In [68]:

```
df.head() #this is used to fetch the top part of df
```

Out[68]:

	State	year	pop
0	Mumbai	2001	1.5
1	Pune	2011	1.6
2	Chennai	2018	1.7
3	Nagpur	2019	3.6
4	Delhi	2020	3.2

In [66]:

```
df
```

Out[66]:

	State	year	pop
0	Mumbai	2001	1.5
1	Pune	2011	1.6
2	Chennai	2018	1.7
3	Nagpur	2019	3.6
4	Delhi	2020	3.2

In [67]:

```
df.tail(2)  #fetching the last part of dataframe
```

Out[67]:

	State	year	pop
3	Nagpur	2019	3.6
4	Delhi	2020	3.2

In [ ]:

```

```

22nd August 2021

In [69]:

```
df1 = pd.DataFrame(data,index=['one','two','three','four','five'])
```

In [70]:

```
df1
```

Out[70]:

	State	year	pop
one	Mumbai	2001	1.5
two	Pune	2011	1.6
three	Chennai	2018	1.7
four	Nagpur	2019	3.6
five	Delhi	2020	3.2

In [71]:

```
df1['State'] #fetching data on the basis of columns
```

Out[71]:

```
one       Mumbai
two         Pune
three    Chennai
four      Nagpur
five       Delhi
Name: State, dtype: object
```

In [ ]:

```

```

In [72]:

```
df1['year']
```

Out[72]:

```
one      2001
two      2011
three    2018
four     2019
five     2020
Name: year, dtype: int64
```

In [73]:

```
df1['pop']
```

Out[73]:

```
one      1.5
two      1.6
three    1.7
four     3.6
five     3.2
Name: pop, dtype: float64
```

In [74]:

```
df1.columns #names of your columns
```

Out[74]:

```
Index(['State', 'year', 'pop'], dtype='object')
```

In [77]:

```
df1
```

Out[77]:

	State	year	pop
one	Mumbai	2001	1.5
two	Pune	2011	1.6
three	Chennai	2018	1.7
four	Nagpur	2019	3.6
five	Delhi	2020	3.2

In [76]:

```
df1.loc['four'] #fetching the data on the basis of rows
```

Out[76]:

```
State    Nagpur
year       2019
pop         3.6
Name: four, dtype: object
```

In [78]:

```
df1
```

Out[78]:

	State	year	pop
one	Mumbai	2001	1.5
two	Pune	2011	1.6
three	Chennai	2018	1.7
four	Nagpur	2019	3.6
five	Delhi	2020	3.2

In [79]:

```
df1['country'] = 'India'  #create columns and provide relevant values
```

In [81]:

```
df1
```

Out[81]:

	State	year	pop	country
one	Mumbai	2001	1.5	India
two	Pune	2011	1.6	India
three	Chennai	2018	1.7	India
four	Nagpur	2019	3.6	India
five	Delhi	2020	3.2	India

In [82]:

```
df1
```

Out[82]:

	State	year	pop	country
one	Mumbai	2001	1.5	India
two	Pune	2011	1.6	India
three	Chennai	2018	1.7	India
four	Nagpur	2019	3.6	India
five	Delhi	2020	3.2	India

In [ ]:

```

```

In [84]:

```
df1.iloc[2] #fetching the data on the basis of indexed location (by default)
```

Out[84]:

```
State      Chennai
year          2018
pop            1.7
country      India
Name: three, dtype: object
```

In [ ]:

```

```

In [85]:

```
df1['debt'] = np.arange(1,6)
```

In [86]:

```
df1
```

Out[86]:

	State	year	pop	country	debt
one	Mumbai	2001	1.5	India	1
two	Pune	2011	1.6	India	2
three	Chennai	2018	1.7	India	3
four	Nagpur	2019	3.6	India	4
five	Delhi	2020	3.2	India	5

In [87]:

```
df1
```

Out[87]:

