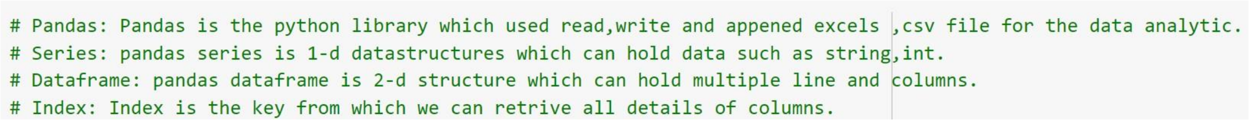
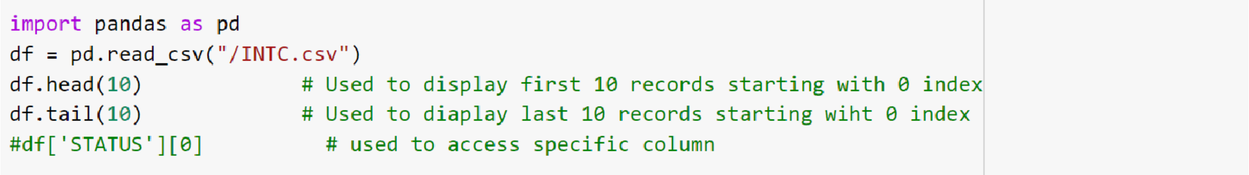
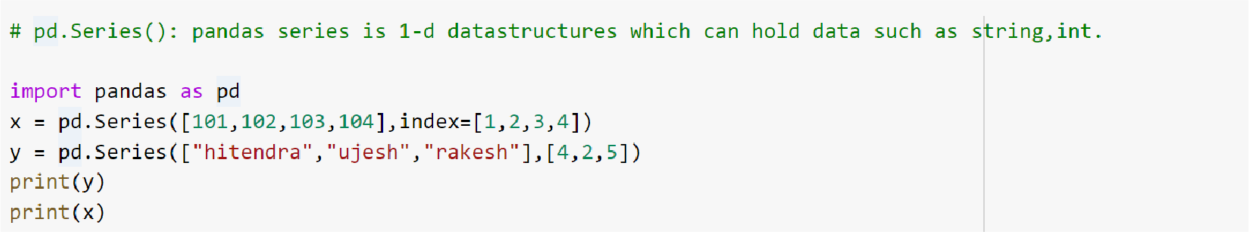
Pandas



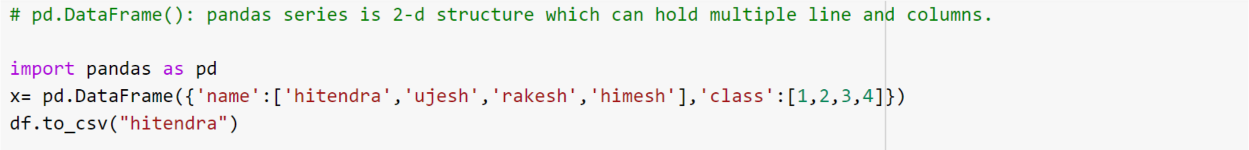
```
# Pandas: Pandas is the python library which used read,write and appened excels ,csv file for the data analytic.
# Series: pandas series is 1-d datastructures which can hold data such as string,int.
# Dataframe: pandas dataframe is 2-d structure which can hold multiple line and columns.
# Index: Index is the key from which we can retrive all details of columns.
```



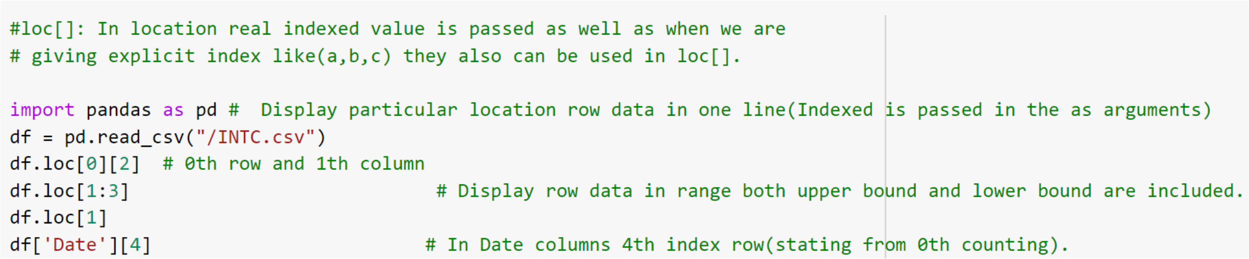
```
import pandas as pd
df = pd.read_csv("/INTC.csv")
df.head(10)             # Used to display first 10 records starting with 0 index
df.tail(10)             # Used to diaplay last 10 records starting wiht 0 index
#df['STATUS'][0]          # used to access specific column
```



