PANDAS

The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals [Wikipedia]

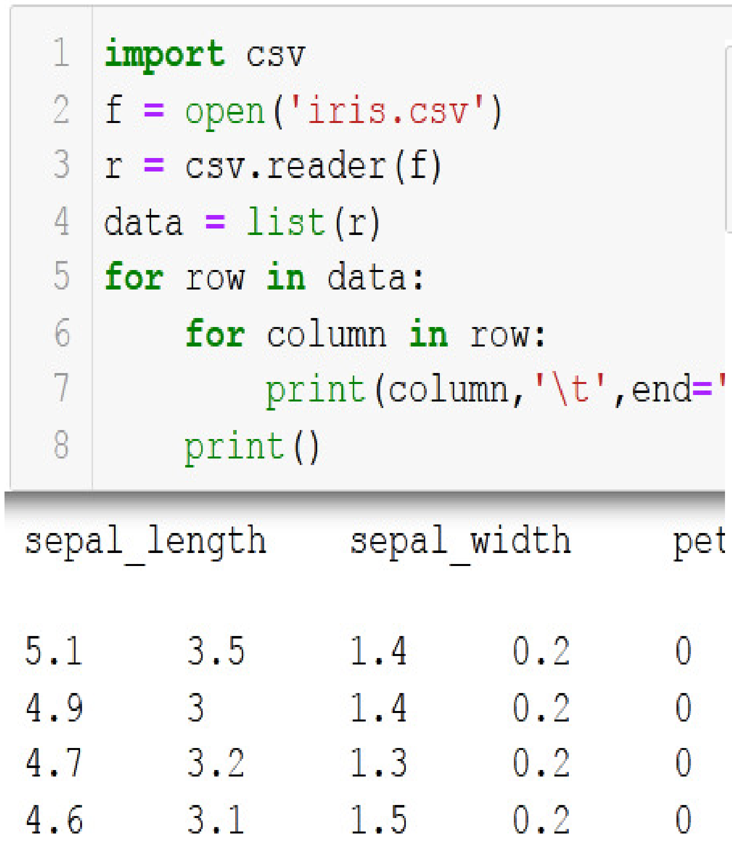
What is Pandas?

- Widely used open-source python library which is built on top of Numpy.
- Provides high performance easy to use structures for data analysis.
- Can perform operations like Data Cleaning, Transforming and Analyzing.
- Calculate statistics and it answers, questions about data.

What we can do with Pandas?

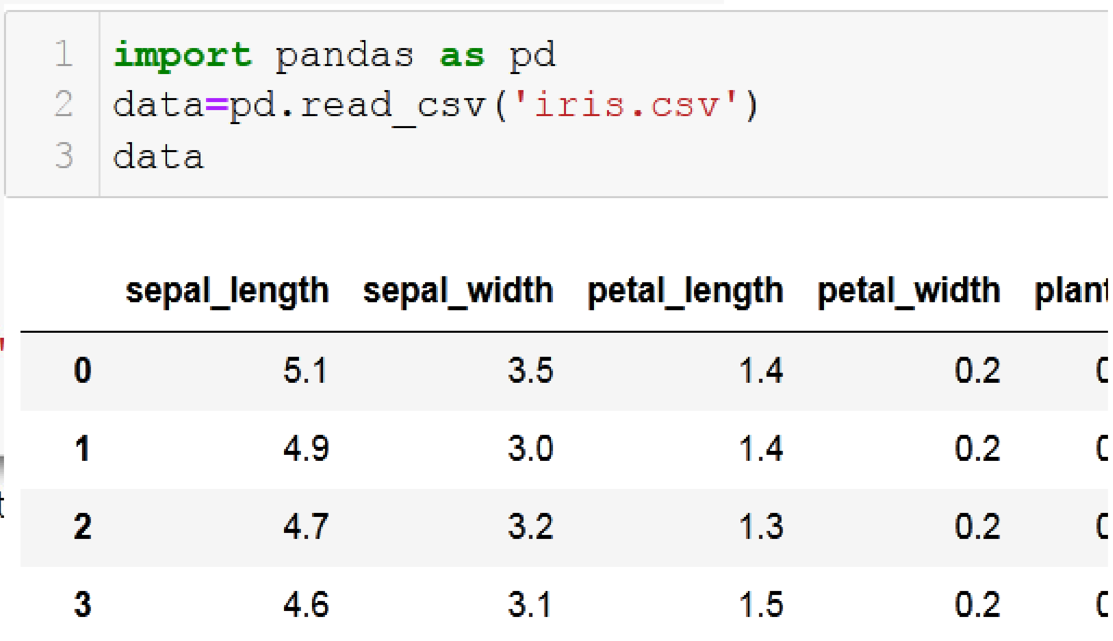
- Read and write files in different formats like csv,tsv,text,XML, JSON,ZIP etc.
- Check information and description about our data.
- Filter data.
- Handle missing values and noise.
- Do aggregation and summarization.
- Merge and concatenate datasets.