	State	year	pop	country	debt
one	Mumbai	2001	1.5	India	1
two	Pune	2011	1.6	India	2
three	Chennai	2018	1.7	India	3
four	Nagpur	2019	3.6	India	4
five	Delhi	2020	3.2	India	5

In [ ]:

```
1
```

In [89]:

```
1 del df1['country']
```

In [91]:

```
1 df1
```

Out[91]:

	State	year	pop	debt
one	Mumbai	2001	1.5	1
two	Pune	2011	1.6	2
three	Chennai	2018	1.7	3
four	Nagpur	2019	3.6	4
five	Delhi	2020	3.2	5

In [92]:

```
1 df1['country'] = 'India'
```

In [93]:

```
1 df1
```

Out[93]:

	State	year	pop	debt	country
one	Mumbai	2001	1.5	1	India
two	Pune	2011	1.6	2	India
three	Chennai	2018	1.7	3	India
four	Nagpur	2019	3.6	4	India
five	Delhi	2020	3.2	5	India

In [103]:

```
1 df1['budget'] = np.arange(100,600,100)
```

In [104]:

```
1 df1
```

Out[104]:

	State	year	pop	debt	country	budget
one	Mumbai	2001	1.5	1	India	100
two	Pune	2011	1.6	2	India	200
three	Chennai	2018	1.7	3	India	300
four	Nagpur	2019	3.6	4	India	400
five	Delhi	2020	3.2	5	India	500

In [106]:

```
1 df1['budget']
```

Out[106]:

```
one      100
two      200
three    300
four     400
five     500
Name: budget, dtype: int64
```

In [107]:

```
1 df1['budget']['three'] = 3000
```

```
<ipython-input-107-7d0d039475f7>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retu
rning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-vi
ew-versus-a-copy)
  df1['budget']['three'] = 3000
```

In [108]:

```
df1
```

Out[108]:

	State	year	pop	debt	country	budget
one	Mumbai	2001	1.5	1	India	100
two	Pune	2011	1.6	2	India	200
three	Chennai	2018	1.7	3	India	3000
four	Nagpur	2019	3.6	4	India	400
five	Delhi	2020	3.2	5	India	500

In [ ]:

```

```

index objects

In [109]:

```
obj = pd.Series(range(3),index=['a','b','c'])
```

In [110]:

```
obj
```

Out[110]:

```
a    0
b    1
c    2
dtype: int64
```

In [111]:

```
obj.index
```

Out[111]:

```
Index(['a', 'b', 'c'], dtype='object')
```

In [112]:

```
df1['State']
```

Out[112]:

```
one       Mumbai
two         Pune
three    Chennai
four      Nagpur
five       Delhi
Name: State, dtype: object
```

In [113]:

```
df1['State'].index
```

Out[113]:

```
Index(['one', 'two', 'three', 'four', 'five'], dtype='object')
```

In [ ]:

```

```

In [114]:

```
val = df1['State'].index
```

In [115]:

```
val
```

Out[115]:

```
Index(['one', 'two', 'three', 'four', 'five'], dtype='object')
```

In [116]:

```
val[0]
```

Out[116]:

```
'one'
```

In [117]:

```
1 val[1]
```

Out[117]:

```
'two'
```

In [118]:

```
1 val[0] = 'ten'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-118-1543a5572cdf> in <module>
----> 1 val[0] = 'ten'

~/anaconda3/lib/python3.8/site-packages/pandas/core/indexes/base.py in __setitem__(self, key, value)
   4275     @final
   4276     def __setitem__(self, key, value):
-> 4277         raise TypeError("Index does not support mutable operations")
   4278
   4279     def __getitem__(self, key):

TypeError: Index does not support mutable operations
```