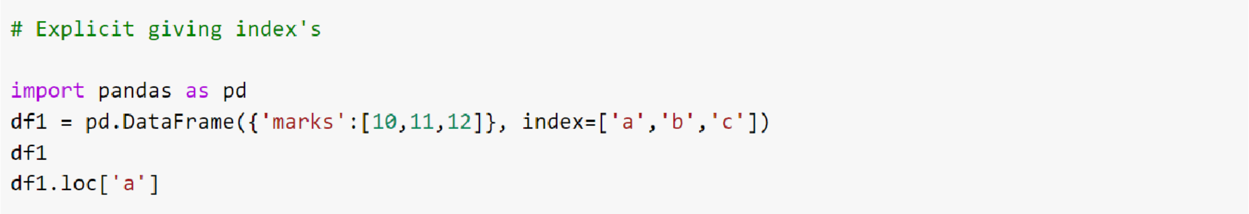
```
# pd.Series(): pandas series is 1-d datastructures which can hold data such as string,int.

import pandas as pd
x = pd.Series([101,102,103,104],index=[1,2,3,4])
y = pd.Series(["hitendra","ujesh","rakesh"],[4,2,5])
print(y)
print(x)
```



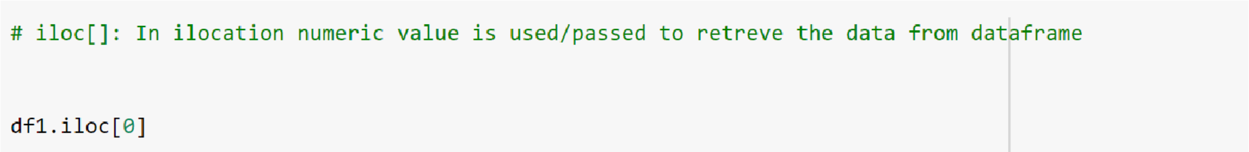
```
# pd.DataFrame(): pandas series is 2-d structure which can hold multiple line and columns.

import pandas as pd
x= pd.DataFrame({'name':['hitendra','ujesh','rakesh','himesh'],'class':[1,2,3,4]})
df.to_csv("hitendra")
```



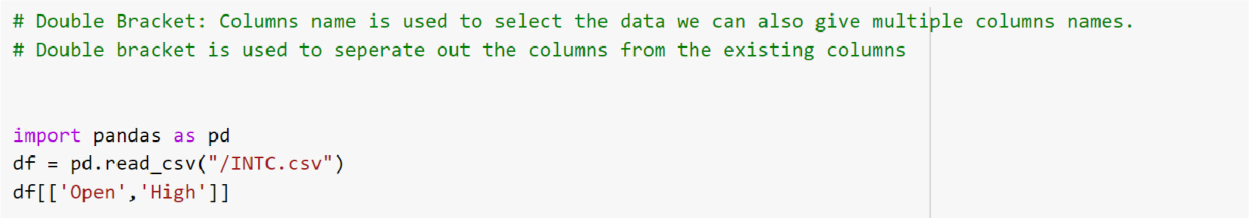
```
#loc[]: In location real indexed value is passed as well as when we are
# giving explicit index like(a,b,c) they also can be used in loc[].

import pandas as pd #  Display particular location row data in one line(Indexed is passed in the as arguments)
df = pd.read_csv("/INTC.csv")
df.loc[0][2]  # 0th row and 1th column
df.loc[1:3]                           # Display row data in range both upper bound and lower bound are included.
df.loc[1]
df['Date'][4]                        # In Date columns 4th index row(stating from 0th counting).
```



```
# Explicit giving index's

import pandas as pd
df1 = pd.DataFrame({'marks':[10,11,12]}, index=['a','b','c'])
df1
df1.loc['a']
```

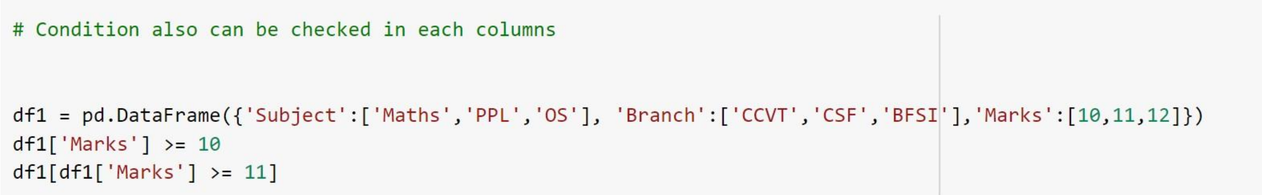


```
# iloc[]: In ilocation numeric value is used/passed to retreve the data from dataframe


df1.iloc[0]
```