Pandas Vs. Python



```python
import csv
f = open('iris.csv')
r = csv.reader(f)
data = list(r)
for row in data:
    for column in row:
        print(column,'\t',end='
    print()
```

```
sepal_length        sepal_width        pet

5.1       3.5       1.4       0.2       0
4.9       3         1.4       0.2       0
4.7       3.2       1.3       0.2       0
4.6       3.1       1.5       0.2       0
```

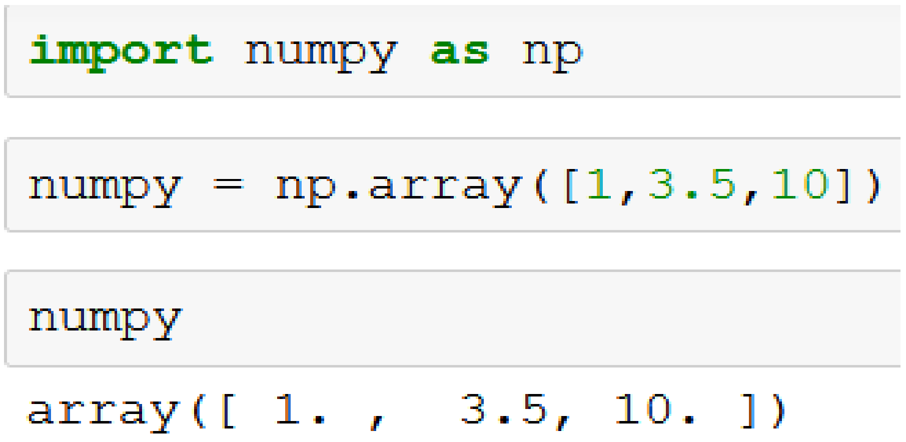


```python
import pandas as pd
data=pd.read_csv('iris.csv')
data
```

	sepal_length	sepal_width	petal_length	petal_width	plan
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0

Pandas Vs. Numpy

- **Numpy can handle homogenous data**



```
import numpy as np
```

```
numpy = np.array([1,3.5,10])
```

```
numpy
```

```
array([ 1. ,  3.5, 10. ])
```

- **But pandas can handle heterogeneous data**

Pandas (Series)

- Pandas uses three data structures to hold user data.
- 1. Series
- 2. Data Frame
- 3. Panel

Pandas (Series)

- 1. Series : 1D labeled array
- Series can hold heterogeneous values (int, float, string etc.)
- Series == Column

Pandas (Series)

Series 1	Series 2	Series 3
Name	ID Number	Marks
Priyang	123	98
Aadhya	124	99
Parshv	125	97
Vedant	126	99
Krisha	127	88

Pandas (Data Frame)

- 2. Data Frame : 2D data structure , Holds a data into table-like format.
- Data frames consist of the row, column and data

Name	ID Number	Marks
Priyang	123	98
Aadhya	124	99
Parshv	125	97
Vedant	126	99
Krisha	127	88

Pandas (Data Frame)

