

Import Data

`df = pd.read_csv('filename.csv')` Read CSV into a Pandas DataFrame

`df = pd.to_csv('filename.csv')` Export Pandas DataFrame to CSV

Import Options:

`header=False, Index=False, usecols=(5,6)`

Can also read CSV / HTML / Excel / JSON

Combine multiple files into one (1) DataFrame

`all_files = glob.glob('*.txt')` Finds all txt files in the

`df_raw = [pd.read_csv(f) for f in all_files]` Makes multiple DataFrames

`df_all = pd.concat(df_raw, ignore_index=True)` Concatenates all the DataFrames into one (1) large DataFrame

Select Data

`df.head(5)` Reads the first 5 rows

`df.tail(5)` Reads the last 5 rows

`df.shape()` Gives the number of columns and rows in the DataFrame

Select Row

`data.iloc[0]` First row of DataFrame

`data.iloc[1]` Second row of DataFrame

Select Column

`data.iloc[:,0]` First column of DataFrame

`data.iloc[:,1]` Second column of DataFrame

Select Column and Row Combined

`df.iloc[:3, :2]` Selecting first 3 rows and first 2 columns

`df.iloc[:3, ['Column1', 'Column2']]` Selecting first 3 rows and first 2 columns

Re-Order Columns

`df = df[['Column3', 'Column2', 'Column1']]` Re-orders the columns to the order specified in this list

Drop Columns

`df = df.drop(columns=['Column1'], axis=1)` Drops 'Column1_Name' from DataFrame

Sort Columns

`df = df.sort_values(by=['Column1'], ascending=False)` Sort values by Column1

Filter Column

`df = df[(df['Column1'] >= some_number)]` Filter DataFrame by certain value

Rename Columns

`df.columns = ['A', 'B']` Renamed Columns to 'A' and 'B'

Merge DataFrames

`df1.append(df2)` Joins df1 and df2

Filter on Condition

`df = df[(df > 2).all(axis=1)]` Removes any values less than 2

Select Row Based on Condition

`row = df[df.A > 3].iloc[0]` Select first row where A > 0

Dealing with NAN values

`df = df.fillna(method='ffill')` Fills blank values using forward fill method

`df = df.fillna(method='bfill')` Fills blank values using backwards fill method

`df.dropna(inplace=True)` Removes rows with no values

Padding Values With Zero's

`df['Column1'] = df['Column1'].astype(str).str.zfill(6)` Sets the number to six (6) long, which adds zeros

Change Column Data Type

`df['Column1'] = df['Column1'].astype(float)` Change 'Column1' to float



By **dstark0011**

Published 27th November, 2020.

Last updated 27th November, 2020.

Page 1 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Convert Column to Date / Time

```
df['Time'] = df['Time'].apply(p-  
d.to_datetime)
```

Converts the time column to a
Datetime Series



By **dstark0011**

cheatography.com/dstark0011/

Published 27th November, 2020.

Last updated 27th November, 2020.

Page 2 of 2.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>