

Creating and Accessing Pandas DataFrames	
Course Code: CPE 031	Program: Computer Engineering
Course Title: Visualization and Data Analysis	Date Performed: 10-08-2024
Section: CpE21 S4	Date Submitted: 10-08-2024
Name: BONIFACIO, NYKO ADREIN L.	Instructor: Prof. Sayo
Intended Learning Outcomes (ILO): By the end of this laboratory session, learners will be able to <ul style="list-style-type: none"> - Construct and manipulate Pandas DataFrames from various data structures (such as lists, dictionaries, and NumPy arrays) while demonstrating an understanding of DataFrame attributes and methods. This includes loading the dataset, creating DataFrames with appropriate column labels and accessing data from rows and columns. 	
Instructions: <ol style="list-style-type: none"> 1. Loading your dataset: Refer back to your chosen dataset from the PRELIM period. Whether you downloaded it or stored it in your Google Drive, you are required to load it into the Google Colab. Watch this video to learn more about how to read CSV files in Google Colab. (Take a screenshot to document successful execution.) 2. Creating a dataframe from your CSV file: Once you have successfully loaded your dataset, you need to create a dataframe from your uploaded CSV file. (Take a screenshot to document successful execution.) 3. Creating a dataframe from a dictionary of lists: Manually create a dictionary where each value is composed of a list from your original dataset, then load it into a dataframe, before printing it. You are required to provide at least five (5) observations in your list. (Take a screenshot to document successful execution.) 4. Creating a dataframe from a list of dictionaries: Manually create a list of dictionaries from your original dataset, then pass it into a dataframe, before printing it. You are required to provide at least five (5) observations in your list. (Take a screenshot to document successful execution.) 5. Selecting dataframe columns: Execute a method that would allow you to select a single and multiple dataframe columns. (Take a screenshot to document successful execution.) 6. Selecting dataframe rows: Execute a method that would allow you to select a single and multiple dataframe rows using panda indexing and python indexing. 	

Output:

1.

```
[20] import pandas as pd

path = "/content/ComputerData.csv"
df = pd.read_csv(path)
```

2.

df.head()

	Computer ID	Brand	Model	Processor	RAM (GB)	Storage (GB)	Graphics Card	Price (\$)
0	1	Dell	XPS 13	Intel i7	16	512	Intel Iris Xe	1200
1	2	HP	Pavilion 15	AMD Ryzen 5	8	256	AMD Radeon	800
2	3	Apple	MacBook Pro	Apple M1	16	1024	Apple Integrated	2400
3	4	Lenovo	ThinkPad X1	Intel i9	32	512	NVIDIA GTX 1650	1800
4	5	Asus	ZenBook 14	Intel i5	16	512	NVIDIA MX250	1000

3.

```
data_dict = {
    'Computer ID': [1, 2, 3, 4, 5],
    'Brand': ['Dell', 'HP', 'Apple', 'Lenovo', 'Asus'],
    'Model': ['XPS 13', 'Pavilion 15', 'MacBook Pro', 'ThinkPad X1', 'ZenBook 14'],
    'Processor': ['Intel i7', 'AMD Ryzen 5', 'Apple M1', 'Intel i9', 'Intel i5'],
    'RAM (GB)': [16, 8, 16, 32, 16],
    'Price ($)': [1200, 800, 2400, 1800, 1000]
}

df_from_dict = pd.DataFrame(data_dict)

df_from_dict
```

	Computer ID	Brand	Model	Processor	RAM (GB)	Price (\$)
0	1	Dell	XPS 13	Intel i7	16	1200
1	2	HP	Pavilion 15	AMD Ryzen 5	8	800
2	3	Apple	MacBook Pro	Apple M1	16	2400
3	4	Lenovo	ThinkPad X1	Intel i9	32	1800
4	5	Asus	ZenBook 14	Intel i5	16	1000

4.

```
data_list = [
    {'Computer ID': 1, 'Brand': 'Dell', 'Model': 'XPS 13', 'Price ($)': 1200},
    {'Computer ID': 2, 'Brand': 'HP', 'Model': 'Pavilion 15', 'Price ($)': 800},
    {'Computer ID': 3, 'Brand': 'Apple', 'Model': 'MacBook Pro', 'Price ($)': 2400},
    {'Computer ID': 4, 'Brand': 'Lenovo', 'Model': 'ThinkPad X1', 'Price ($)': 1800},
    {'Computer ID': 5, 'Brand': 'Asus', 'Model': 'ZenBook 14', 'Price ($)': 1000}
]

df_from_list = pd.DataFrame(data_list)

df_from_list
```

	Computer ID	Brand	Model	Price (\$)
0	1	Dell	XPS 13	1200
1	2	HP	Pavilion 15	800
2	3	Apple	MacBook Pro	2400
3	4	Lenovo	ThinkPad X1	1800
4	5	Asus	ZenBook 14	1000

5.

```
single_column = df['Brand']

single_column
```

	Brand
0	Dell
1	HP
2	Apple
3	Lenovo
4	Asus

dtype: object

0s `multiple_columns = df[['Brand', 'Model', 'Price ($)']]`

`multiple_columns`

	Brand	Model	Price (\$)
0	Dell	XPS 13	1200
1	HP	Pavilion 15	800
2	Apple	MacBook Pro	2400
3	Lenovo	ThinkPad X1	1800
4	Asus	ZenBook 14	1000

6.

USING PANDAS

0s `single_row = df.loc[0]`

`single_row`

	0
Computer ID	1
Brand	Dell
Model	XPS 13
Processor	Intel i7
RAM (GB)	16
Storage (GB)	512
Graphics Card	Intel Iris Xe
Price (\$)	1200

`dtype: object`

0s [42] `multiple_rows = df.iloc[0:3]`

`multiple_rows`

	Computer ID	Brand	Model	Processor	RAM (GB)	Storage (GB)	Graphics Card	Price (\$)
0	1	Dell	XPS 13	Intel i7	16	512	Intel Iris Xe	1200
1	2	HP	Pavilion 15	AMD Ryzen 5	8	256	AMD Radeon	800
2	3	Apple	MacBook Pro	Apple M1	16	1024	Apple Integrated	2400

USING PYTHON

```
[43] first_row = df.iloc[0]

first_row
```



	0
Computer ID	1
Brand	Dell
Model	XPS 13
Processor	Intel i7
RAM (GB)	16
Storage (GB)	512
Graphics Card	Intel Iris Xe
Price (\$)	1200

dtype: object

```
first_three_rows = df.iloc[0:3]

first_three_rows
```



	Computer ID	Brand	Model	Processor	RAM (GB)	Storage (GB)	Graphics Card	Price (\$)	
0	1	Dell	XPS 13	Intel i7	16	512	Intel Iris Xe	1200	
1	2	HP	Pavilion 15	AMD Ryzen 5	8	256	AMD Radeon	800	
2	3	Apple	MacBook Pro	Apple M1	16	1024	Apple Integrated	2400	