In [ ]:

```
1
```

Re-indexing

In [119]:

```
1 ser = pd.Series([4.1,7.5,-8.3,3.6],index=['d','b','a','c'])
```

In [120]:

```
1 ser
```

Out[120]:

```
d    4.1
b    7.5
a   -8.3
c    3.6
dtype: float64
```

In [121]:

```
1 ser.reindex(['a','b','c','d'])
```

Out[121]:

```
a   -8.3
b    7.5
c    3.6
d    4.1
dtype: float64
```

In [ ]:

```
1
```

In [132]:

```
1 obj = pd.Series(['blue','purple','yellow'],index=[1,2,4])
```

In [133]:

```
1 obj
```

Out[133]:

```
1      blue
2    purple
4    yellow
dtype: object
```

In [ ]:

```
1
```

In [134]:

```
1 new_obj = obj.reindex(range(6),method='ffill')
```

In [135]:

```
1 new_obj
```

Out[135]:

```
0       NaN
1      blue
2    purple
3    purple
4    yellow
5    yellow
dtype: object
```

In [ ]:

```
1
```

Droping an Entry from Series

In [136]:

```
1 myser = pd.Series(np.arange(5),index=['a','b','c','d','e'])
```

In [137]:

```
1 myser
```

Out[137]:

```
a    0
b    1
c    2
d    3
e    4
dtype: int64
```

In [138]:

```
1 myser.drop('c')
```

Out[138]:

```
a    0
b    1
d    3
e    4
dtype: int64
```

In [139]:

```
1 myser
```

Out[139]:

```
a    0
b    1
c    2
d    3
e    4
dtype: int64
```

In [140]:

```
1 myser.drop(['b','d','e'])
```

Out[140]:

```
a    0
c    2
dtype: int64
```

In [141]:

```
1 myser
```

Out[141]:

```
a    0
b    1
c    2
d    3
e    4
dtype: int64
```

In [ ]:

```
1
```

In [142]:

```
df2 = pd.DataFrame(np.arange(16).reshape((4,4)),
                   index = ['Mumbai','Pune','Chennai','Delhi'],
                   columns=['one','two','three','four']
                  )
```

In [143]:

```
df2
```

Out[143]:

	one	two	three	four
Mumbai	0	1	2	3
Pune	4	5	6	7
Chennai	8	9	10	11
Delhi	12	13	14	15

In [144]:

```
df2.drop(['Chennai','Delhi'])
```

Out[144]:

	one	two	three	four
Mumbai	0	1	2	3
Pune	4	5	6	7

In [145]:

```
df2
```

Out[145]:

	one	two	three	four
Mumbai	0	1	2	3
Pune	4	5	6	7
Chennai	8	9	10	11
Delhi	12	13	14	15

In [ ]:

```

```

Dropping a column

In [146]:

```
df2.drop('three',axis=1)
```

Out[146]:

	one	two	four
Mumbai	0	1	3
Pune	4	5	7
Chennai	8	9	11
Delhi	12	13	15

In [ ]:

```

```

In [148]:

```
df2.drop(['one','four'],axis=1) #dropping multiple columns
```

Out[148]:

	two	three
Mumbai	1	2
Pune	5	6
Chennai	9	10
Delhi	13	14

In [ ]:

```
1
```

In [149]:

```
1 df2
```

Out[149]:

	one	two	three	four
Mumbai	0	1	2	3
Pune	4	5	6	7
Chennai	8	9	10	11
Delhi	12	13	14	15

