In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
#pandas operations
#conditional selection
#access rows and col's
#add and delete columns
#concatenation merging & joining data frames
#handling missing values
#apply()
```

In [93]:
```
d={'USN':[100,101,102],
   'name':['raj','ram','rajesh'],
  'mobile':[99,88,77],
   'marks':[10,9,8]
  }
```

In [94]:
```
std=pd.DataFrame(d)
```

In [95]:
```
print(std)
```

```
   USN    name  mobile  marks
0  100     raj      99     10
1  101     ram      88      9
2  102  rajesh      77      8
```

In [9]:
```
#dataframe operations
std.head(2)
#head displays row - head
#if we pass any parameter it shows upto that row
#by default it shows all rows
```

Out[9]:

	USN	name	mobile	marks
0	100	raj	99	10
1	101	ram	88	9

In [11]:
```
std.info()
#all the info regarding the dataframe
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   USN     3 non-null      int64 
 1   name    3 non-null      object
 2   mobile  3 non-null      int64 
 3   marks   3 non-null      int64 
dtypes: int64(3), object(1)
memory usage: 224.0+ bytes
```

In [13]:
```
std.columns
#return list of columns in dataframe
```

Out[13]:
```
Index(['USN', 'name', 'mobile', 'marks'], dtype='object')
```

In [17]:
```
std.isnull()
```

Out[17]:

	USN	name	mobile	marks
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False

In [28]:
```
#accessing columns
std['USN']
```

Out[28]:
```
0    100
1    101
2    102
Name: USN, dtype: int64
```

In [33]:
```
#accessing multiple columns we need to send the list as parameter that we ne
std[['USN','name']]
```

Out[33]:

	USN	name
0	100	raj
1	101	ram
2	102	rajesh

In [82]:
```
#access rows - returns row - index start from 0
#it takes row label
std.loc[2]
```

Out[82]:
```
USN           102
name       rajesh
mobile         77
marks        null
Name: 2, dtype: object
```

In [85]:
```
std.iloc[2]
```

Out[85]:
```
USN           102
name       rajesh
mobile         77
marks        null
Name: 2, dtype: object
```

In [87]:
```
index=['A','B','c']
std.set_index('USN')
```

Out[87]:

	name	mobile	marks
USN			
100	raj	99	10
101	ram	88	null
102	rajesh	77	null

In [89]:
```
std.iloc[2]
```

Out[89]:
```
USN           102
name       rajesh
mobile         77
marks        null
Name: 2, dtype: object
```

In [ ]:
```
QN : Create a data frame cars with attributes car id , car name ,
```

In [47]:
```
cars={'carid':[1,2,3,4,5],
      'name':['rollsroyce','rangerover','benz','bmw','lambhorgini']
    }
```

In [49]:
```
ra=pd.DataFrame(cars)
```

In [50]:
```
ra
```

Out[50]:

	carid	name
0	1	rollsroyce
1	2	rangerover
2	3	benz
3	4	bmw
4	5	lambhorgini

In [52]:
```
ra.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   carid   5 non-null      int64 
 1   name    5 non-null      object
dtypes: int64(1), object(1)
memory usage: 208.0+ bytes
```

In [77]:
```
#to prove that isnull is true
f={'USN':[100,101,102],
   'name':['raj','ram','rajesh'],
  'mobile':[99,88,77],
   'marks':[10,None,None]
  }
```

In [78]:
```
td=pd.DataFrame(f)
```

In [79]:
```
td
```

Out[79]:

	USN	name	mobile	marks
0	100	raj	99	10.0
1	101	ram	88	NaN
2	102	rajesh	77	NaN

In [80]:
```
td.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   USN     3 non-null      int64  
 1   name    3 non-null      object 
 2   mobile  3 non-null      int64  
 3   marks   1 non-null      float64
dtypes: float64(1), int64(2), object(1)
memory usage: 224.0+ bytes
```

In [81]:
```
td.isnull()
```

Out[81]:

	USN	name	mobile	marks
0	False	False	False	False
1	False	False	False	True
2	False	False	False	True

In [96]:
```
std
```

Out[96]:

	USN	name	mobile	marks
0	100	raj	99	10
1	101	ram	88	9
2	102	rajesh	77	8

In [97]:
```
#conditional selection
std['marks']<9
```

Out[97]:
```
0    False
1    False
2     True
Name: marks, dtype: bool
```

In [104…
```
#we will get only boolean values
#using this to overcomme this and get the values we need to use new data fra
newdf=std['marks']<9
```

In [105…
```
std[newdf]
```

Out[105]:

	USN	name	mobile	marks
2	102	rajesh	77	8

In [106…
```
#to get the particular attribute in the dataframe we use this
std[newdf]['name']
```

Out[106]:
```
2    rajesh
Name: name, dtype: object
```

In [107…
```
newd=std['marks']>9
```

In [109…
```
std[newd]
```

Out[109]:

	USN	name	mobile	marks
0	100	raj	99	10

In [110…

```
#multiple conditions
std[(std['USN']>101)|(std['marks']<40)]
```

Out[110]:

	USN	name	mobile	marks
0	100	raj	99	10
1	101	ram	88	9
2	102	rajesh	77	8

In [113…

```
std[(std['USN']>101)&(std['marks']<10)]
```

Out[113]:

	USN	name	mobile	marks
2	102	rajesh	77	8

In [115…

```
#adding new column
std
```

Out[115]:

	USN	name	mobile	marks
0	100	raj	99	10
1	101	ram	88	9
2	102	rajesh	77	8

In [122…

```
s='banglore,chennai,dmm'.split(',')
#split divides the string and makes into a list
print(s)
print(type(s))
```

```
['banglore', 'chennai', 'dmm']
<class 'list'>
```

In [124…

```
std['Address']=s
```

In [125…

```
std
```

Out[125]:

	USN	name	mobile	marks	Address
0	100	raj	99	10	banglore
1	101	ram	88	9	chennai
2	102	rajesh	77	8	dmm

In [130…

```
#drop will effect only on that instance
#original data will not be affected
std.drop(2)
```

Out[130]:

	USN	name	mobile	marks	Address
0	100	raj	99	10	banglore
1	101	ram	88	9	chennai

In [138…

```
d2={'USN':[None,101,102],
    'name':[None,None,'rajesh'],
   'mobile':[99,88,77],
    'marks':[35,33,None]
   }
```

In [139…

```
df=pd.DataFrame(d2)
```

In [140…

```
df
```

Out[140]:

	USN	name	mobile	marks
0	NaN	None	99	35.0
1	101.0	None	88	33.0
2	102.0	rajesh	77	NaN

In [ ]: