



Task 3 - DataFrames

1. What is Dataframe?

A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

2. Applications of Dataframe

Used to efficiently manage and analyze large datasets of financial transactions, stock market data, and economic indicators to make informed investment decisions

3. Creating Dataframe

```
import pandas as pd
data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}
df = pd.DataFrame(data)
print(df)
```

```
main.py > ...
1 # 1. Creating Dataframe
2
3 import pandas as pd
4 data = {
5     "calories": [420, 380, 390],
6     "duration": [50, 40, 45]
7 }
8 df = pd.DataFrame(data)
9 print(df)
10
```

	calories	duration
0	420	50
1	380	40
2	390	45

PS C:\Users\De11\Desktop\brushup\XML\Day3Internship>

4. Add, remove operation with rows and columns

a. Add Operation with Rows

```
import pandas as pd
dict = {'Name':['Martha', 'Tim', 'Rob', 'Georgia'],
        'Maths':[87, 91, 97, 95],
```

```

        'Science':[83, 99, 84, 76]
    }
    df = pd.DataFrame(dict)
    print(df)
    df.loc[len(df.index)] = ['Amy', 89, 93]
    print(df)

```

The screenshot shows a code editor with a file named 'main.py'. The code defines a dictionary with columns 'Name', 'Maths', and 'Science', creates a DataFrame, and then adds a new row for 'Amy'. The terminal output shows the DataFrame before and after the row is added.

```

main.py > ...
11
12 #2(a). Add Operation with Rows
13
14 import pandas as pd
15 dict = {'Name': ['Martha', 'Tim', 'Rob', 'Georgia'],
16         'Maths': [87, 91, 97, 95],
17         'Science': [83, 99, 84, 76]}
18
19 df = pd.DataFrame(dict)
20 print(df)
21 df.loc[len(df.index)] = ['Amy', 89, 93]
22 print(df)
23
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TER
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship> python -i
Name Maths Science
0 Martha 87 83
1 Tim 91 99
2 Rob 97 84
3 Georgia 95 76
Name Maths Science
0 Martha 87 83
1 Tim 91 99
2 Rob 97 84
3 Georgia 95 76
4 Amy 89 93
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship>

```

b. Add Operation with Columns

```

import pandas as pd
data = {'Name': ['Jai', 'Princi', 'Gaurav', 'Anuj'],
        'Height': [5.1, 6.2, 5.1, 5.2],
        'Qualification': ['Msc', 'MA', 'Msc', 'Msc']}
df = pd.DataFrame(data)
address = ['Delhi', 'Bangalore', 'Chennai', 'Patna']
df['Address'] = address
print(df)

```

The screenshot shows a code editor with a file named 'main.py'. The code defines a dictionary with columns 'Name', 'Height', and 'Qualification', creates a DataFrame, and then adds a new column 'Address'. The terminal output shows the DataFrame after the column is added.

```

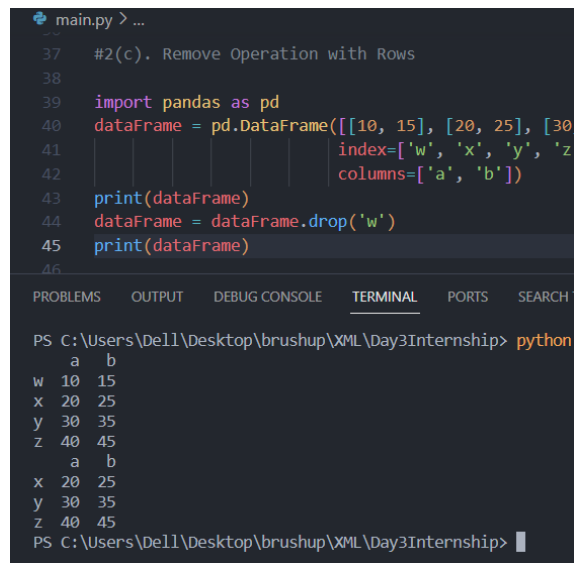
main.py > ...
24
25 #2(b). Add Operation with Columns
26
27 import pandas as pd
28 data = {'Name': ['Jai', 'Princi', 'Gaurav', 'Anuj'],
29         'Height': [5.1, 6.2, 5.1, 5.2],
30         'Qualification': ['Msc', 'MA', 'Msc', 'Msc']}
31 df = pd.DataFrame(data)
32 address = ['Delhi', 'Bangalore', 'Chennai', 'Patna']
33 df['Address'] = address
34 print(df)
35
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OU
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship> python -u "c:\Us
Name Height Qualification Address
0 Jai 5.1 Msc Delhi
1 Princi 6.2 MA Bangalore
2 Gaurav 5.1 Msc Chennai
3 Anuj 5.2 Msc Patna
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship>

