Cheatography

Pandas Cheat Sheet

by Ianh via cheatography.com/85121/cs/20025/

Show installed versions

pd.__versions__	Python version
pd.show_versions()	Dependency & versions

Create an example DataFrame

df = pd.DataFrame({'col one':[100, 200], 'col two':[300, 400]})	Dictionary method
pd.DataFrame(np.random.rand(4, 8), columns=list('abcdefgh'))	Rand method

Rename columns

df = df.rename({'col one':'col_one', 'col two':'col_two'}, axis='columns')	Overwrite old names (keys) with new names (values)
df.columns = ['col_one', 'col_two']	Rename all of the columns at once
df.add_prefix('X_')	Add a prefix
df.add_suffix('_Y')	Add a suffix

Reverse row order

drinks.loc[::-1].head()	Reverse only
drinks.loc[::-1].reset_index(drop=True).head()	Reverse and reset index

Reverse column order

drinks.loc[:, ::-1].head()	Reverse the left-to-right order of your columns

Select columns by data type

drinks.select_dtypes(include='number')	To select only the numeric columns
drinks.select_dtypes(include=['number', 'object', 'category', 'datetime'])	Include multiple data types by passing a list
drinks.select_dtypes(exclude='number')	Exclude certain data types

Convert strings to numbers

df.astype({'col_one':'float', 'col_two':'float'}).dtypes	To do mathematical operations on these columns, we need to convert the data types to numeric. This will fail if there are '-' or NAN
pd.to_numeric(df.col_three, errors='coerce').fillna(0)	If you know that the NaN values actually represent zeros, you can fill them with zeros using the fillna() method
df = df.apply(pd.to_numeric, errors='coerce').fillna(0) df	you can apply this function to the entire DataFrame all at once by using the apply() method

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