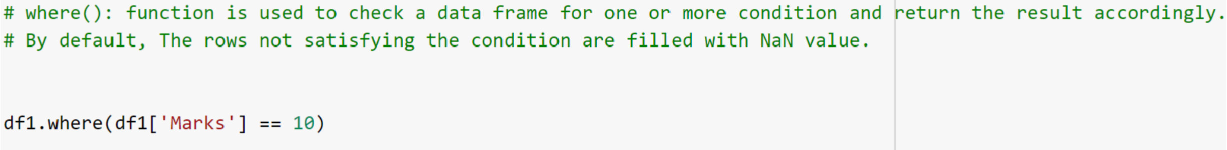


```
# Double Bracket: Columns name is used to select the data we can also give multiple columns names.
# Double bracket is used to seperate out the columns from the existing columns


import pandas as pd
df = pd.read_csv("/INTC.csv")
df[['Open','High']]
```

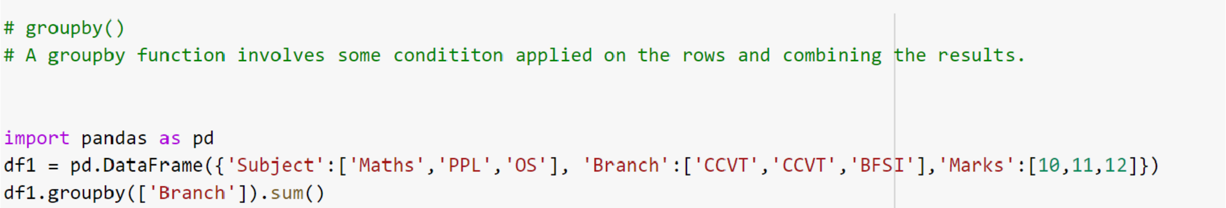
Pandas



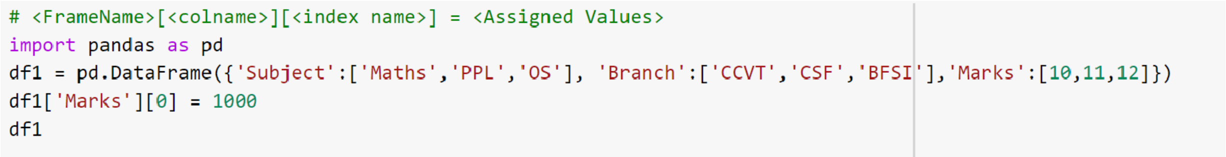
```
# Condition also can be checked in each columns


df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
df1['Marks'] >= 10
df1[df1['Marks'] >= 11]
```



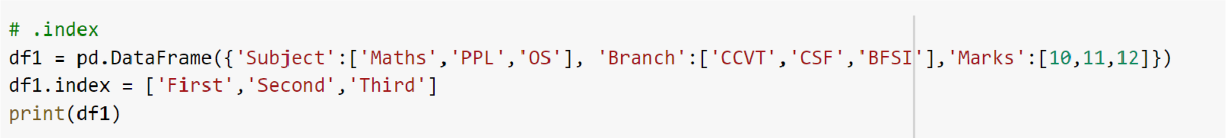
```
# where(): function is used to check a data frame for one or more condition and return the result accordingly.
# By default, The rows not satisfying the condition are filled with NaN value.


df1.where(df1['Marks'] == 10)
```



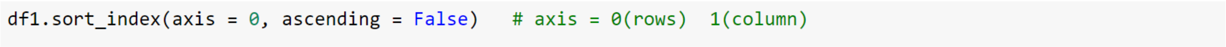
```
# groupby()
# A groupby function involves some condititon applied on the rows and combining the results.


import pandas as pd
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CCVT','BFSI'],'Marks':[10,11,12]})
df1.groupby(['Branch']).sum()
```



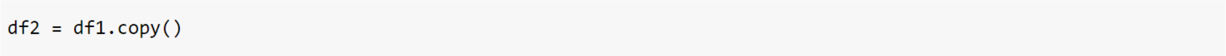
```
# <FrameName>[<colname>][<index name>] = <Assigned Values>
import pandas as pd
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
df1['Marks'][0] = 1000
df1
```



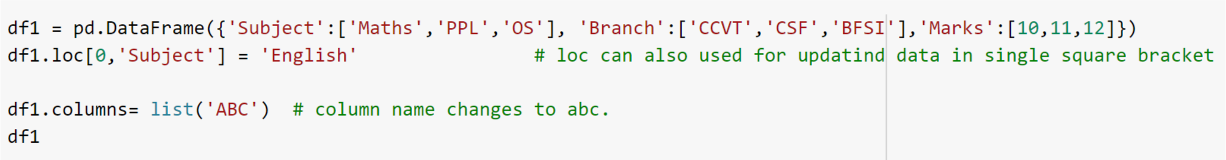
```
# .index
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
df1.index = ['First','Second','Third']
print(df1)
```