Name
Priyang
Aadhya
Parshv
Vedant
Krisha

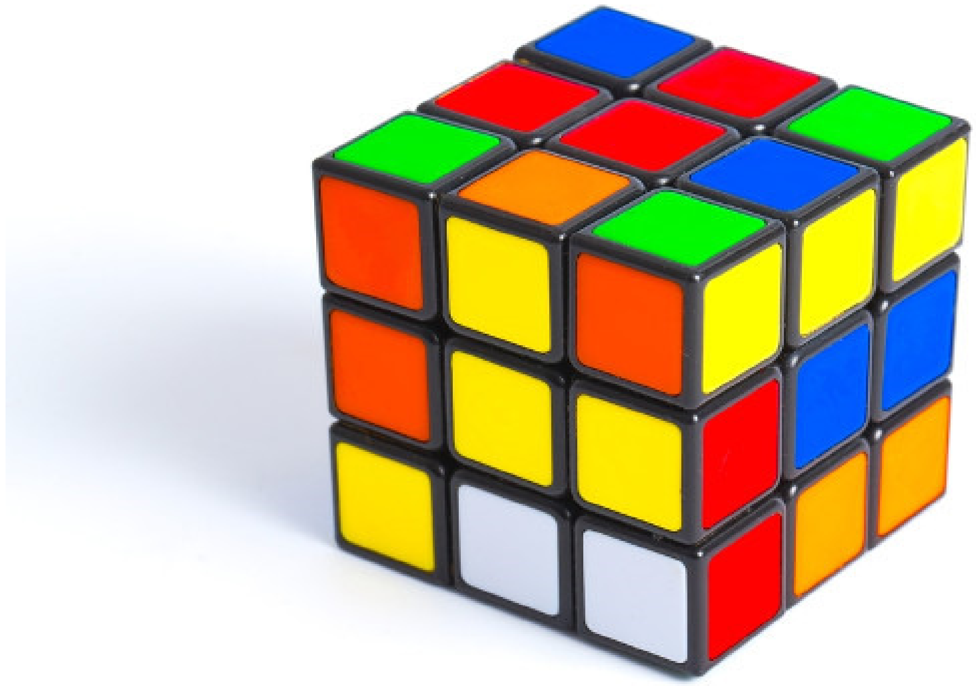
ID Number
123
124
125
126
127

Marks
98
99
97
99
88

Name	ID Number	Marks
Priyang	123	98
Aadhya	124	99
Parshv	125	97
Vedant	126	99
Krisha	127	88

Pandas (Panel)

- **3. Panel :** **3D data structure , Holds heterogeneous data**



Pandas (Series)

- 1. Series : 1D labeled array
- Series can hold heterogeneous values (int, float, string etc.)
- Series == Column

Pandas (Series)

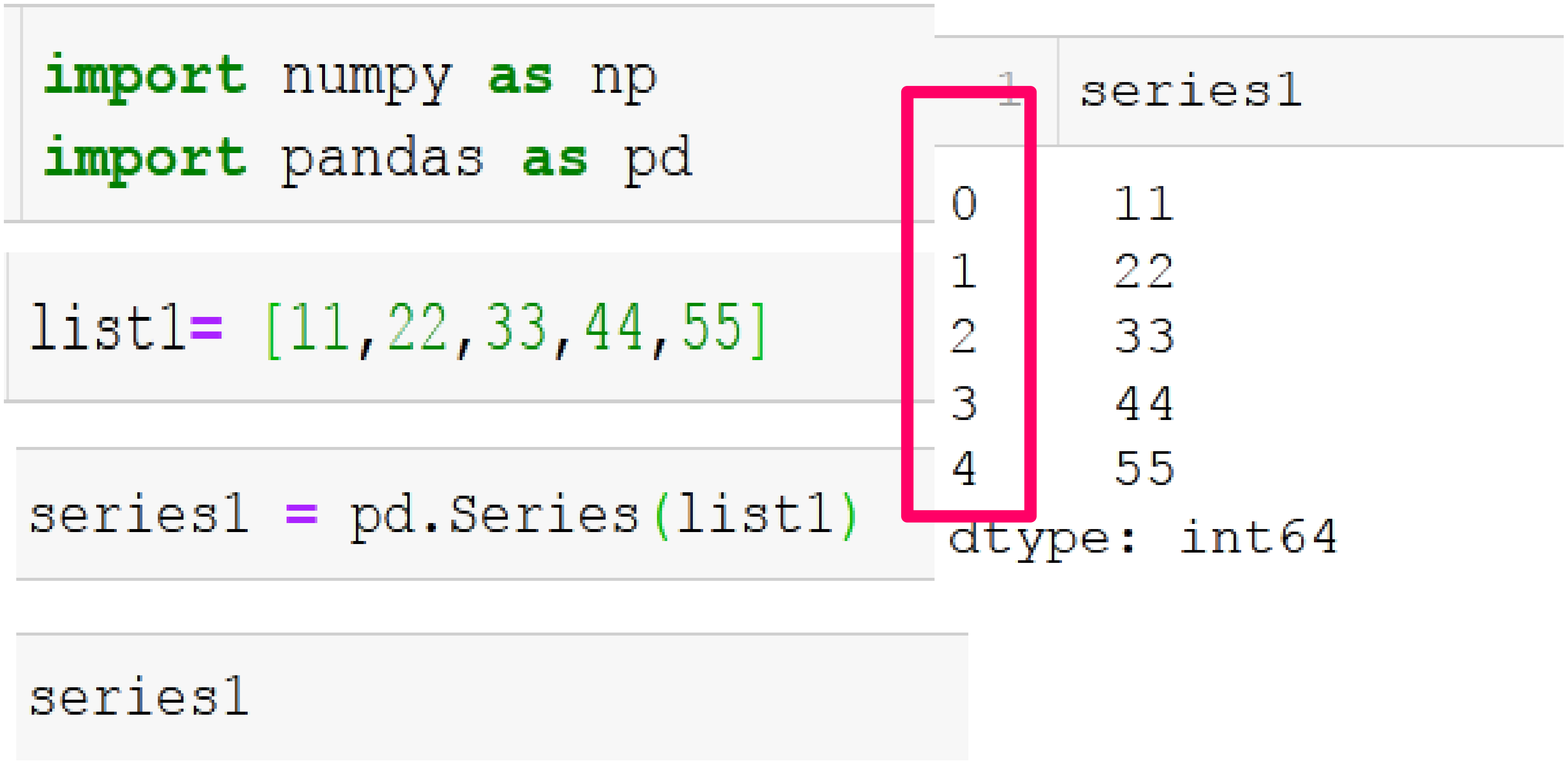
Series 1	Series 2	Series 3
Name	ID Number	Marks
Priyang	123	98
Aadhya	124	99
Parshv	125	97
Vedant	126	99
Krisha	127	88

