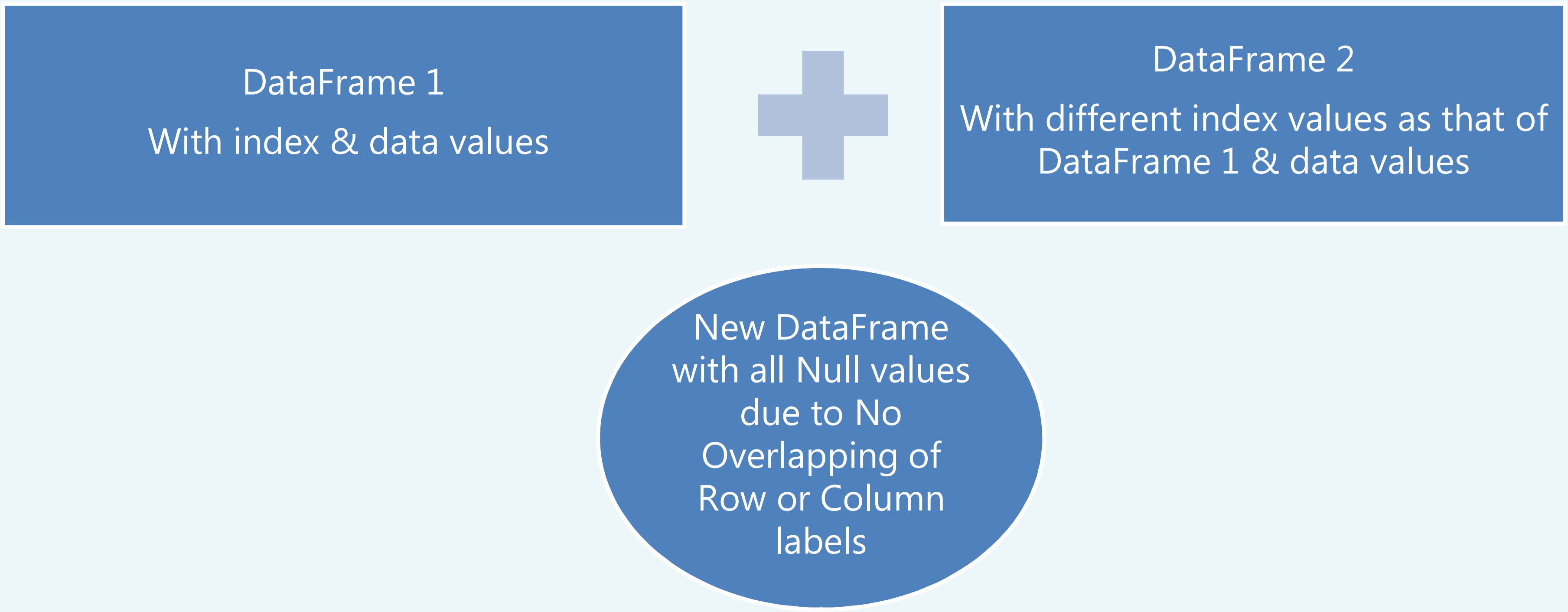
With different index values as that of DataFrame 1 & data values

New DataFrame with union of index values and NaN values in the data operation that don't overlap each other

