

Loading Pandas

Import Pandas Module with the alias *pd*

```
import pandas as pd
```

Creating Dataframes From Files

From a csv file

```
df = pd.read_csv('file.csv')
```

From a python dictionary

```
df = pd.DataFrame.from_dict(<dict>)
```

Displaying Dataframe Info

Display first five rows in dataframe

```
df.head()
```

Display last five rows in dataframe

```
df.tail()
```

Show all column names

```
df.columns
```

Show all object types in dataframe

```
df.dtypes
```

Show statistics for all int and float columns

```
df.describe()
```

Show statistics for 'object' type columns

```
df.describe(include='object')
```

Show number of rows and columns

```
df.shape
```

Display True for each NaN value, False otherwise

```
df.isnull()
```

Display a table with the number of NaN values for each column

```
df.isnull().sum()
```

Updating

Delete all rows containing *NaN* values in the *df* Dataframe

```
df.dropna(inplace=True)
```

Delete '*col_name*' column

```
df.drop('col_name', axis=1)
```

Example of a calculated column

```
df['new_col'] = df['col_1'] + df['col_2']
```

Update the entire column to value *<value>*

```
df['new_col'] = <value>
```

Update the cell at (*a,b*) to *<value>*

```
df.iloc[a,b] = <value>
```

Update (or creates) '*col_a*' with the result of lambda function applied to '*col_b*'

```
df['col_a'] = df['col_b'].apply(<lambda function>)
```

Filtering Columns

Indexing with iloc

Displays the entire row indexed *n*

```
df.iloc[n]
```

Displays the element in row *n* & column *m*

```
df.iloc[n, m]
```

Displays a slice of rows: from row *a* to row *b*

```
df.iloc[a:b]
```

Displays rows *a* to *b* only in the columns *c* to *d*

```
df[a:b, c:d]
```

Indexing with loc

Shows all rows indexed with *<index>*

```
df.loc[<index>]
```

Manipulating Dataframes

Create a copy of the dataframe

```
new_df = df.copy()
```

Set '*column_name*' as the index

```
df.set_index('column_name', inplace=True)
```

Delete / Output

Output to *csv* file

```
df.to_csv('output.csv')
```

Output to *json* file

```
df.to_json()
```

Output to *html* file

```
df.to_html()
```

Delete a Dataframe

```
del df
```

Display an entire column as a series

```
df['column_name']
```

Display all columns in the given list

```
df[['col_1', 'col_2', ... 'col_n' ]]
```

Show all unique elements in 'column_name'

```
df['column_name'].unique()
```

Filtering Rows

Display all rows satisfying <condition>

```
df[<condition>]
```

Display all rows where df['column_name'] == <value>

```
df[df['column_name'] == <value>]
```

Show all rows satisfying both conditions

```
df[(<condition_1>) & (<condition_2>)]
```



By Cheto
cheatography.com/cheto/

Published 6th September, 2022.
Last updated 6th September, 2022.
Page 1 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>