```

c. Remove Operation with Rows

```
import pandas as pd
dataFrame = pd.DataFrame([[10, 15], [20, 25], [30, 35], [40, 45]],
                          index=['w', 'x', 'y', 'z'],
                          columns=['a', 'b'])

print(dataFrame)
dataFrame = dataFrame.drop('w')
print(dataFrame)
```



The screenshot shows a code editor with a file named 'main.py'. The code implements the removal of a row from a pandas DataFrame. It starts with a comment '#2(c). Remove Operation with Rows'. The code then imports pandas as 'pd' and creates a DataFrame with two columns, 'a' and 'b', and four rows indexed 'w', 'x', 'y', and 'z'. The values for column 'a' are 10, 20, 30, and 40, and for column 'b' are 15, 25, 35, and 45. The code prints the DataFrame, then uses the 'drop' method to remove the row with index 'w', and prints the resulting DataFrame. The output shows the DataFrame with the first row removed.

```
main.py > ...
37 #2(c). Remove Operation with Rows
38
39 import pandas as pd
40 dataFrame = pd.DataFrame([[10, 15], [20, 25], [30, 35], [40, 45]],
41                          index=['w', 'x', 'y', 'z'],
42                          columns=['a', 'b'])
43 print(dataFrame)
44 dataFrame = dataFrame.drop('w')
45 print(dataFrame)
46
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH

```
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship> python
a b
w 10 15
x 20 25
y 30 35
z 40 45
a b
x 20 25
y 30 35
z 40 45
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship>
```

d. Remove Operation with Column

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie'],
        'age': [25, 30, 35],
        'gender': ['F', 'M', 'M']}

df = pd.DataFrame(data)
print('Original DataFrame:\n', df)

#to delete columns
df = df.drop(columns=['gender'])

#to delete rows
# df = df[df['age'] != 30]
print('Modified DataFrame:\n', df)
```

```
main.py > ...
47 #2(d) Remove Operation with Column
48
49 import pandas as pd
50 data = {'name': ['Alice', 'Bob', 'Charlie'],
51         'age': [25, 30, 35],
52         'gender': ['F', 'M', 'M']}
53
54 df = pd.DataFrame(data)
55 print('Original DataFrame:\n', df)
56 #to delete columns
57 df = df.drop(columns=['gender'])
58 #to delete rows
59 # df = df[df['age'] != 30]
60 print('Modified DataFrame:\n', df)
61
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS SEARCH TERMINAL OUTPUT

```
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship> python -u "c:\Users\Dell\Desktop\brushup\XML\Day3Internship\main.py"
Original DataFrame:
   name  age gender
0  Alice   25     F
1   Bob   30     M
2 Charlie   35     M
Modified DataFrame:
   name  age
0  Alice   25
1   Bob   30
2 Charlie   35
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship>
```

5. Indexing the data

```
import pandas as pd
data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}
df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
print(df)
```

```
main.py > ...
63 #3. Indexing the data
64
65 import pandas as pd
66 data = {
67     "calories": [420, 380, 390],
68     "duration": [50, 40, 45]
69 }
70 df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
71 print(df)
72
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS SEARCH TERMINAL OUTPUT

```
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship> python -u "c:\Users\Dell\Desktop\brushup\XML\Day3Internship\main.py"
calories  duration
day1      420      50
day2      380      40
day3      390      45
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship>
```

6. Selecting the data

```
import pandas as pd
data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}
df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
```

```
print(df)
print(df.loc[["day1", "day3"]])
```

```
main.py > ...
73 #4. Selecting the data
74
75 import pandas as pd
76 data = {
77     "calories": [420, 380, 390],
78     "duration": [50, 40, 45]
79 }
80 df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
81 print(df)
82 print(df.loc[["day1", "day3"]])
83
```

	calories	duration
day1	420	50
day2	380	40
day3	390	45

```
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship> python -u "c:\Users\
calories duration
day1 420 50
day2 380 40
day3 390 45
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship>
```

