NAME - ADYA KUMARI
ROLL - 22352002

```
import pandas as pd;
data = {'Name': ['Jai', 'Princi', 'Gaurav',
 'Anuj', 'Ravi', 'Natasha', 'Riya'],
 'Age': [17, 17, 18, 17, 18, 17, 17],
 'Gender': ['M', 'F', 'M', 'M', 'M', 'F', 'F'],
 'Marks': [90, 76, 'NaN', 74, 65, 'NaN', 71]}
# Convert into DataFrame
df = pd.DataFrame(data)
# Display data
df
```

	Name	Age	Gender	Marks
0	Jai	17	M	90
1	Princi	17	F	76
2	Gaurav	18	M	NaN
3	Anuj	17	M	74
4	Ravi	18	M	65
5	Natasha	17	F	NaN
6	Riya	17	F	71

```
c = avg = 0
for ele in df['Marks']:
 if str(ele).isnumeric():
  c += 1
  avg += ele
avg /= c
df = df.replace(to_replace="NaN",
 value=avg)
# Display data
df
```

	Name	Age	Gender	Marks
0	Jai	17	M	90.0
1	Princi	17	F	76.0
2	Gaurav	18	M	75.2
3	Anuj	17	M	74.0
4	Ravi	18	M	65.0
5	Natasha	17	F	75.2
6	Riya	17	F	71.0

Categorizing data. (In this we categorized gender for male its 0 and for female its one .

```
df['Gender']=df['Gender'].map({'M':0,'F':1,}).astype(int)
df
```

	Name	Age	Gender	Marks

FILTERING TOP SCORING MARKS

```
df[df['Marks']>=90]
```

	Name	Age	Gender	Marks
0	Jai	17	0	90.0

```
df
```

	Name	Age	Gender	Marks
0	Jai	17	0	90.0
1	Princi	17	1	76.0
2	Gaurav	18	0	75.2
3	Anuj	17	0	74.0
4	Ravi	18	0	65.0
5	Natasha	17	1	75.2
6	Riya	17	1	71.0

MERGING TWO DATASETS

Syntax for it is : pd.merge(data_frame1,data_frame2, on="field ")

```
#CREATING  TWO NEW DATASET DETAILS and FEE STATUS
details=pd.DataFrame({
    'ID':[101,102,103,104,105,106,107,108,109,110],
    'Name':['Jagroop', 'Praveen', 'Harjot','Pooja', 'Rahul', 'Nikita','Saurabh', 'Ayush', 'Dolly', 'Mohit'],
    'Branch':['CSE', 'CSE', 'CSE', 'CSE', 'CSE','CSE', 'CSE', 'CSE', 'CSE', 'CSE']
})
details
```

	ID	Name	Branch
0	101	Jagroop	CSE
1	102	Praveen	CSE
2	103	Harjot	CSE
3	104	Pooja	CSE
4	105	Rahul	CSE
5	106	Nikita	CSE
6	107	Saurabh	CSE
7	108	Ayush	CSE
8	109	Dolly	CSE
9	110	Mohit	CSE

```
fee_status=pd.DataFrame({
    'ID':[101,102,103,104,105,106,107,108,109,110],
    'PENDING': ['5000', '250', 'NIL','9000', '15000', 'NIL','4500', '1800', '250', 'NIL']
})

fee_status
```

	ID	PENDING
0	101	5000
1	102	250
2	103	NIL
3	104	9000
4	105	15000
5	106	NIL
6	107	4500

```
new=pd.merge(details,fee_status,on="ID")
new
```

	ID	Name	Branch	PENDING
0	101	Jagroop	CSE	5000
1	102	Praveen	CSE	250
2	103	Harjot	CSE	NIL
3	104	Pooja	CSE	9000
4	105	Rahul	CSE	15000
5	106	Nikita	CSE	NIL
6	107	Saurabh	CSE	4500
7	108	Ayush	CSE	1800
8	109	Dolly	CSE	250
9	110	Mohit	CSE	NIL

GROUP BY

groupby is used for grouping the data according to the categories and applying a function to the categories.