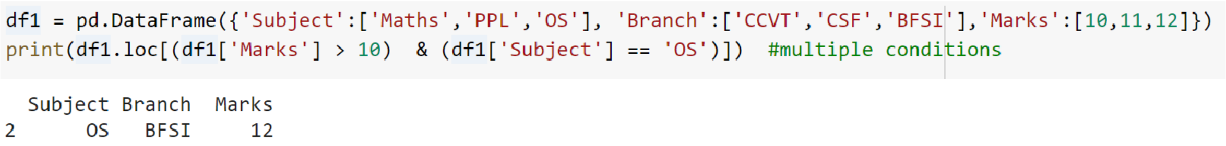
```
df1.sort_index(axis = 0, ascending = False)   # axis = 0(rows)  1(column)
```



```
df2 = df1.copy()
```

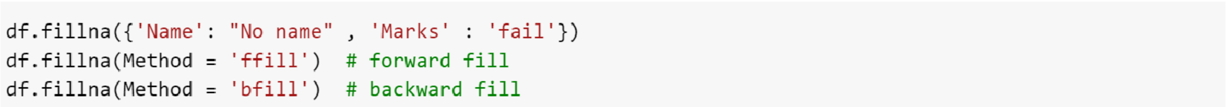


```
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
df1.loc[0,'Subject'] = 'English'                # loc can also used for updatind data in single square bracket

df1.columns= list('ABC')  # column name changes to abc.
df1
```



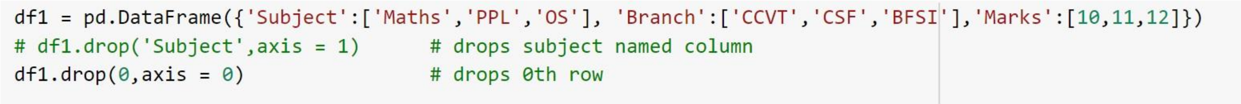
```
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
print(df1.loc[(df1['Marks'] > 10)  & (df1['Subject'] == 'OS')])  #multiple conditions
```

```
  Subject Branch  Marks
2      OS   BFSI     12
```

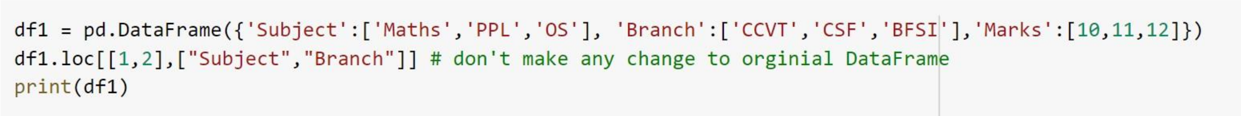


```
df.fillna({'Name': "No name" , 'Marks' : 'fail'})
df.fillna(Method = 'ffill')  # forward fill
df.fillna(Method = 'bfill')  # backward fill
```

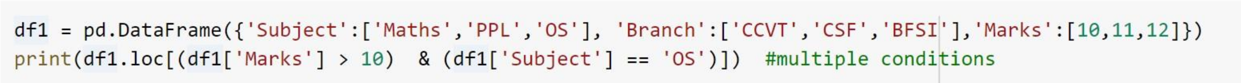
Pandas



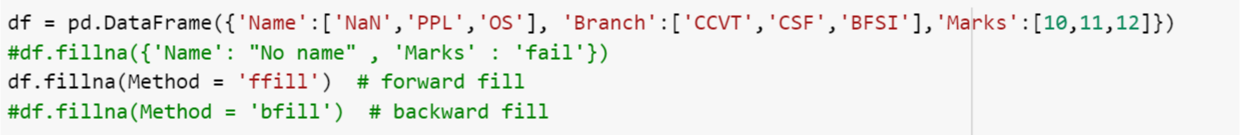
```
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
# df1.drop('Subject',axis = 1)      # drops subject named column
df1.drop(0,axis = 0)                # drops 0th row
```



```
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
df1.loc[[1,2],["Subject","Branch"]] # don't make any change to orginial DataFrame
print(df1)
```



```
df1 = pd.DataFrame({'Subject':['Maths','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
print(df1.loc[(df1['Marks'] > 10)  & (df1['Subject'] == 'OS')])  #multiple conditions
```



```
df = pd.DataFrame({'Name':['NaN','PPL','OS'], 'Branch':['CCVT','CSF','BFSI'],'Marks':[10,11,12]})
#df.fillna({'Name': "No name" , 'Marks' : 'fail'})
df.fillna(Method = 'ffill')  # forward fill
#df.fillna(Method = 'bfill')  # backward fill
```