

## ▼ Practical-5

### ▼ Aim :

1. Write following program on Pandas DataFrame: Create an array "rank" with 5 element (rank1, rank2,...,rank5). Create and display a DataFrame "exam" from a specified dictionary "exam\_data\_array" with "rank" as label. Also display a summary of basic information and its data. Perform following operations on DataFrame "exam":
  1. Select the rows where the score is between 15 and 20 (inclusive).
  2. Sort the data first by "score" in ascending order, then by "name" in descending order.
  3. Replace the 'yes' and 'no' values from column "qualify" with True and False.
  4. Display specified columns (columns: 2 and 4) and rows (row: 1,3 and 5).
  5. Select the rows where number of attempts in the examination is less than 2 and score greater than 15.
  6. Change the name 'James' to 'Suresh' in "name" column of the data frame.
  7. Calculate the sum of the examination attempts by the students
  8. Append one row.
  9. Insert a new column "exam\_name" and then Delete the "exam\_name" column.
  10. Convert a NumPy array, dictionary and first column of a DataFrame to a series

```
import numpy as np
import pandas as pd
from termcolor import colored

class color:
    BOLD = '\033[1m'
    END = '\033[0m'

print("12002040701067")
def Seperator():
    print("-----")

rank = np.array(['rank1', 'rank2', 'rank3', 'rank4', 'rank5', 'rank6'])
Exam_Data_Array = {"Name": ["Aeshwary", "Dhruval", "Harshad", "Hitendra", "Hunaid", "James"],
                    "Score": [17, 35, 65, 90, 10, 27],
                    "Attempts": [1, 1, 1, 1, 1, 2],
                    "Qualify": ["No", "Yes", "Yes", "Yes", "No", "No"]}
exam = pd.DataFrame(data=Exam_Data_Array, index=rank, columns=Exam_Data_Array.keys())
#1
print(color.BOLD+"1.Rows where score between 15 and 20 (inclusive):\n"+color.END)
print(exam[exam['Score'].between(15, 20)])
Seperator()

#2
print(color.BOLD+"2.Sort the data frame first by 'score' in ascending order :\n"+color.END)
print(exam.sort_values(by=['Score'], ascending=[True]))
print(color.BOLD+"\n Sort the data frame first by 'name' in descending order :\n"+color.END)
print(exam.sort_values(by=['Name'], ascending=[False]))
```

```
#3
print(color.BOLD+"3.Replace the 'qualify' column contains the values"+color.END)
print(color.BOLD+" 'yes' and 'no' with True and False :\n"+color.END)
exam['Qualify'] = exam['Qualify'].map({'Yes': True, 'No': False})
print(exam)
Seperator()

#4
print(color.BOLD+"4.Select specific columns(columns: 2) & rows(row: 1,3 and 4):\n"+color.END)
print(exam.iloc[[1, 3, 4], [2]])
Seperator()

#5
print(color.BOLD+"5.Number of attempts in the examination"+color.END)
print(color.BOLD+" is less than 2 and score greater than 15 :\n"+color.END)
print(exam[(exam['Attempts'] < 2) & (exam['Score'] > 15)])
Seperator()

#6
print(color.BOLD+"6.Change the name 'James' to 'Suresh' :\n"+color.END)
exam['Name'] = exam['Name'].replace('James', 'Suresh')
print(exam)
Seperator()

#7
print(color.BOLD+"7.Sum of the examination attempts by the students :"+color.END)
print(exam['Attempts'].sum())
Seperator()

#8
print(color.BOLD+"8.Append one row :\n"+color.END)
exam2 = {'Name': "Harsh", 'Score': 56, 'Attempts': 1, 'Qualify': "Yes"}
exam = exam.append(exam2, ignore_index=True)
print(exam)
Seperator()

#9
print(color.BOLD+"9.DataFrame after insertion of new column :\n"+color.END)
exam = exam.assign(Exam_Name=["Mid-Sem", "Mid-Sem", "Mid-Sem", "Mid-Sem", "Mid-Sem", "Mid-Sem", "Mid-Sem"])
print(exam)
del exam['Exam_Name']
print(color.BOLD+"\n DataFrame after deletion :\n"+color.END)
print(exam)
Seperator()

#10
print(color.BOLD+"10.Convert a NumPy array to a series :\n"+color.END)
a = pd.Series(rank)
print(a)

print(color.BOLD+"\n Convert dictionary to a series :\n"+color.END)
d = pd.Series(Exam_Data_Array)
print(d)

print(color.BOLD+"\n Convert first column of a DataFrame to a series :\n"+color.END)
df = exam.iloc[:,0]
```

12002040701067

1.Rows where score between 15 and 20 (inclusive):

	Name	Score	Attempts	Qualify
rank1	Aeshwary	17	1	No

2.Sort the data frame first by 'score' in ascending order :

	Name	Score	Attempts	Qualify
rank5	Hunaïd	10	1	No
rank1	Aeshwary	17	1	No
rank6	James	27	2	No
rank2	Dhruval	35	1	Yes
rank3	Harshad	65	1	Yes
rank4	Hitendra	90	1	Yes