Pandas (Series)

Pandas series can be created using

- List
- Tuple
- Dictionary
- Numpy array
- Scalar value

Pandas (Series using List)



```python
import numpy as np
import pandas as pd
```

```python
list1= [11,22,33,44,55]
```

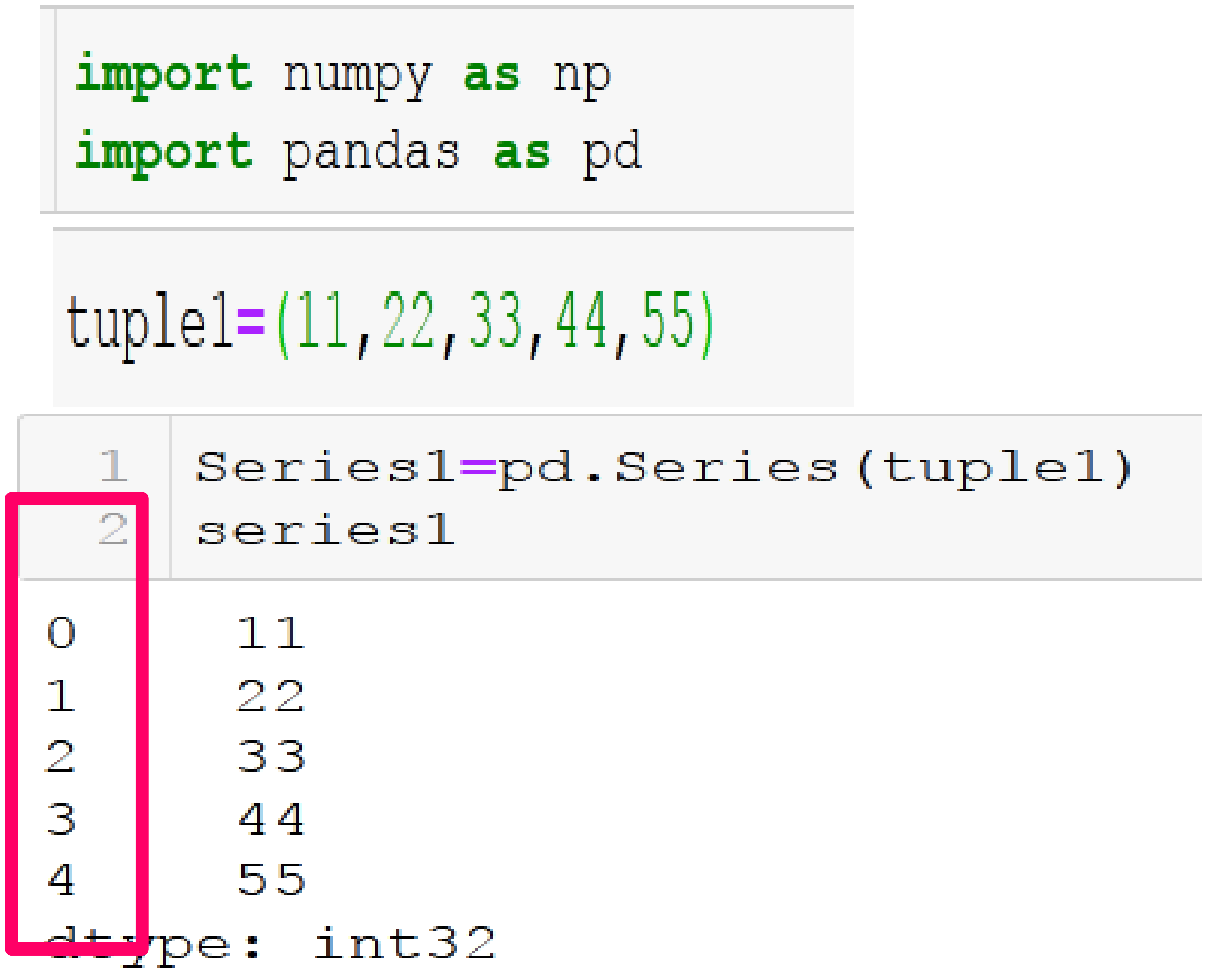
```python
series1 = pd.Series(list1)
```

```python
series1
```

```python
series1
```

```
0    11
1    22
2    33
3    44
4    55
dtype: int64
```

