Pandas - II

🎯 Session Objectives:

- ✅ Differentiate Pandas Series vs NumPy Arrays
- ✅ Create Series from scalar, list, array, and dictionary
- ✅ Access Series elements using indexing and slicing
- ✅ Understand attributes of Series
- ✅ Learn basic mathematical operations on Series
- ✅ Understand the key attributes of a DataFrame
- ✅ Use various DataFrame methods to explore and manipulate data
- ✅ Perform row and column operations
- ✅ Use joining, merging, and concatenation techniques across DataFrames

What is DataFrame?

- A DataFrame is a 2D Labelled Data Structure in Pandas - Like an Excel Sheet or a SQL Table
- It has rows + cols and each columns contains different types of data(object,int,str,float,etc...)

```
import numpy as np
import pandas as pd
empty_df = pd.DataFrame()
empty_df
```

```
—
```

```
type(empty_df)
```

```
pandas.core.frame.DataFrame
```

```
print(empty_df)
```

```
Empty DataFrame
Columns: []
Index: []
```

```python
# Let's Create a DataFrame which includes Person Data

# Name | Age | State
personal_info = np.array([
    ['Aarif Pathan',28,'Rajasthan'],
    ['Lubhani Gola', 26, 'Delhi'],
    ['Utkarsh', 32, 'Maharashtra'],
    ['Kushagra',27,'Chandigarh'],
    ['Aditya Verma', 23, 'Delhi'],
    ['Palash',32,'Uttarakhand'],
    ['Arpit Gupta',30, 'Ghaziabad'],
    ['Ali',27,'Delhi'],
    ['Akansha', 26 ,'Delhi']
])
person_df = pd.DataFrame(personal_info , columns = ['Name','Age','State'])
person_df
```

	Name	Age	State
0	Aarif Pathan	28	Rajasthan
1	Lubhani Gola	26	Delhi
2	Utkarsh	32	Maharashtra
3	Kushagra	27	Chandigarh
4	Aditya Verma	23	Delhi
5	Palash	32	Uttarakhand
6	Arpit Gupta	30	Ghaziabad
7	Ali	27	Delhi
8	Akansha	26	Delhi

```python
person_df.head() # By Default top 5 rows from original DataFrame
```

	Name	Age	State
0	Aarif Pathan	28	Rajasthan
1	Lubhani Gola	26	Delhi
2	Utkarsh	32	Maharashtra
3	Kushagra	27	Chandigarh
4	Aditya Verma	23	Delhi

```python
person_df.tail() # By Default Bottom 5 rows from original DataFrame
```

	Name	Age	State
4	Aditya Verma	23	Delhi
5	Palash	32	Uttarakhand
6	Arpit Gupta	30	Ghaziabad
7	Ali	27	Delhi
8	Akansha	26	Delhi

```
person_df.describe() # Statistical Analysis
```

	Name	Age	State
count	9	9	9
unique	9	6	6
top	Aarif Pathan	26	Delhi
freq	1	2	4

```
count -> non-null values
unique -> number of distinct values
top -> Most Frequent Value [Mode]
freq -> Frequency of the most frequent value
```

```
person_df['Age'].value_counts()
```

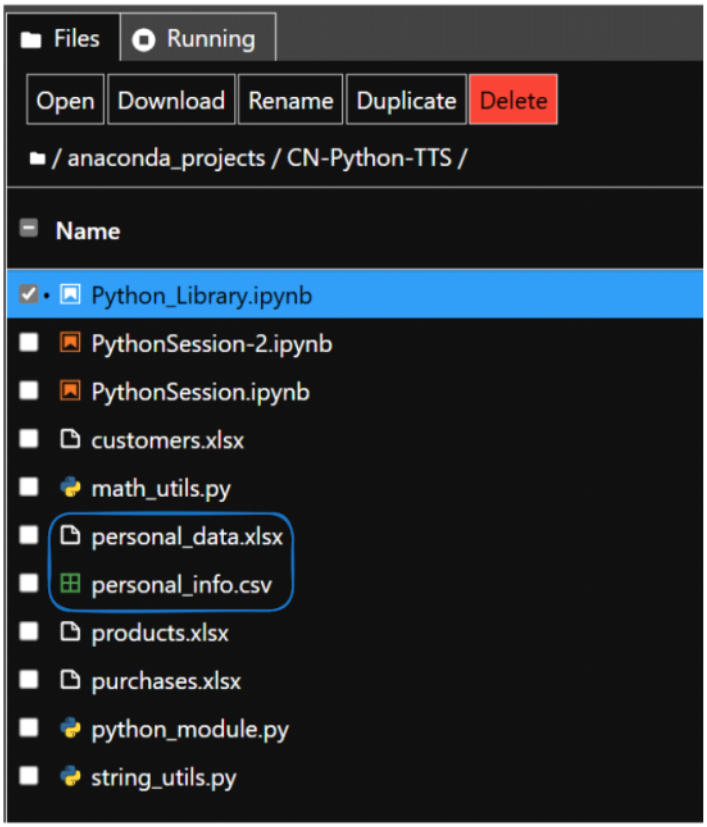
```
Age
26    2
32    2
27    2
28    1
23    1
30    1
Name: count, dtype: int64
```