Sort the data frame first by 'name' in descending order :

	Name	Score	Attempts	Qualify
rank6	James	27	2	No
rank5	Hunaïd	10	1	No
rank4	Hitendra	90	1	Yes
rank3	Harshad	65	1	Yes
rank2	Dhruval	35	1	Yes
rank1	Aeshwary	17	1	No

3.Replace the 'qualify' column contains the values  
'yes' and 'no' with True and False :

	Name	Score	Attempts	Qualify
rank1	Aeshwary	17	1	False
rank2	Dhruval	35	1	True
rank3	Harshad	65	1	True
rank4	Hitendra	90	1	True
rank5	Hunaïd	10	1	False
rank6	James	27	2	False

4.Select specific columns(columns: 2) & rows(row: 1,3 and 4):

	Attempts
rank2	1
rank4	1
rank5	1

5.Number of attempts in the examination  
is less than 2 and score greater than 15 :

	Name	Score	Attempts	Qualify
rank1	Aeshwary	17	1	False
rank2	Dhruval	35	1	True
rank3	Harshad	65	1	True
rank4	Hitendra	90	1	True

6.Change the name 'James' to 'Suresh' :

	Name	Score	Attempts	Qualify
rank1	Aeshwary	17	1	False
rank2	Dhruval	35	1	True

```
print(df)
print(type(df))
```

6.Change the name 'James' to 'Suresh' :

	Name	Score	Attempts	Qualify
rank1	Aeshwary	17	1	False
rank2	Dhruval	35	1	True
rank3	Harshad	65	1	True
rank4	Hitendra	90	1	True
rank5	Hunaid	10	1	False
rank6	Suresh	27	2	False

7.Sum of the examination attempts by the students :

7

8.Append one row :

	Name	Score	Attempts	Qualify
0	Aeshwary	17	1	False
1	Dhruval	35	1	True
2	Harshad	65	1	True
3	Hitendra	90	1	True
4	Hunaid	10	1	False
5	Suresh	27	2	False
6	Harsh	56	1	Yes

9.DataFrame after insertion of new column :

	Name	Score	Attempts	Qualify	Exam_Name
0	Aeshwary	17	1	False	Mid-Sem
1	Dhruval	35	1	True	Mid-Sem
2	Harshad	65	1	True	Mid-Sem
3	Hitendra	90	1	True	Mid-Sem
4	Hunaid	10	1	False	Mid-Sem
5	Suresh	27	2	False	Mid-Sem
6	Harsh	56	1	Yes	Mid-Sem

DataFrame after deletion :

	Name	Score	Attempts	Qualify
0	Aeshwary	17	1	False
1	Dhruval	35	1	True
2	Harshad	65	1	True
3	Hitendra	90	1	True
4	Hunaid	10	1	False
5	Suresh	27	2	False
6	Harsh	56	1	Yes

10.Convert a NumPy array to a series :

```
0    rank1
1    rank2
2    rank3
3    rank4
4    rank5
5    rank6
dtype: object
```

Convert dictionary to a series :

```
print(df)
```

```
print(type(df))
```

```

1  Dhruval    35      1    True
2  Harshad   65      1    True
3  Hitendra  90      1    True
4  Hunaid    10      1   False
5  Suresh    27      2   False
6  Harsh     56      1    Yes

```

#### 9.DataFrame after insertion of new column :

	Name	Score	Attempts	Qualify	Exam_Name
0	Aeshwary	17	1	False	Mid-Sem
1	Dhruval	35	1	True	Mid-Sem
2	Harshad	65	1	True	Mid-Sem
3	Hitendra	90	1	True	Mid-Sem
4	Hunaid	10	1	False	Mid-Sem
5	Suresh	27	2	False	Mid-Sem
6	Harsh	56	1	Yes	Mid-Sem

#### DataFrame after deletion :

	Name	Score	Attempts	Qualify
0	Aeshwary	17	1	False
1	Dhruval	35	1	True
2	Harshad	65	1	True
3	Hitendra	90	1	True
4	Hunaid	10	1	False
5	Suresh	27	2	False
6	Harsh	56	1	Yes

#### 10.Convert a NumPy array to a series :

```

0    rank1
1    rank2
2    rank3
3    rank4
4    rank5
5    rank6
dtype: object

```

#### Convert dictionary to a series :

```

Name      [Aeshwary, Dhruval, Harshad, Hitendra, Hunaid,...
Score      [17, 35, 65, 90, 10, 27]
Attempts    [1, 1, 1, 1, 1, 2]
Qualify     [No, Yes, Yes, Yes, No, No]
dtype: object

```

#### Convert first column of a DataFrame to a series :

```

0    Aeshwary
1    Dhruval
2    Harshad
3    Hitendra
4    Hunaid
5    Suresh
6    Harsh
Name: Name, dtype: object
<class 'pandas.core.series.Series'>

```