Pandas (Series using Tuple)



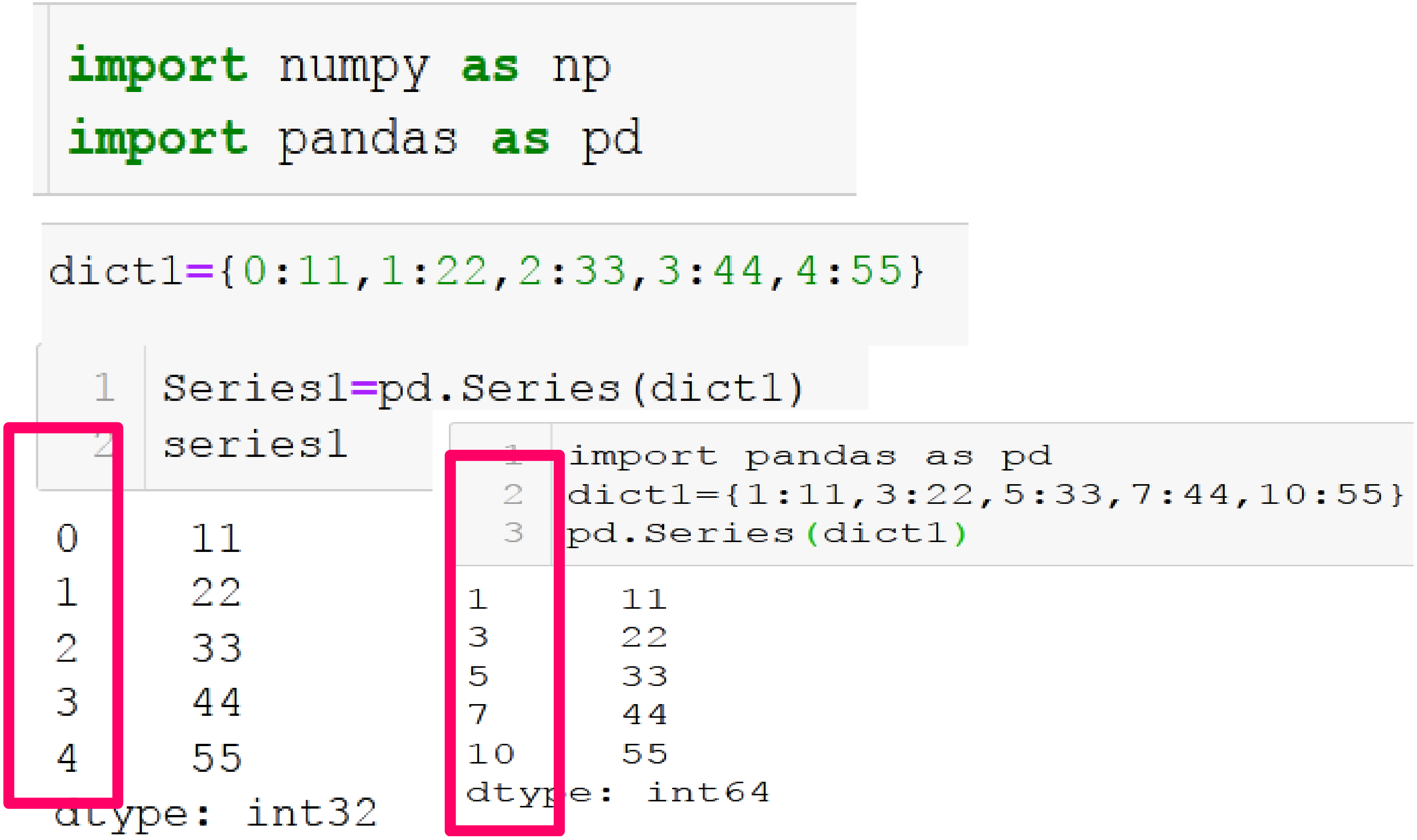
```python
import numpy as np
import pandas as pd
```

```python
tuple1=(11,22,33,44,55)
```

```python
Series1=pd.Series(tuple1)
series1
```

```
0      11
1      22
2      33
3      44
4      55
dtype: int32
```