Syntax for it is : df.groupby('parameter name onto which you want to create group')

```
gr=new.groupby('PENDING')
print(gr.get_group('250'))
```

```
    ID     Name Branch PENDING
1  102  Praveen    CSE     250
8  109    Dolly    CSE     250
```

Removing Duplicate

Pandas duplicated() method is used to remove duplicate from the dataset

Syntax :dataframe_name.duplicated('column_name')

for getting non duplicate data add ~ (not) sign before the dataframe in the syntax

```
student_data = {'Name': ['Amit', 'Praveen', 'Jagroop', 'Rahul', 'Vishal', 'Suraj', 'Rishab', 'Satyapal', 'Amit', 'Rahul', 'Praveen', 'Amit'],
                'Roll_no': [23, 54, 29, 36, 59, 38, 12, 45, 34, 36, 54, 23],
                'Email': ['xxxx@gmail.com', 'xxxxxx@gmail.com', 'xxxxxx@gmail.com', 'xx@gmail.com','xxxx@gmail.com', 'xxxxx@gmail.com','xxxxx
```

```
dff = pd.DataFrame(student_data)
dff
```

	Name	Roll_no	Email
0	Amit	23	xxxx@gmail.com
1	Praveen	54	xxxxxx@gmail.com
2	Jagroop	29	xxxxxx@gmail.com
3	Rahul	36	xx@gmail.com
4	Vishal	59	xxxx@gmail.com
5	Suraj	38	xxxxx@gmail.com
6	Rishab	12	xxxxx@gmail.com
7	Satyapal	45	xxxxx@gmail.com
8	Amit	34	xxxxx@gmail.com
9	Rahul	36	xxxxxx@gmail.com

```
#for getting the duplicated value
dff[dff.duplicated('Roll_no')]
```

	Name	Roll_no	Email
9	Rahul	36	xxxxxx@gmail.com
10	Praveen	54	xxxxxxxxxx@gmail.com
11	Amit	23	xxxxxxxxxx@gmail.com

```
#for getting the non duplicated value ~(NOT) is used
dff[~dff.duplicated('Roll_no')]
```

	Name	Roll_no	Email
0	Amit	23	xxxx@gmail.com
1	Praveen	54	xxxxxx@gmail.com
2	Jagroop	29	xxxxxx@gmail.com
3	Rahul	36	xx@gmail.com
4	Vishal	59	xxxx@gmail.com
5	Suraj	38	xxxxx@gmail.com
6	Rishab	12	xxxxx@gmail.com
7	Satyapal	45	xxxxx@gmail.com
8	Amit	34	xxxxx@gmail.com

Concatenation of Two Datasets

Syntax for it is : pd.concat(data_frame1,data_frame2)

```
data1 = {'Name':['Jai', 'Princi', 'Gaurav', 'Anuj'],
         'Age':[27, 24, 22, 32],
         'Address':['Nagpur', 'Kanpur', 'Allahabad', 'Kannuaj'],
         'Qualification':['Msc', 'MA', 'MCA', 'Phd'],
         'Mobile No': [97, 91, 58, 76]}
d1=pd.DataFrame(data1,index=[0,1,2,3])
d1
```

	Name	Age	Address	Qualification	Mobile No
0	Jai	27	Nagpur	Msc	97
1	Princi	24	Kanpur	MA	91
2	Gaurav	22	Allahabad	MCA	58
3	Anuj	32	Kannuaj	Phd	76

```
data2 = {'Name':['Gaurav', 'Anuj', 'Dhiraj', 'Hitesh'],
         'Age':[22, 32, 12, 52],
         'Address':['Allahabad', 'Kannuaj', 'Allahabad', 'Kannuaj'],
         'Qualification':['MCA', 'Phd', 'Bcom', 'B.hons'],
         'Salary':[1000, 2000, 3000, 4000]}

d2=pd.DataFrame(data2,index=[2,3,6,7])
d2
```

	Name	Age	Address	Qualification	Salary
2	Gaurav	22	Allahabad	MCA	1000
3	Anuj	32	Kannuaj	Phd	2000
6	Dhiraj	12	Allahabad	Bcom	3000
7	Hitesh	52	Kannuaj	B.hons	4000

```
res=pd.concat([d1,d2])
res
```

	Name	Age	Address	Qualification	Mobile No	Salary
0	Jai	27	Nagpur	Msc	97.0	NaN
1	Princi	24	Kanpur	MA	91.0	NaN
2	Gaurav	22	Allahabad	MCA	58.0	NaN
3	Anuj	32	Kannuaj	Phd	76.0	NaN
2	Gaurav	22	Allahabad	MCA	NaN	1000.0
3	Anuj	32	Kannuaj	Phd	NaN	2000.0
6	Dhiraj	12	Allahabad	Bcom	NaN	3000.0
7	Hitesh	52	Kannuaj	B.hons	NaN	4000.0

✓ 0s completed at 4:38 PM