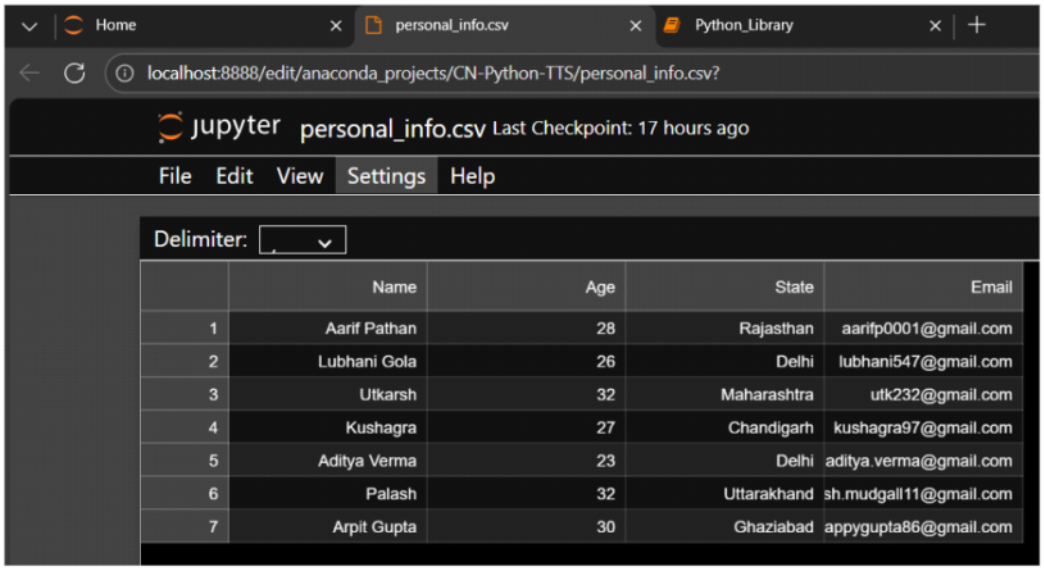
```
# Personal Details

# Name | Age | City | Email
personal_data = {
    'Name' : ['Aarif Pathan','Lubhani Gola','Utkarsh','Kushagra','Aditya Verma','Palash','Arpit Gupta'],
    'Age' : [28,26,32,27,23,32,30],
    'State' : ['Rajasthan','Delhi','Maharashtra','Chandigarh','Delhi','Uttarakhand','Ghaziabad'],
    'Email' : ['aarifp0001@gmail.com','lubhani547@gmail.com','utk232@gmail.com','kushagra97@gmail.com',
               'aditya.verma@gmail.com', 'palash.mudgall11@gmail.com','appygupta86@gmail.com']
}

person_df = pd.DataFrame(personal_data)
person_df
```

	Name	Age	State	Email
0	Aarif Pathan	28	Rajasthan	aarifp0001@gmail.com
1	Lubhani Gola	26	Delhi	lubhani547@gmail.com
2	Utkarsh	32	Maharashtra	utk232@gmail.com
3	Kushagra	27	Chandigarh	kushagra97@gmail.com
4	Aditya Verma	23	Delhi	aditya.verma@gmail.com
5	Palash	32	Uttarakhand	palash.mudgall11@gmail.com
6	Arpit Gupta	30	Ghaziabad	appygupta86@gmail.com

```python
# Exporting a File .csv , .excel
person_df.to_csv('personal_info.csv', index=False)
```

```python
# Exporting to .excel
person_df.to_excel('personal_data.xlsx', index = False)
```