Pandas (Series using Dictionary)



```
import numpy as np
import pandas as pd
```

```
dict1={0:11,1:22,2:33,3:44,4:55}
```

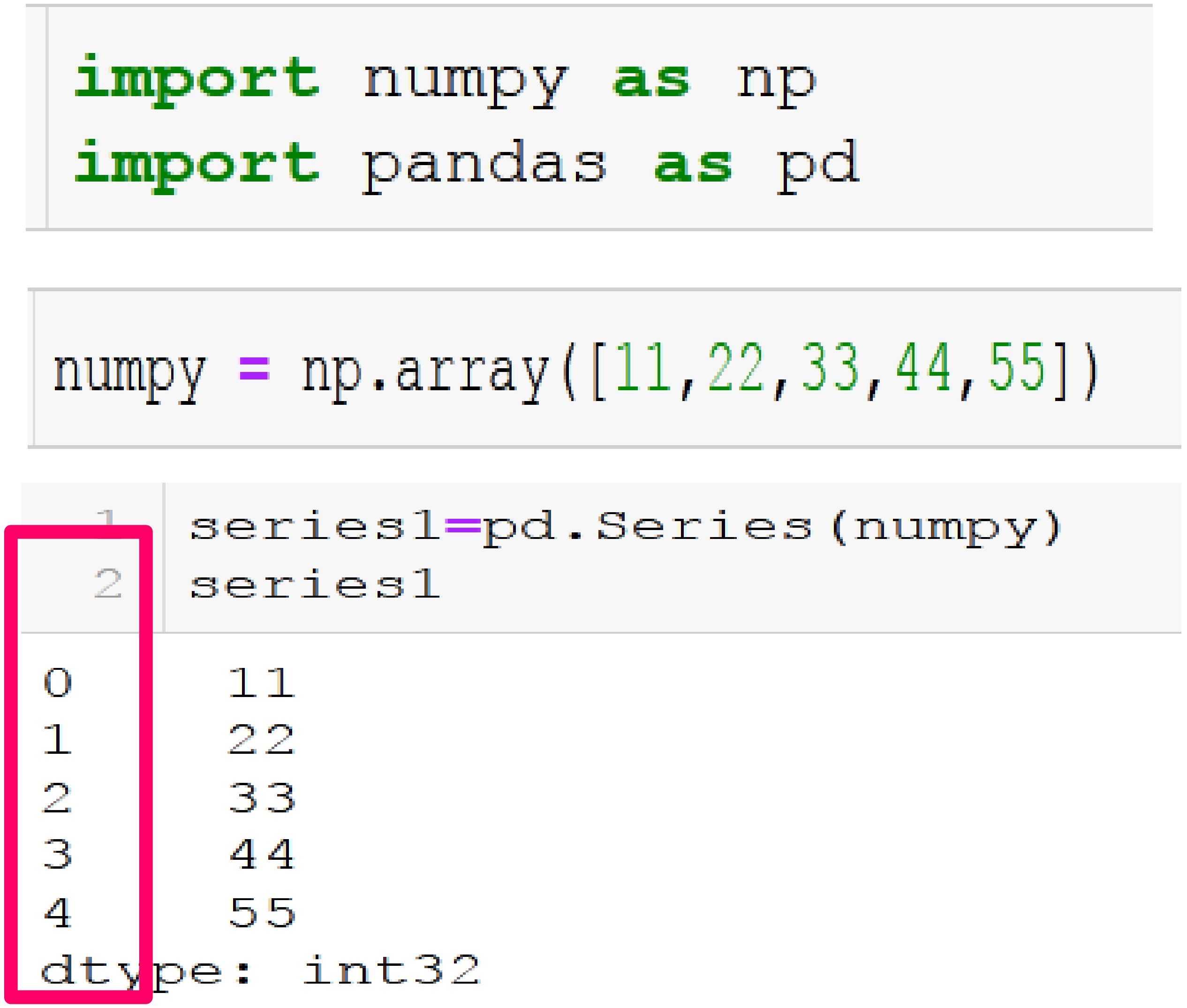
```
Series1=pd.Series(dict1)
series1
```

```
0     11
1     22
2     33
3     44
4     55
dtype: int32
```

```
import pandas as pd
dict1={1:11,3:22,5:33,7:44,10:55}
pd.Series(dict1)
```

```
1     11
3     22
5     33
7     44
10    55
dtype: int64
```