Essential Functionalities

Arithmetic & Data Alignment

DataFrame with No Common labels/index values



DataFrame 1

With index & data values

DataFrame 2

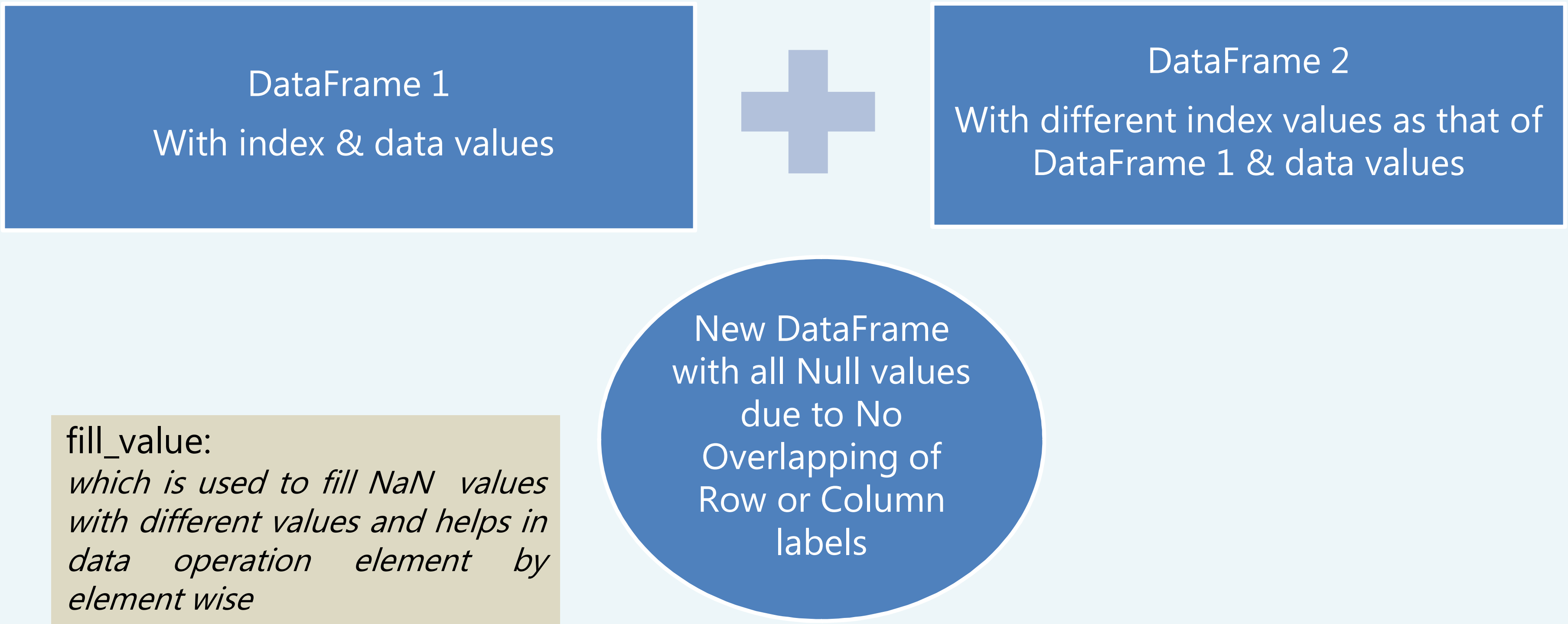
With different index values as that of DataFrame 1 & data values