7. Handling Missing Data

```
import pandas as pd
df = pd.read_csv('data1.csv')

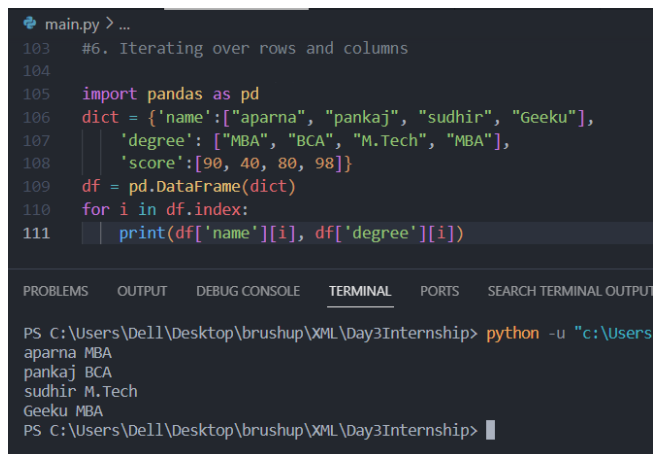
new_df = df.dropna()
print(new_df.to_string())
#OR
df.dropna(inplace = True) #Remove all rows with NULL values
print(df.to_string())
#OR
df.fillna(130, inplace = True) #Replace NULL values with the number 130
print(df.to_string())
#OR
df["Calories"].fillna(130, inplace = True) #Replace NULL values with the number 130 in calories column
print(df.to_string())
```

```
main.py > ...
84 #5(a). Handling Missing Data- remove rows that contain empty cells
85
86 import pandas as pd
87 df = pd.read_csv('data1.csv')
88
89 # new_df = df.dropna()
90 # print(new_df.to_string())
91 # OR
92 # df.dropna(inplace = True) #Remove all rows with NULL values
93 # print(df.to_string())
94 # OR
95 # df.fillna(130, inplace = True) #Replace NULL values with the number 130
96 # print(df.to_string())
97 #OR
98 df["calories"].fillna(130, inplace = True) #Replace NULL values with the number 130 in calories column
99 print(df.to_string())
100
101 #6. Iterating over rows and columns
```

0	60	'2020/12/01'	110	130	489.4
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340.0
3	45	'2020/12/04'	109	175	282.4
4	45	'2020/12/05'	117	148	486.0
5	60	'2020/12/06'	102	127	300.0
6	60	'2020/12/07'	110	136	374.0
7	450	'2020/12/08'	104	134	253.3
8	30	'2020/12/09'	109	133	199.1
9	60	'2020/12/10'	98	124	269.0
10	60	'2020/12/11'	103	147	329.3
11	60	'2020/12/12'	100	120	250.7
12	60	'2020/12/12'	100	120	250.7
13	60	'2020/12/13'	106	128	345.3
14	60	'2020/12/14'	104	132	379.3
15	60	'2020/12/15'	98	123	275.0
16	60	'2020/12/16'	98	120	215.2
17	60	'2020/12/17'	100	120	300.0
18	45	'2020/12/18'	90	112	130.0

8. Iterating over rows and columns

```
import pandas as pd
dict = {'name': ["aparna", "pankaj", "sudhir", "Geeku"],
        'degree': ["MBA", "BCA", "M.Tech", "MBA"],
        'score': [90, 40, 80, 98]}
df = pd.DataFrame(dict)
for i in df.index:
    print(df['name'][i], df['degree'][i])
```



```
main.py > ...
103 #6. Iterating over rows and columns
104
105 import pandas as pd
106 dict = {'name': ["aparna", "pankaj", "sudhir", "Geeku"],
107         'degree': ["MBA", "BCA", "M.Tech", "MBA"],
108         'score': [90, 40, 80, 98]}
109 df = pd.DataFrame(dict)
110 for i in df.index:
111     print(df['name'][i], df['degree'][i])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OUTPUT

```
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship> python -u "c:\Users
aparna MBA
pankaj BCA
sudhir M.Tech
Geeku MBA
PS C:\Users\Dell\Desktop\brushup\XML\Day3Internship>
```

Team 2

Abhijna

Anusha S

Varshaa Shetty

Shalini L