Pandas (Series using Numpy)



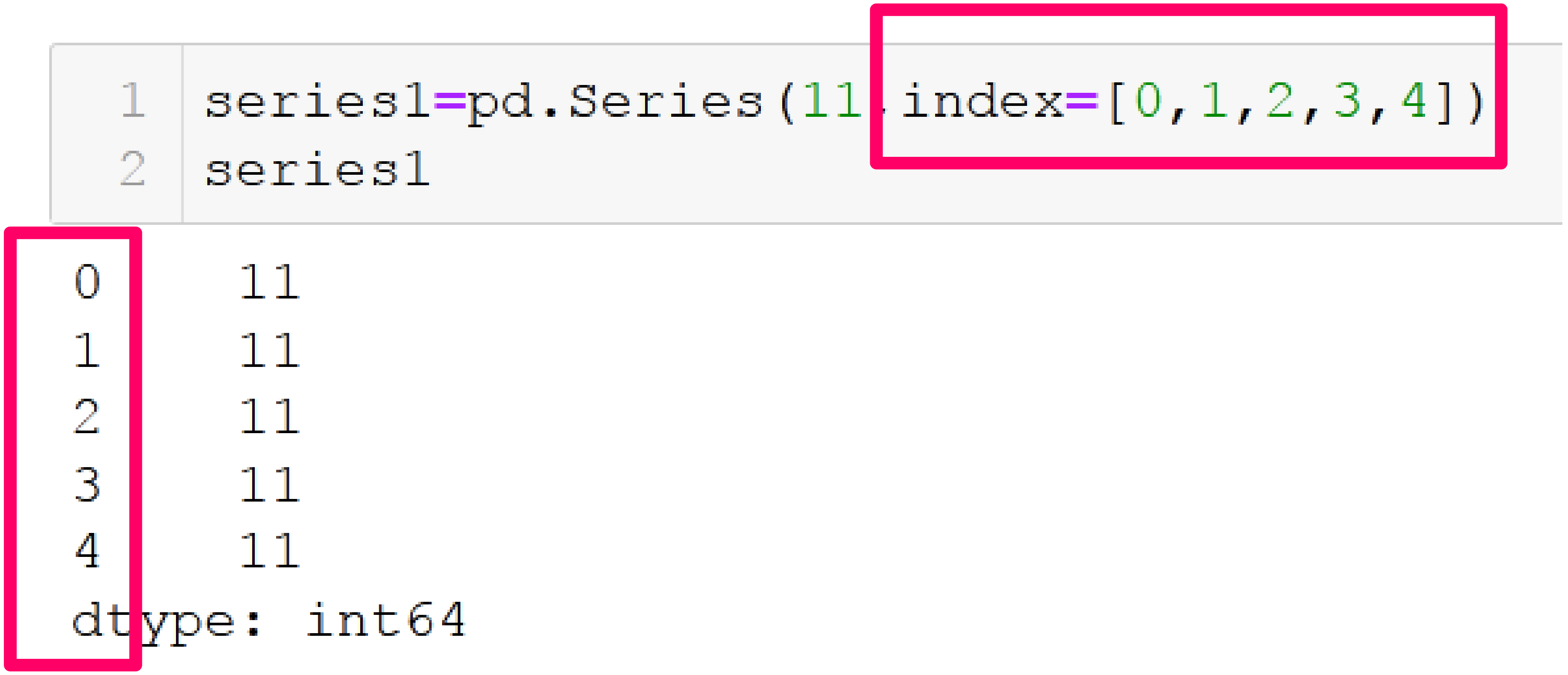
```
import numpy as np
import pandas as pd
```

```
numpy = np.array([11,22,33,44,55])
```

```
series1=pd.Series(numpy)
series1
```

```
0    11
1    22
2    33
3    44
4    55
dtype: int32
```

