















in linkedin.com/in/romansyasetyo/





# **Table of Content**What will We Learn Today?

- 1. About Pandas
- 2. About DataFrame
- 3. Creating DataFrame
- 4. Read DataFrame
- 5. Export DataFrame

**Hands on using Python** 









# **Google Colab**

https://colab.research.google.com/







# **About Pandas**







- Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language
- Tools for **reading and writing data** between in-memory data structures and different formats: CSV and text files, Microsoft Excel, SQL databases, and the fast HDF5 format







#### **About DataFrame**

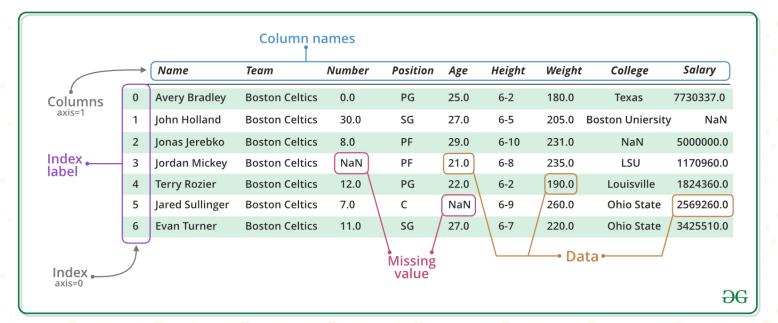






#### **About DataFrame**

- Main object in data analysis with Python
- Represent data with rows & columns





Source: GeeksforGeeks











```
# Method 1: Creating Pandas DataFrame from lists of lists.

# initialize list of lists
data_1 = [['tom', 10], ['nick', 15], ['juli', 14]]

# Create the pandas DataFrame
df_1 = pd.DataFrame(data_1, columns = ['Name', 'Age'])

# print dataframe.
df_1
```

	Name	Age
0	tom	10
1	nick	15
2	juli	14



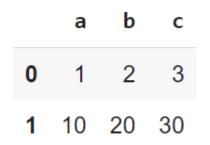




	Name	Age	
0	Tom	20	
1	nick	21	
2	krish	19	
3	jack	18	,
		٦٦	













```
# Method 4: Creating DataFrame using zip() function.
# Python program to demonstrate creating
# pandas Datadaframe from lists using zip.
# List1
Name = ['tom', 'krish', 'nick', 'juli']
# List2
Age = [25, 30, 26, 22]
# get the list of tuples from two lists.
# and merge them by using zip().
data_4 = list(zip(Name, Age))
# Converting lists of tuples into
# pandas Dataframe.
df_4 = pd.DataFrame(data_4,
        columns = ['Name', 'Age'])
# Print data.
df_4
```

	Name	Age
0	tom	25
1	krish	30
2	nick	26
3	juli	22







#### **Read DataFrame**







#### Read DataFrame

 Pandas is a strong library since it can get data from many sources, for example CSV, TSV, Excel, JSON, HTML, etc.
 Getting data from external is pretty straightforward, the example can be seen at following code:

```
df = pd.read_csv('filepath.type', delimiter=',')
df = pd.read_excel('filepath.type', sheet_name=0, header=0, names=None, index_col=None)
```







# **Export DataFrame**







#### **Export DataFrame**

• When we have processed the DataFrame and want to export it to external file, pandas have covered the functions for us:

```
# Export to CSV
df.to_csv('filepath.csv', index=False, delimiter=';')
# Export to Excel
df.to_excel('filepath.xlsx', sheet_name='Sheet1', index=False)
```





# Thank You