New DataFrame with all Null values due to No Overlapping of Row or Column labels

Essential Functionalities

Arithmetic & Data Alignment

DataFrame with No Common labels/index values



DataFrame 1

With index & data values

DataFrame 2

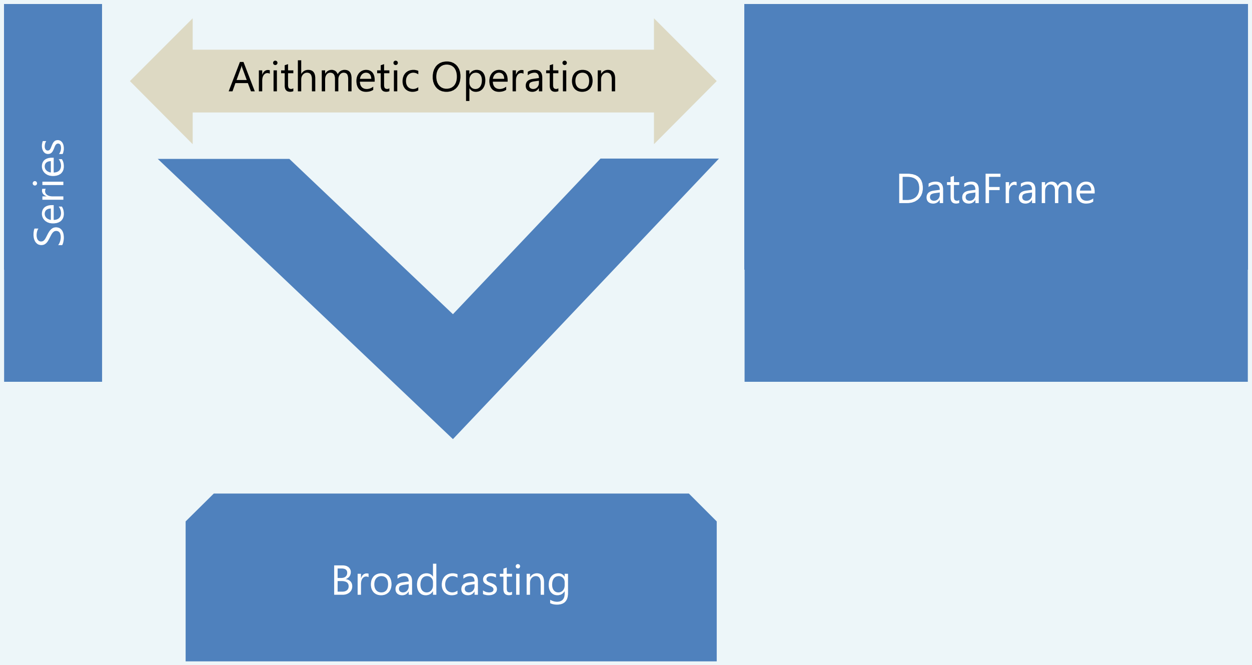
