

Shared Attributes

```
.head(n)
.tail(n)
.index
.values
.shape
.axes
.info()
```

Common attributes shared between pd.Series and pd.DataFrame

Null Unique Values

Drop Null Values

```
df.dropna (how= any/all, inplace = T/F)
df.dropna (subset = ['col_name1', 'col_name2'], how = any/all, inplace = T/F) - from specific columns
```

Fill Null Values

```
df.fillna (value, inplace = T/F) - acts on entire data frame.
df['col_name'].fillna (value = 'fill_value', inplace = T/F) - act on selected columns
```

Unique Values

```
df['col'].unique() - Returns unique values as a list(including null values)
df.nunique() - No of unique Values.(by default doesn't count null)
df.unique (dropna = False)
df['col'].nunique()
df['col'].unique(dropna = True/False)
```

Panda Series

```
pd.Series.hasnans
pds.nunique() - No of unique Values.
pdf.unique(dropna = True/False) - Returns unique values in a list.
```

Dealing Columns

```
df.columns -returns column names as a list
df['new_col'] = list /series
df.insert(loc = n, column= col_name, value= New_value /array) -inserts column at desired position / location
```

Sort, Rank, Count

Sort

```
df.sort_index(ascending = True/False) - Sort Index
df.sort_values( col_name, na_position = " First/ Last", ascending = True/False) - Sort based on values
df.sort_values( ['col_name', 'col2'], ascending = [True, False]) - Sort by Multiple Columns
```

Rank

```
df['col'].rank(ascending = True/False) - Ranks are assigned based on sorted values
```

counts

```
df.value_counts() - counts exact rows
df.value_counts(normalize = True)
```



By email2automate

Not published yet.
Last updated 3rd May, 2023.
Page 1 of 3.

Sponsored by [Readable.com](https://readable.com)
Measure your website readability!
<https://readable.com>

Sort, Rank, Count (cont)

```
pds.value_counts() - pd series
```

Data Type Conversions and Optimization

```
df['col'] = pd.to_datetime(df['col'])
or we can parse dates in import itself.
pd.read_csv(....., parse_dates = ['col1', 'col2'])
df['col'] = df['col'].astype(dtype)
dtype = bool, category, int
```

Filtering Data

Multiple Conditional Filtering

```
mask1 = df[col1] == 'value' - Returns True/False boolean series
mask2 = df[col2] <= 'value'
mask3 = df[col3] >= 'value'
df[(mask1 & mask2) | mask3]
```

Inclusion Check

```
mask = df['col'].isin(['val1', 'val2', 'val3']) - Check for inclusion using isin() method.
mask1 = df[col1] == 'val1' - isin method is equal to three conditional checks.
mask1 = df[col1] == 'val2'
mask1 = df[col1] == 'val3'
```

For NULL values

```
mask = df['col'].isnull() - Returns True/False boolean series
mask = df['col'].notnull() - Returns True/False boolean series
```

Inclusion Check within a range

```
mask = df['col'].between(val1, val2) -Returns True/False boolean series. True for values within range.
```

Duplicate Values

```
mask = df['col'].duplicated(keep = "first/last/false") - Returns boolean series, True for Duplicates
df.drop_duplicates() -Deletes duplicate from df. if applied on df complete row should be identical.
df.drop_duplicates(subset = ['col1', 'col2']) - Drops if the combination of col1 and col2 are identical.
```

Data Extraction

```
SET INDEX
RESET INDEX
LOC ACCESSOR
ILOC ACCESSOR
CODE
```



By email2automate

Not published yet.

Last updated 3rd May, 2023.

Page 3 of 3.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>