	Name	Age	State	Email
1	Aarif Pathan	28	Rajasthan	aarifp0001@gmail.com
2	Lubhani Gola	26	Delhi	lubhani547@gmail.com
3	Utkarsh	32	Maharashtra	utk232@gmail.com
4	Kushagra	27	Chandigarh	kushagra97@gmail.com
5	Aditya Verma	23	Delhi	aditya.verma@gmail.com
6	Palash	32	Uttarakhand	sh.mudgall11@gmail.com
7	Arpit Gupta	30	Ghaziabad	appygupta86@gmail.com

```python
# Import the DataSet....
try :
    customers = pd.read_excel('customers.xlsx')
    products = pd.read_excel('products.xlsx')
    purchases = pd.read_excel('purchases.xlsx')
except FileNotFoundError:
    print("File Not Found!")
```

customers

2	3	Bambi	Westrip	bwestrip2@symantec.com	Female	4057.0	Arkansas	Circle	San Antonio	
3	4	Roarke	Pankettman	rpankettman3@wiley.com	Male	74.0	Debs	Point	Memphis	Te
4	5	Mikaela	Althorpe	malthorpe4@51.la	NaN	NaN	2nd	Drive	NaN	
...	...	...	...	...	...	...	...	...	...	
995	996	Merrili	Alman	malmanrn@cornell.edu	Female	0.0	Thompson	Way	Reading	Penn
996	997	Winonah	Heckle	whecklero@fc2.com	Female	701.0	Rowland	Hill	Indianapolis	
997	998	Tobit	Birt	NaN	Male	2.0	Basil	Road	Waterbury	Con
998	999	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	NaN	Kenwood	Drive	Savannah	
999	1000	Elmore	Malpas	NaN	Male	205.0	Farwell	Park	Atlanta	

products

	Unnamed: 0	id	product	cost	company
0	0	1	Liners - Baking Cups	$6.36	Skipfire
1	1	2	Nori Sea Weed - Gold Label	$85.74	Dynazzy
2	2	3	Bar Bran Honey Nut	$65.40	Ntag
3	3	4	Soup - Campbells Beef Stew	$68.16	Photojam
4	4	5	Wine - Shiraz Wolf Blass Premium	$87.39	Eare
5	5	6	Wine - White, Riesling, Semi - Dry	$99.22	Livepath
6	6	7	Brandy - Bar	$13.83	Oloo
7	7	8	Onions - White	$42.19	Oozz
8	8	9	Lettuce - Baby Salad Greens	$30.01	Meevee

purchases

	Unnamed: 0	id	purch_date	customer_num	product_num	amount	paid
0	0	1	2019-01-03 00:00:00	823	27	12	$568.92
1	1	2	2019-01-03 00:00:00	606	28	14	$395.36
2	2	3	2019-01-03 00:00:00	955	9	17	$510.17
3	3	4	2019-01-03 00:00:00	577	19	3	$68.49
4	4	5	2019-01-03 00:00:00	429	8	18	$759.42
...	...	...	...	...	...	...	...
5995	5995	5996	06/20/2019	893	33	5	$411.10
5996	5996	5997	06/20/2019	566	23	11	$178.97
5997	5997	5998	06/20/2019	114	19	9	$205.47
5998	5998	5999	06/20/2019	404	11	20	$429.40
5999	5999	6000	06/20/2019	88	57	4	$274.52

6000 rows × 7 columns

```
# Import the .csv DataSet ....
try :
    customers = pd.read_csv('customers.csv')
    products = pd.read_csv('products.csv')
    purchases = pd.read_csv('purchases.csv')
except FileNotFoundError:
    print("File Not Found!")
```

```
# Finding the Statistical Analysis of Customers Table
customers.describe()
```

	id	street_num	postcode
count	1000.000000	738.000000	843.000000
mean	500.500000	10536.439024	52669.548043
std	288.819436	23050.537603	28140.041026
min	1.000000	0.000000	214.000000
25%	250.750000	21.000000	29279.500000
50%	500.500000	445.500000	48232.000000
75%	750.250000	6918.500000	78337.500000
max	1000.000000	99918.000000	99812.000000

```
products.describe()
```

	Unnamed: 0	id
count	60.000000	60.000000
mean	29.500000	30.500000
std	17.464249	17.464249
min	0.000000	1.000000
25%	14.750000	15.750000
50%	29.500000	30.500000
75%	44.250000	45.250000
max	59.000000	60.000000

```
purchases.describe()
```