With different index values as that of DataFrame 1 & data values

New DataFrame with all Null values due to No Overlapping of Row or Column labels

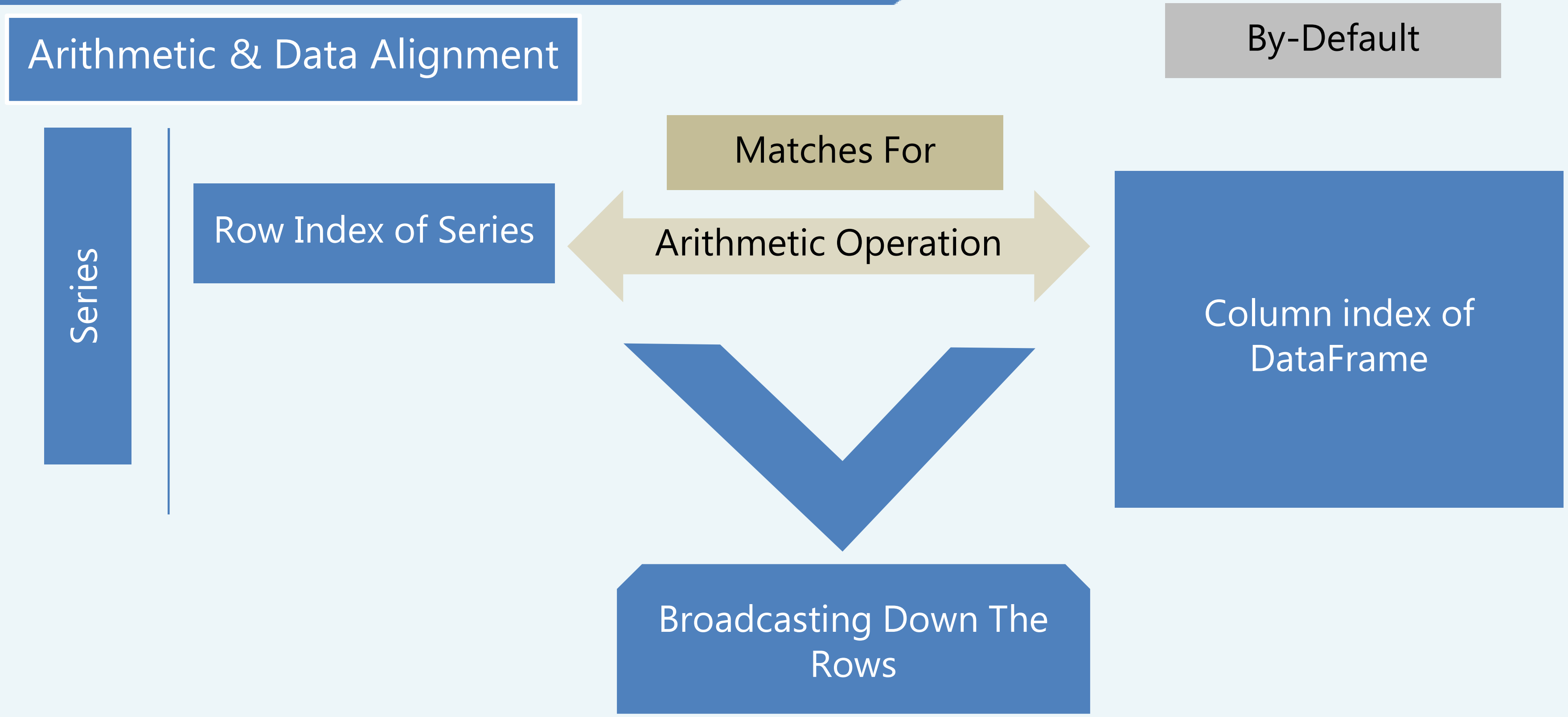
fill_value:
which is used to fill NaN values with different values and helps in data operation element by element wise

