Pandas cheat sheet Cheat Sheet

by monika.g via cheatography.com/175674/cs/36874/

To Start

```
import numpy as np
import pandas as pd
```

Create

`pd.DataFrame( dict/list, index = None, columns = None)`	create DataFrame from list or dictionary
`df.ind ex[ names]`	set custom indexes
`pd.Series( list/np_array/dict, index = None)`	create series from the list or np_array or dictionary

Input and Output

`pd.rea d_c sv( 'name', index_col = None)`	read csv
`pd.rea d_e xce l(' name' )`	read excel
`df.to_ csv ('n ame', index = False)`	save to csv
`df.to_ exc el( 'name', 'sheet_name = 'name', index = False)`	save to excel

Iteration

```
for lab, row in df.iterrrows():
  pri nt(lab)
  pri nt(row)
```

Functions/Methods

`s.drop (ro w_i ndex, axis = 0 )`	drop values from rows of series
`df.dro p(c ol_ name, axis = 0 )`	drop values from columns
`df.dro p(c olu mns =[c ol_ na mes])`	drop columns from DataFrame
`df.dro p_d upl ica tes()`	remove duplicate rows (only considers columns)
`df.sor t_i ndex(by = col_name s)`	sort by the values along an axis
`df.sort_values( by = col_names, ascending = False)`	order rows by values of a column high to low
`df.ren ame (co lumns = {'old_name':'new_name'})`	rename the columns of a DataFrame
`df.rank()`	assign ranks to entries
`pd.con cat ([d f1, df2])`	append rows of DataFrames
`len(df)`	number of rows in DataFrame
`df1.jo in (df2)`	join two DataFrames
`df['co l_n ame '].u ni que()`	return unique values from column
`df[col_name].apply( func/type.method)`	apply function to column
`df.app ly ( f un c/t yp e. m e thod)`	apply function

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

Cheatography

Pandas cheat sheet Cheat Sheet

by monika.g via cheatography.com/175674/cs/36874/

Extract

`df[col _names]`	series of column
`df[[co l_n ames]]`	select column
`df[sta rt:end]`	select many columns
`df.loc [in dex _name]`	select row
`df.ilo c[i nde x_num]`	select rows
`df.loc [[i nde x_n ames]]`	select rows
`df.ilo c[[ ind ex_ nums]]`	select rows
`df.loc [[i nde x_n ames], [col_names]]`	select rows and columns
`df.loc[:, [col_n ames]]`	select all rows and few columns
`df.iloc[:, [col_n ums]]`	select all rows and few columns
`df.head(n)`	select first n rows
`df.tail(n)`	select last n rows
`df.fil ter (regex = 'regex ')`	select columns whose name matches regular expression regex

Boolean Operators

`df[np.logica l_and( con1, con2, ...)]`	1 'and' 2 condition ...
`df.loc [con1 & con2]`	1 'and' 2 condition ...
`df[np.logical_or( con1, con2, ...)]`	1 'or' 2 condition ...
`df.loc [con1 | con2]`	1 'or' 2 condition ...
`df[np.logica l_not(con)]`	'not' condition
`df.loc[ ~ con1]`	'not' condition
`df[var]`	for condition

Get DataFrame Information

`df.shape`	(rows, colums)
`df.index`	decribe index
`df.columns`	describe DataFrame columns
`df.info()`	info on DataFrame
`df.count()`	number of non_NA values for columns
`df.des cribe()`	summary statics

Math

`df.sum()`	sum of values for columns
`df.cum sum()`	cummulative sum of values
`df.min()` `df.max()`	minimum values for columns maximum values for columns
`df.med ian()`	median of values columns
`df.mean()`	mean of values for columns
`df.std()`	standard deviation of each object
`df.var()`	variance of each object

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