	Unnamed: 0	id	customer_num	product_num	amount
count	6000.000000	6000.000000	6000.000000	6000.000000	6000.000000
mean	2999.500000	3000.500000	500.889333	30.140667	10.576167
std	1732.195139	1732.195139	288.377188	17.249613	5.768889
min	0.000000	1.000000	1.000000	1.000000	1.000000
25%	1499.750000	1500.750000	244.000000	15.000000	6.000000
50%	2999.500000	3000.500000	511.000000	30.000000	11.000000
75%	4499.250000	4500.250000	751.000000	45.000000	16.000000
max	5999.000000	6000.000000	1000.000000	60.000000	20.000000

```
customers['gender'].describe()
```

```
count      957
unique       2
top       Male
freq       485
Name: gender, dtype: object
```

```
customers.describe(include = [object] )
```

	first_name	last_name	email	gender	street_name	street_suffix	city	state
count	1000	1000	878	957	963	963	921	920
unique	932	993	878	2	427	21	296	48
top	Berty	Sedworth	rsouthcott0@clickbank.net	Male	Arizona	Place	Washington	Texas
freq	4	3	1	485	6	57	29	107

```
customers['first_name'].value_counts()
```

```
first_name
Berty        4
Leland       3
Abel         3
Gunther      2
Silvain      2
            ..
Gaven        1
Chuck        1
Florentia    1
Raffarty     1
Elmore       1
Name: count, Length: 932, dtype: int64
```

```
customers['street_name'].value_counts()
```

```
street_name
Arizona          6
Farmco           6
Weeping Birch    6
Hansons          6
Cambridge        6
                ..
Dennis           1
Merchant         1
Johnson          1
Sutherland       1
Kenwood          1
Name: count, Length: 427, dtype: int64
```

```
customers['state'].value_counts()
```

```
state
Texas                   107
California              101
Florida                  75
New York                 43
Ohio                     41
Pennsylvania             35
District of Columbia     33
Georgia                  28
Virginia                 27
Illinois                 23
Tennessee                23
Missouri                 22
Indiana                  22
North Carolina           20
Louisiana                20
```

```
customers.describe(include = 'all')
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city
count	1000.000000	1000	1000	878	957	738.000000	963	963	921
unique	NaN	932	993	878	2	NaN	427	21	296
top	NaN	Berty	Sedworth	rsouthcott0@clickbank.net	Male	NaN	Arizona	Place	Washington
freq	NaN	4	3	1	485	NaN	6	57	29
mean	500.500000	NaN	NaN	NaN	NaN	10536.439024	NaN	NaN	NaN
std	288.819436	NaN	NaN	NaN	NaN	23050.537603	NaN	NaN	NaN
min	1.000000	NaN	NaN	NaN	NaN	0.000000	NaN	NaN	NaN
25%	250.750000	NaN	NaN	NaN	NaN	21.000000	NaN	NaN	NaN
50%	500.500000	NaN	NaN	NaN	NaN	445.500000	NaN	NaN	NaN
75%	750.250000	NaN	NaN	NaN	NaN	6918.500000	NaN	NaN	NaN
max	1000.000000	NaN	NaN	NaN	NaN	99918.000000	NaN	NaN	NaN

```
customers.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             1000 non-null   int64
 1   first_name     1000 non-null   object
 2   last_name      1000 non-null   object
 3   email          878 non-null    object
 4   gender         957 non-null    object
 5   street_num     738 non-null    float64
 6   street_name    963 non-null    object
 7   street_suffix  963 non-null    object
 8   city           921 non-null    object
 9   state          920 non-null    object
 10  postcode       843 non-null    float64
dtypes: float64(2), int64(1), object(8)
memory usage: 86.1+ KB
```

```
products.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  60 non-null     int64
 1   id          60 non-null     int64
 2   product     60 non-null     object
 3   cost        60 non-null     object
 4   company     55 non-null     object
dtypes: int64(2), object(3)
memory usage: 2.5+ KB
```

```
purchases.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6000 entries, 0 to 5999
Data columns (total 7 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Unnamed: 0    6000 non-null   int64 
 1   id            6000 non-null   int64 
 2   purch_date    6000 non-null   object
 3   customer_num  6000 non-null   int64 
 4   product_num   6000 non-null   int64 
 5   amount        6000 non-null   int64 
 6   paid          6000 non-null   object
dtypes: int64(5), object(2)
memory usage: 328.3+ KB
```

```
# using .iloc[Positional Based Indexing] & loc[Label Based Indexing]
# DataFrame [2D Matrix] # .iloc[rows , cols]
# Find all the records of Customers with FirstName Columns
FirstName = customers.iloc[: , 1] # .iloc[:(all records) , 1 [col at idx 1]]
FirstName # Series
```

```
0       Romain
1       Cosimo
2        Bambi
3       Roarke
4      Mikaela
        ...
995    Merrili
996    Winonah
997      Tobit
998     Issiah
999     Elmore
Name: first_name, Length: 1000, dtype: object
```