Essential Functionalities

Arithmetic & Data Alignment



Essential Functionalities



Arithmetic & Data Alignment

Series

Row Index of Series

Matches For

Arithmetic Operation

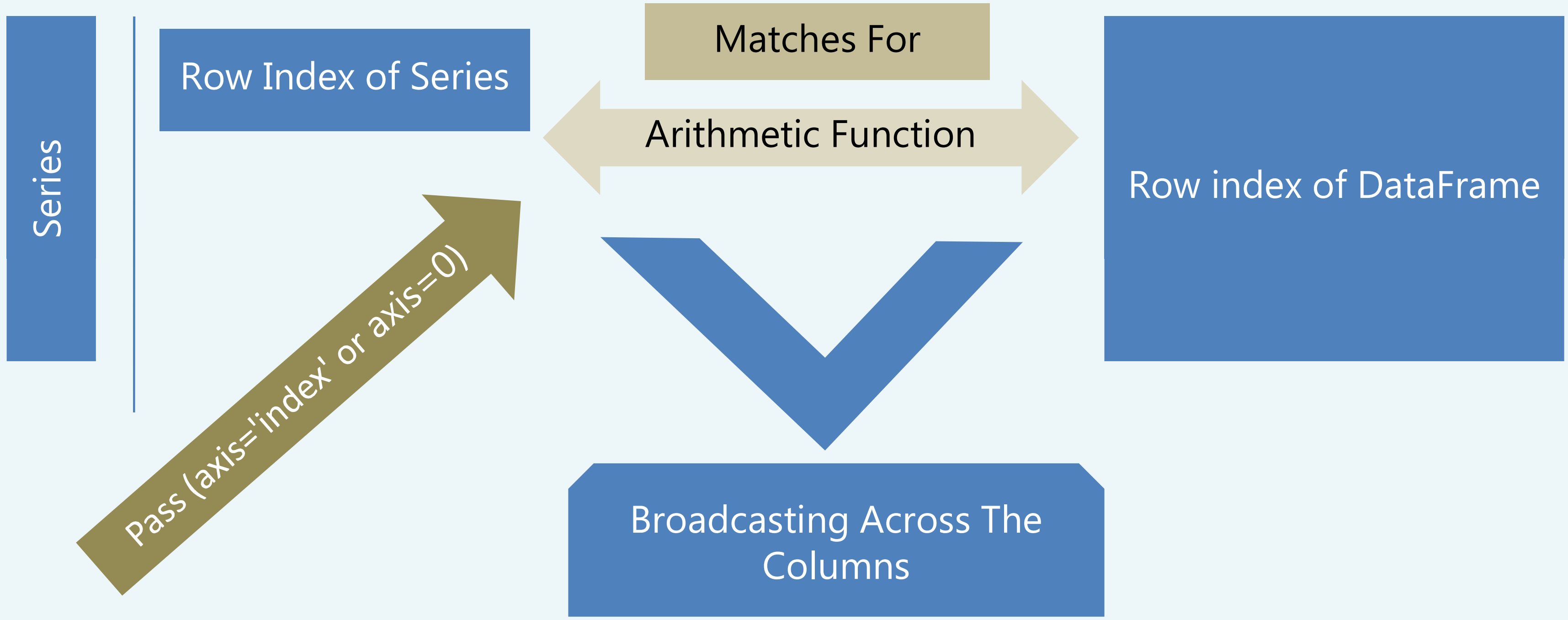
By-Default

Column index of DataFrame

Broadcasting Down The Rows

Essential Functionalities

Arithmetic & Data Alignment



Series

Row Index of Series

Pass (axis='index' or axis=0)

Matches For

Arithmetic Function

Row index of DataFrame

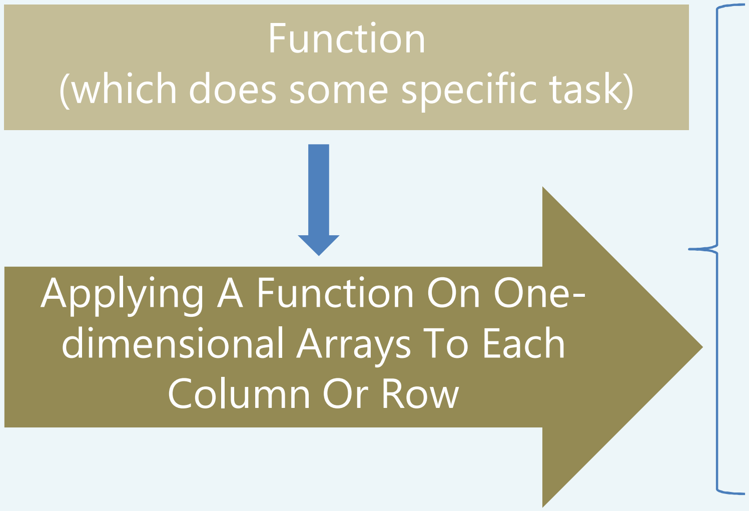
Broadcasting Across The Columns

Essential Functionalities

Function Application & Mapping

Pandas DataFrame Object

Numbers_df



Function
(which does some specific task)

Applying A Function On One-dimensional Arrays To Each Column Or Row

Column Index	**A**	**B**	**C**	**D**
Row Index				
Line1	5	6	1	6
Line2	4	5	6	8
Line3	1	2	3	4

Essential Functionalities

Function Application & Mapping

Pandas DataFrame Object

```
f = lambda x: x.max() - x.min()
```

Use: apply method

Numbers_df.apply(f)

