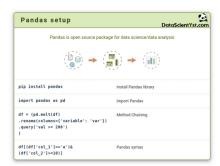
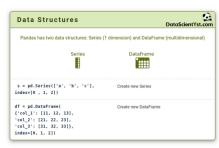
## **Pandas Cheatsheet**

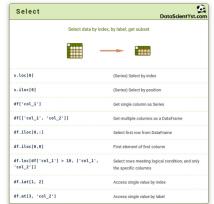


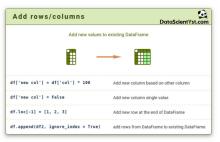


Read	DataScientYst.con
Import data from CSV, Exc	el, JSON, SQL, HTML, web
	<b>→</b> (((i))
pd.read_csv(filename)	From a CSV file
pd.read_csv(filename, header=None, nrows=5)	From a CSV file with parameters
pd.read_excel(filename)	From an Excel file
pd.read_sql(query, connection_object)	Reads from a SQL table/database
pd.read json(json string)	Reads from a JSON formatted string, URL or file.

Write	DataScientYst.co
Write dat	a to CSV, Excel, JSON, HTML
Wille dat	a to GSV, EXCE, GSGN, FTML
	(2 0)
df.to_csv(filename)	Writes to a CSV file
df.to_excel(filename)	Writes to an Excel file
df.to_json(filename)	Writes to a file in JSON format
di.to_json(izcename)	writes to a me in JSON format

Inspect Data	DataScientYst.com
View stats, samp	les and summary of the data
	describe() count 100 mean 5 std 10
df.head(n)	First n rows
df.tail(n)	Last n rows
df.shape	Number of rows and columns
Mr. Anrwey	тоех, разатуре ани метогу птоппалон
df.describe()	Summary statistics for numerical columns
s.value_counts(dropna=False)	(Series) Views unique values and counts
df.sample(n)	Randomly select n rows.
df.nlargest(n, 'col_1')	Select and order top n entries for column
df.nsmallest(n. 'col 1')	Select and order hottom n entries
df.quantile([0.25,0.75])	Quantiles of each object



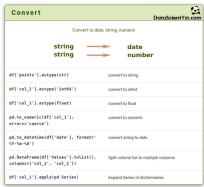




Sort values/index	DataScientYst.com
Sort and rank values/index by one or multiple criteria	
	<b>→</b>
<pre>df.sort_values(by='col_1', ascending=False)</pre>	Sort values by column, ascending order
df.sort_values(by=['col_1', 'col_2'])	Sort values by columns
df.sort_index(ascending=False)	Sort object by labels (along an axis) in descending order
<pre>df.sort_values(by=[('col_1', 'col_2')])</pre>	Sort multindex by multiple levels
df.reset_index()	Reset the index of the DataFrame, moving index to columns

Filter	DataScientYst.cor
Filter data based o	on multiple criteria
quer	y0 ••••
df[df['col_1'] > 100]	Values greater than X
df[(df['col_1']=='a')& (df['col_2']>=10)]	Filter Multiple Conditions - & - and;   - or
df[df['date'] > '2022-02-22']	Date filtering
df[df['date'].dt.month == 2]	Filter with dt attributes
<pre>df[df['col_1'].str.contains('pan*', regex=True)]</pre>	Filter by regex
df[df['col_1'].isin(['pan', 'das'])]	Filter based on list of values
df.query('col_1 > 100')	Filter by queries
df.query('col 1 > 100 and col 2 = 0')	Filter by multiple queries





Merge & Concat	DataScientYst.co
Merging, joniing and concat	enating 2 and more DataFrames
(🗐) —	<b>→</b> (iii)
dfl.append(df2)	Adde the rowe in df1 to the end of df2 (columns
dfl.append(df2)	Adds the rows in df1 to the end of df2 (columns should be identical)
dfl.append(df2) pd.concat([df1, df2],axis=1)	