```
type(FirstName)
```

```
pandas.core.series.Series
```

```
# .Loc [Label Based] [rows , cols]
FirstName = customers.loc[: , 'first_name']
FirstName
```

```
0       Romain
1       Cosimo
2        Bambi
3       Roarke
4      Mikaela
        ...
995    Merrili
996    Winonah
997      Tobit
998     Issiah
999     Elmore
Name: first_name, Length: 1000, dtype: object
```

```
Email = customers.loc[: , 'email']
Email
```

```
0              rsouthcott0@clickbank.net
1                  cmolyneaux1@wiley.com
2                 bwestrip2@symantec.com
3                 rpankettman3@wiley.com
4                       malthorpe4@51.la
                        ...
995               malmanrn@cornell.edu
996                  whecklero@fc2.com
997                                 NaN
998    istandbrookerq@yellowpages.com
999                                 NaN
Name: email, Length: 1000, dtype: object
```

```python
# .loc[label-based]
customers_info = customers.loc[1:10 , ['first_name' , 'last_name' , 'email' , 'gender' , 'city']]
customers_info # DataFrame
```

	first_name	last_name	email	gender	city
1	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	El Paso
2	Bambi	Westrip	bwestrip2@symantec.com	Female	San Antonio
3	Roarke	Pankettman	rpankettman3@wiley.com	Male	Memphis
4	Mikaela	Althorpe	malthorpe4@51.la	NaN	NaN
5	Magdalena	Cullip	mcullip5@tiny.cc	Female	Baltimore
6	Marietta	Heball	mheball6@blog.com	Female	Carol Stream
7	Tine	McSperrin	tmcsperrin7@statcounter.com	NaN	Kansas City
8	Enrichetta	de Villier	edevillier8@ox.ac.uk	Female	Fairbanks
9	Sari	Poulden	spoulden9@xing.com	Female	New Orleans
10	Natale	Martina	nmartinaa@wordpress.com	Male	Bradenton

```python
customers_info = customers.iloc[1:11 , 1:5] # [rows from 1 to 10 , cols [first_name till gender]]
customers_info # DataFrame
```

	first_name	last_name	email	gender
1	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male
2	Bambi	Westrip	bwestrip2@symantec.com	Female
3	Roarke	Pankettman	rpankettman3@wiley.com	Male
4	Mikaela	Althorpe	malthorpe4@51.la	NaN
5	Magdalena	Cullip	mcullip5@tiny.cc	Female
6	Marietta	Heball	mheball6@blog.com	Female
7	Tine	McSperrin	tmcsperrin7@statcounter.com	NaN
8	Enrichetta	de Villier	edevillier8@ox.ac.uk	Female
9	Sari	Poulden	spoulden9@xing.com	Female
10	Natale	Martina	nmartinaa@wordpress.com	Male

```python
customers_info  = customers.iloc[1:11 , [1,3,4,8,9]]
# cols picked ['first_name','email','gender','city','state']
customers_info # DataFrame
```

	first_name	email	gender	city	state
1	Cosimo	cmolyneaux1@wiley.com	Male	El Paso	Texas
2	Bambi	bwestrip2@symantec.com	Female	San Antonio	Texas
3	Roarke	rpankettman3@wiley.com	Male	Memphis	Tennessee
4	Mikaela	malthorpe4@51.la	NaN	NaN	NaN
5	Magdalena	mcullip5@tiny.cc	Female	Baltimore	Maryland
6	Marietta	mheball6@blog.com	Female	Carol Stream	Illinois
7	Tine	tmcsperrin7@statcounter.com	NaN	Kansas City	Missouri
8	Enrichetta	edevillier8@ox.ac.uk	Female	Fairbanks	Alaska
9	Sari	spoulden9@xing.com	Female	New Orleans	Louisiana
10	Natale	nmartinaa@wordpress.com	Male	Bradenton	Florida