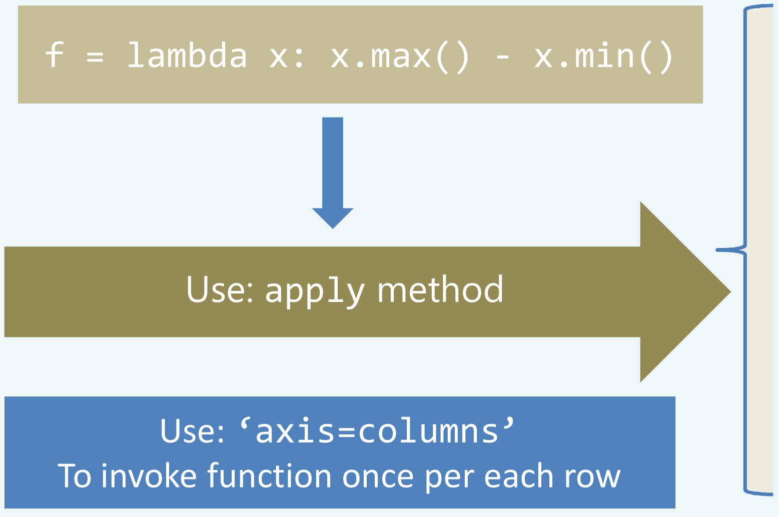
By default it applies on each column

Column Index	A	B	C	D
Row Index				
Line1	5	6	1	6
Line2	4	5	6	8
Line3	1	2	3	4

Essential Functionalities

Function Application & Mapping

Pandas DataFrame Object



`f = lambda x: x.max() - x.min()`

Use: apply method

Use: ‘axis=columns’
To invoke function once per each row

Column Index	A	B	C	D
Row Index				
Line1	5	6	1	6
Line2	4	5	6	8
Line3	1	2	3	4

```
Numbers_df.apply(f, axis=columns)
```

Essential Functionalities

Function Application & Mapping

The function passed to apply not only return a scalar value; but also return a Series with multiple values:

```
def f(x):
    return pd.Series([x.min(), x.max()],
                     index=['min', 'max'])
```

Use: apply method

Pandas DataFrame Object

Column Index	A	B	C	D
Row Index				
Line1	5	6	1	6
Line2	4	5	6	8
Line3	1	2	3	4

```
Numbers_df.apply(f, axis=columns)
```

Essential Functionalities

Function Application & Mapping

Pandas DataFrame Object

We can also use Python-Function on each Row Or Column:

Use: ‘applymap’ method

Column Index	A	B	C	D
Row Index				
Line1	5	6	1	6
Line2	4	5	6	8
Line3	1	2	3	4

```
Numbers_df.applymap(f)
Or
Numbers_df.applymap(f, axis=columns)
```

Essential Functionalities

Sorting & Ranking

Sorting is technique which returns a sorted object by index or values

Use: ‘‘sort_index’ method

which returns a new, sorted object:

- The returned object is in ascending order by default, so we can use ‘ascending = False’ to get the sorted object in descending order.
- Any missing values are sorted to the end of the Series by default:

Pandas Series Object

A	2
C	6
B	5
D	Nan
F	1
E	0

Essential Functionalities

Sorting & Ranking

Pandas DataFrame Object

Sorting is technique which returns a sorted object by index or values

Use: ‘‘sort_index’ method

Column Index	A	C	D	B
Row Index				
Line2	5	6	1	6
Line1	4	5	6	8
Line3	1	2	3	4

which returns a new, sorted object:

- By default it returns sorted object by Row index, we can use ‘axis=1 or column’ to sort the object by column.
- The returned object is in ascending order by default, so we can use ‘ascending = False’ to get the sorted object in descending order.

Essential Functionalities

Sorting & Ranking

Pandas DataFrame Object

Sorting is technique which returns a sorted object by index or values

Use: ‘‘sort_values’ with method

Column Index	A	C	D	B
Row Index				
Line2	5	6	1	6
Line1	4	5	6	8
Line3	1	2	3	4

which returns a new, sorted object:

- To sort values by columns pass ‘by=[column names]’ to ‘sort_values’ method
- The returned object is in ascending order by default, so we can use ‘ascending = False’ to get the sorted object in descending order.

Essential Functionalities

Sorting & Ranking

Pandas DataFrame Object

Column Index	A	C	D	B
Row Index				
Line2	5	6	1	6
Line1	4	5	6	8
Line3	1	2	3	4

Similarly the ‘rank’ method works:

- Ranking assigns ranks from one through the number of valid data points in an array.
- For Series and DataFrame, by default rank breaks ties by assigning each group the mean rank.

Essential Functionalities

Axis Indexing With Duplicate Values

It is used to check duplicated indices in the Row index in series.

The same logic applies for Rows of DataFrame

Use: ‘‘is_unique’ method

Pandas Series Object

A	2
C	6
A	5
D	Nan
D	1
E	0

Essential Functionalities

Let’s Move To The Practical Demo