In [150]:

```
1 df2.drop('Pune',inplace=True)
```

In [151]:

```
1 df2
```

Out[151]:

	one	two	three	four
Mumbai	0	1	2	3
Chennai	8	9	10	11
Delhi	12	13	14	15

In [ ]:

```
1
```

In [152]:

```
1 df2.drop('four',axis=1,inplace=True)
```

In [153]:

```
1 df2
```

Out[153]:

	one	two	three
Mumbai	0	1	2
Chennai	8	9	10
Delhi	12	13	14

In [ ]:

```
1
```

indexing , selection & filtering

In [154]:

```
1 ser = pd.Series(np.arange(4),index=['a','b','c','d'])
```

In [155]:

```
1 ser
```

Out[155]:

```
a    0
b    1
c    2
d    3
dtype: int64
```

In [156]:

```
1 ser[1:3] #deault indexing
```

Out[156]:

```
b    1
c    2
dtype: int64
```

In [159]:

```
ser['b':'c'] #custom indexing
```

Out[159]:

```
b    1
c    2
dtype: int64
```

In [ ]:

```

```

In [ ]:

```

```

In [160]:

```
ser['b':'c']
```

Out[160]:

```
b    1
c    2
dtype: int64
```

In [161]:

```
ser['b':'c'] = 10
```

In [162]:

```
ser
```

Out[162]:

```
a     0
b    10
c    10
d     3
dtype: int64
```

In [163]:

```
ser[1:3] = 2
```

In [164]:

```
ser
```

Out[164]:

```
a    0
b    2
c    2
d    3
dtype: int64
```

In [ ]:

```

```

In [165]:

```
df2
```

Out[165]:

	one	two	three
Mumbai	0	1	2
Chennai	8	9	10
Delhi	12	13	14

In [166]:

```
df2['one']
```

Out[166]:

```
Mumbai      0
Chennai     8
Delhi      12
Name: one, dtype: int64
```

In [168]:

```
df2['two']
```

Out[168]:

```
Mumbai      1
Chennai     9
Delhi      13
Name: two, dtype: int64
```

In [169]:

```
df2['three']
```

Out[169]:

```
Mumbai      2
Chennai    10
Delhi      14
Name: three, dtype: int64
```

In [ ]:

```

```

In [171]:

```
df2[['one','three']] #fetch more than 1 col at the same time
```

Out[171]:

	one	three
Mumbai	0	2
Chennai	8	10
Delhi	12	14

In [ ]:

```

```

In [ ]:

```

```

In [172]:

```
df3 = pd.DataFrame(np.arange(9).reshape((3,3)),
                   columns=['b','c','d'],
                   index=['Mumbai','Pune','Nagpur']
                  )
```

In [173]:

```
df3
```

Out[173]:

	b	c	d
Mumbai	0	1	2
Pune	3	4	5
Nagpur	6	7	8

In [174]:

```
df4 = pd.DataFrame(np.arange(12).reshape(4,3),
                   columns=['b','d','e'],
                     index=['Delhi','Mumbai','Pune','Chennai']
                   )
```

In [177]:

```
df3
```

Out[177]:

	b	c	d
Mumbai	0	1	2
Pune	3	4	5
Nagpur	6	7	8

In [175]:

```
df4
```

Out[175]:

	b	d	e
Delhi	0	1	2
Mumbai	3	4	5
Pune	6	7	8
Chennai	9	10	11

In [178]:

```
df5 = df3 + df4
```

In [179]:

```
df5
```

Out[179]:

	b	c	d	e
Chennai	NaN	NaN	NaN	NaN
Delhi	NaN	NaN	NaN	NaN
Mumbai	3.0	NaN	6.0	NaN
Nagpur	NaN	NaN	NaN	NaN
Pune	9.0	NaN	12.0	NaN

In [183]:

```
df5.fillna(True)
```

Out[183]:

	b	c	d	e
Chennai	True	True	True	True
Delhi	True	True	True	True
Mumbai	3.0	True	6.0	True
Nagpur	True	True	True	True
Pune	9.0	True	12.0	True

In [181]:

```
df5
```

Out[181]:

	b	c	d	e
Chennai	NaN	NaN	NaN	NaN
Delhi	NaN	NaN	NaN	NaN
Mumbai	3.0	NaN	6.0	NaN
Nagpur	NaN	NaN	NaN	NaN
Pune	9.0	NaN	12.0	NaN

In [ ]:

```

```