```
# Only 10 records but with all the columns
customers_info = customers.iloc[0:10 , : ]
customers_info
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city	state
0	1	Romain	Southcott	rsouthcott0@clickbank.net	Male	1.0	Trailsway	Road	San Diego	California
1	2	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	NaN	NaN	NaN	El Paso	Texas
2	3	Bambi	Westrip	bwestrip2@symantec.com	Female	4057.0	Arkansas	Circle	San Antonio	Texas
3	4	Roarke	Pankettman	rpankettman3@wiley.com	Male	74.0	Debs	Point	Memphis	Tennessee
4	5	Mikaela	Althorpe	malthorpe4@51.la	NaN	NaN	2nd	Drive	NaN	NaN
5	6	Magdalena	Cullip	mcullip5@tiny.cc	Female	9190.0	Packers	Drive	Baltimore	Maryland
6	7	Marietta	Heball	mheball6@blog.com	Female	12.0	Mallory	Center	Carol Stream	Illinois
7	8	Tine	McSperrin	tmcsperrin7@statcounter.com	NaN	530.0	Erie	Plaza	Kansas City	Missouri
8	9	Enrichetta	de Villier	edevillier8@ox.ac.uk	Female	745.0	Annamark	Street	Fairbanks	Alaska
9	10	Sari	Poulden	spoulden9@xing.com	Female	2.0	Pond	Hill	New Orleans	Louisiana

```
# Slicing using .iloc
# customers_info = customers.iloc[rows , cols]
customers_info = customers.iloc[1:11 , 1:15:2] # Starts from first_name and gives alternative columns till end
customers_info
```

	first_name	email	street_num	street_suffix	state
1	Cosimo	cmolyneaux1@wiley.com	NaN	NaN	Texas
2	Bambi	bwestrip2@symantec.com	4057.0	Circle	Texas
3	Roarke	rpankettman3@wiley.com	74.0	Point	Tennessee
4	Mikaela	malthorpe4@51.la	NaN	Drive	NaN
5	Magdalena	mcullip5@tiny.cc	9190.0	Drive	Maryland
6	Marietta	mheball6@blog.com	12.0	Center	Illinois
7	Tine	tmcsperrin7@statcounter.com	530.0	Plaza	Missouri
8	Enrichetta	edevillier8@ox.ac.uk	745.0	Street	Alaska
9	Sari	spoulden9@xing.com	2.0	Hill	Louisiana
10	Natale	nmartinaa@wordpress.com	221.0	Pass	Florida

```
# Accessing a single row
# Using .loc [Label Based]
customers.loc[999]
```

```
id                  1000
first_name         Elmore
last_name          Malpas
email                 NaN
gender               Male
street_num          205.0
street_name       Farwell
street_suffix        Park
city              Atlanta
state             Georgia
postcode          30386.0
Name: 999, dtype: object
```

```
customers.iloc[999]
```

```
id                  1000
first_name         Elmore
last_name          Malpas
email                 NaN
gender               Male
street_num          205.0
street_name       Farwell
street_suffix        Park
city              Atlanta
state             Georgia
postcode          30386.0
Name: 999, dtype: object
```

```
# Accessing a single row -> DataFrame
# Using .loc [Label Based]
customers.loc[customers['id'] == 1000] # Row Extraction [Based on a filter]
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city	state	postcode
999	1000	Elmore	Malpas	NaN	Male	205.0	Farwell	Park	Atlanta	Georgia	30386.0

```
# Boolean Result => (Filtering Rows)
customers_info = customers[customers['last_name'] == 'Sedworth']
customers_info
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city	state
554	555	Kimble	Sedworth	ksedworthfe@parallels.com	Male	1581.0	Morningstar	Trail	Denver	Colorado
631	632	Leif	Sedworth	lsedworthhj@myspace.com	Male	27823.0	Laurel	Lane	Shreveport	Louisiana
923	924	Xymenes	Sedworth	xsedworthpn@so-net.ne.jp	Male	4148.0	Lotheville	Hill	Charlotte	North Carolina

```
customers['city'].value_counts()
```

```
city
Washington          29
El Paso             19
Dallas              15
Houston             14
New York City       13
                    ..
Saint Augustine      1
Pinellas Park        1
Terre Haute          1
Stamford             1
Waterbury            1
Name: count, Length: 296, dtype: int64
```

```
# Boolean Result => (Filtering Rows)
customers_info = customers[customers['city'] == 'Washington']
customers_info
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city
13	14	Deny	Thraves	dthravesd@ibm.com	Female	71865.0	Valley Edge	Place	Washington
24	25	Tomlin	Massinger	NaN	Male	9.0	Victoria	Trail	Washington
35	36	Ernest	Armal	earmalz@dailymail.co.uk	Male	4.0	Susan	Center	Washington
36	37	Elspeth	Mendus	emendus10@ocn.ne.jp	Female	127.0	Mockingbird	Hill	Washington

```
# Boolean Result => (Filtering Rows)
customers_info = customers[customers['state'] == 'Texas']
customers_info
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city	state	postcode
1	2	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	NaN	NaN	NaN	El Paso	Texas	NaN
2	3	Bambi	Westrip	bwestrip2@symantec.com	Female	4057.0	Arkansas	Circle	San Antonio	Texas	78220.0
17	18	Reinhold	Woolforde	rwoolfordeh@cbc.ca	Male	NaN	Northwestern	Way	Amarillo	Texas	79118.0
19	20	Rubina	Hustings	rhustingsj@wikimedia.org	Female	249.0	Esker	Hill	San Antonio	Texas	78210.0
20	21	Karim	Woosnam	kwoosnamk@ifeng.com	Male	NaN	Grayhawk	Place	Houston	Texas	NaN
...	...	...	...	...	...	...	...	...	...	...	...
956	957	Ruddie	Eckhard	reckhardqk@free.fr	Male	853.0	Little Fleur	Trail	Abilene	Texas	NaN
969	970	Georgetta	Bartoszek	gbartoszekqx@is.gd	Female	4023.0	Donald	Street	NaN	Texas	79911.0
973	974	Ingmar	Muzzlewhite	NaN	Male	0.0	Carpenter	Parkway	Houston	Texas	77260.0
976	977	Corena	Pelz	NaN	Female	62987.0	Grayhawk	Plaza	Dallas	Texas	NaN
981	982	Judith	Otham	NaN	Female	96973.0	Saint Paul	Avenue	Arlington	Texas	76096.0

107 rows × 11 columns

```
# Find the Male Customers Lives in Texas
customers_info = customers[(customers['gender'] == 'Male') & (customers['state'] == 'Texas')]
customers_info
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city	st
1	2	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	NaN	NaN	NaN	El Paso	Te
17	18	Reinhold	Woolforde	rwoolfordeh@cbc.ca	Male	NaN	Northwestern	Way	Amarillo	Te
20	21	Karim	Woosnam	kwoosnamk@ifeng.com	Male	NaN	Grayhawk	Place	Houston	Te
25	26	Franklin	Goodge	fgoodgep@mapquest.com	Male	97670.0	Kensington	Center	Houston	Te
39	40	Melvyn	Seifert	mseifert13@photobucket.com	Male	NaN	Park Meadow	Pass	San Antonio	Te
42	43	Bendicty	Sunnex	bsunnex16@elpais.com	Male	39695.0	North	Place	Irving	Te
43	44	Garrett	Bartram	gbartram17@sun.com	Male	15.0	Everett	Way	Houston	Te
46	47	Burch	Issacov	bissacov1a@tiny.cc	Male	54399.0	NaN	NaN	Dallas	Te
50	51	Correy	Raatz	[illegible]	Male	661.0	Truax	Trail	Dallas	Te

```
customers_info.shape
```

```
(46, 11)
```

```
male_filter = (customers['gender'] == 'Male')
texas_filter = (customers['state'] == 'Texas')
customers_info = customers[male_filter & texas_filter]
customers_info
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city
1	2	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	NaN	NaN	NaN	El Paso
17	18	Reinhold	Woolforde	rwoolfordeh@cbc.ca	Male	NaN	Northwestern	Way	Amarillo
20	21	Karim	Woosnam	kwoosnamk@ifeng.com	Male	NaN	Grayhawk	Place	Houston
25	26	Franklin	Goodge	fgoodgep@mapquest.com	Male	97670.0	Kensington	Center	Houston
39	40	Melvyn	Seifert	mseifert13@photobucket.com	Male	NaN	Park Meadow	Pass	San Antonio
42	43	Bendicty	Sunnex	bsunnex16@elpais.com	Male	39695.0	North	Place	Irving
43	44	Garrett	Bartram	gbartram17@sun.com	Male	15.0	Everett	Way	Houston
46	47	Burch	Issacov	bissacov1a@tiny.cc	Male	54399.0	NaN	NaN	Dallas

```
products[products['company'].isna()]
```

	Unnamed: 0	id	product	cost	company
15	15	16	Sausage - Chorizo	$55.45	NaN
21	21	22	Scotch - Queen Anne	$60.26	NaN
26	26	27	Spaghetti Squash	$47.41	NaN
27	27	28	Wine - Niagara,vqa Reisling	$28.24	NaN
45	45	46	Aromat Spice / Seasoning	$61.34	NaN

```
Skipfire_filter = products['company'] == 'Skipfire'
Ntag_filter = products['company'] == 'Ntag'
Livepath_filter = products['company'] == 'Livepath'
Zoombox_filter = products['company'] == 'Zoombox'
products[Skipfire_filter | Ntag_filter | Livepath_filter |  Zoombox_filter]
```

	Unnamed: 0	id	product	cost	company
0	0	1	Liners - Baking Cups	$6.36	Skipfire
2	2	3	Bar Bran Honey Nut	$65.40	Ntag
5	5	6	Wine - White, Riesling, Semi - Dry	$99.22	Livepath
9	9	10	Sambuca - Ramazzotti	$88.99	Livepath
22	22	23	Puree - Blackcurrant	$16.27	Zoombox
30	30	31	Wine - Carmenere Casillero Del	$55.77	Ntag

```
Boolean Filters Summary :
df[df['column'] == 'value']  # Equals
df[(df['column'] == 'value') | (df['column'] == 'value')] # OR Logic
df[(df['column'] == 'value') & (df['column'] == 'value')] # And Logic
df[~(df['column'] == 'value')] # Not Logic
```

```
customers[customers['gender'] == 'Male']
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city
0	1	Romain	Southcott	rsouthcott0@clickbank.net	Male	1.0	Trailsway	Road	San Diego
1	2	Cosimo	Molyneaux	cmolyneaux1@wiley.com	Male	NaN	NaN	NaN	El Paso
3	4	Roarke	Pankettman	rpankettman3@wiley.com	Male	74.0	Debs	Point	Memphis
10	11	Natale	Martina	nmartinaa@wordpress.com	Male	221.0	Sauthoff	Pass	Bradenton
12	13	Mikol	MacWhan	mmacwhanc@patch.com	Male	9.0	Eagle Crest	Center	Bronx
...	...	...	...	...	...	...	...	...	...
984	985	Tailor	Sealeaf	tsealeafrc@irs.gov	Male	NaN	Summer Ridge	Pass	Beaufort
986	987	Eric	Mountford	emountfordre@blogspot.com	Male	NaN	6th	Court	Las Vegas
997	998	Tobit	Birt	NaN	Male	2.0	Basil	Road	Waterbury
998	999	Issiah	Standbrooke	istandbrookerq@yellowpages.com	Male	NaN	Kenwood	Drive	Savannah
999	1000	Elmore	Malpas	NaN	Male	205.0	Farwell	Park	Atlanta

485 rows × 11 columns

```
customers[~(customers['gender'] == 'Male')] # Not Logic
```

	id	first_name	last_name	email	gender	street_num	street_name	street_suffix	city	state
2	3	Bambi	Westrip	bwestrip2@symantec.com	Female	4057.0	Arkansas	Circle	San Antonio	Texas
4	5	Mikaela	Althorpe	malthorpe4@51.la	NaN	NaN	2nd	Drive	NaN	NaN
5	6	Magdalena	Cullip	mcullip5@tiny.cc	Female	9190.0	Packers	Drive	Baltimore	Maryland
6	7	Marietta	Heball	mheball6@blog.com	Female	12.0	Mallory	Center	Carol Stream	Illinois
7	8	Tine	McSperrin	tmcsperrin7@statcounter.com	NaN	530.0	Erie	Plaza	Kansas City	Missouri
...	...	...	...	...	...	...	...	...	...	...
992	993	Johnath	Clancy	jclancyrk@smugmug.com	Female	7.0	Washington	Crossing	Juneau	Alaska
993	994	Binnie	Dearth	bdearthrl@ed.gov	NaN	8993.0	Elgar	Trail	Minneapolis	Minnesota
994	995	Brana	Dixon	bdixonrm@myspace.com	Female	97.0	Truax	Avenue	Maple Plain	Minnesota
995	996	Merrili	Alman	malmanrn@cornell.edu	Female	0.0	Thompson	Way	Reading	Pennsylvania
996	997	Winonah	Heckle	whecklero@fc2.com	Female	701.0	Rowland	Hill	Indianapolis	Indiana

515 rows × 11 columns

```
# Like Keyword in SQL
# LIKE %Nut% , '%nut%' , '%NuT%'
products[products['product'].str.contains('NUT', case = False)]
```

	Unnamed: 0	id	product	cost	company
2	2	3	Bar Bran Honey Nut	$65.40	Ntag
57	57	58	Bar Bran Honey Nut	$73.05	Yakijo
58	58	59	Nut - Almond, Blanched, Whole	$74.28	Eazzy

```
# Like Keyword in SQL
# LIKE 'Nut%' -> Startswith
products[products['product'].str.startswith('Nut')]
```

	Unnamed: 0	id	product	cost	company
58	58	59	Nut - Almond, Blanched, Whole	$74.28	Eazzy

```
# Like Keyword in SQL
# LIKE '#Nut' -> Endswith
products[products['product'].str.endswith('Nut')]
```

	Unnamed: 0	id	product	cost	company
2	2	3	Bar Bran Honey Nut	$65.40	Ntag
57	57	58	Bar Bran Honey Nut	$73.05	Yakijo