Pandas (Series using Scalar value)



```
series1=pd.Series(11,index=[0,1,2,3,4])
series1
```

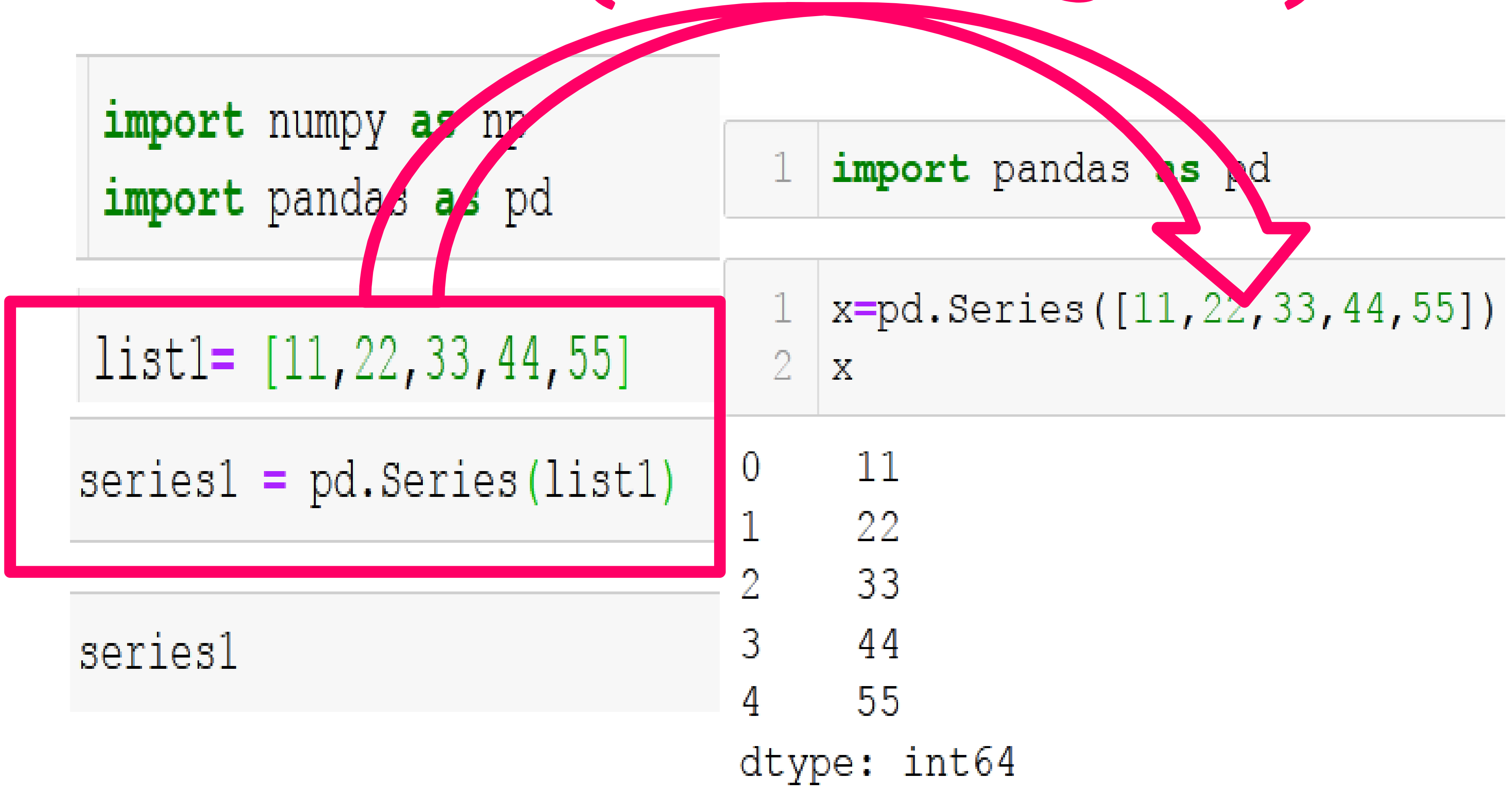
```
0    11
1    11
2    11
3    11
4    11
dtype: int64
```

Pandas (Series)

Pandas series can be created using

- List
- Tuple
- Dictionary
- Numpy array
- Scalar value

Pandas (Series using List)



```python
import numpy as np
import pandas as pd
```

```python
list1= [11,22,33,44,55]
```

```python
series1 = pd.Series(list1)
```

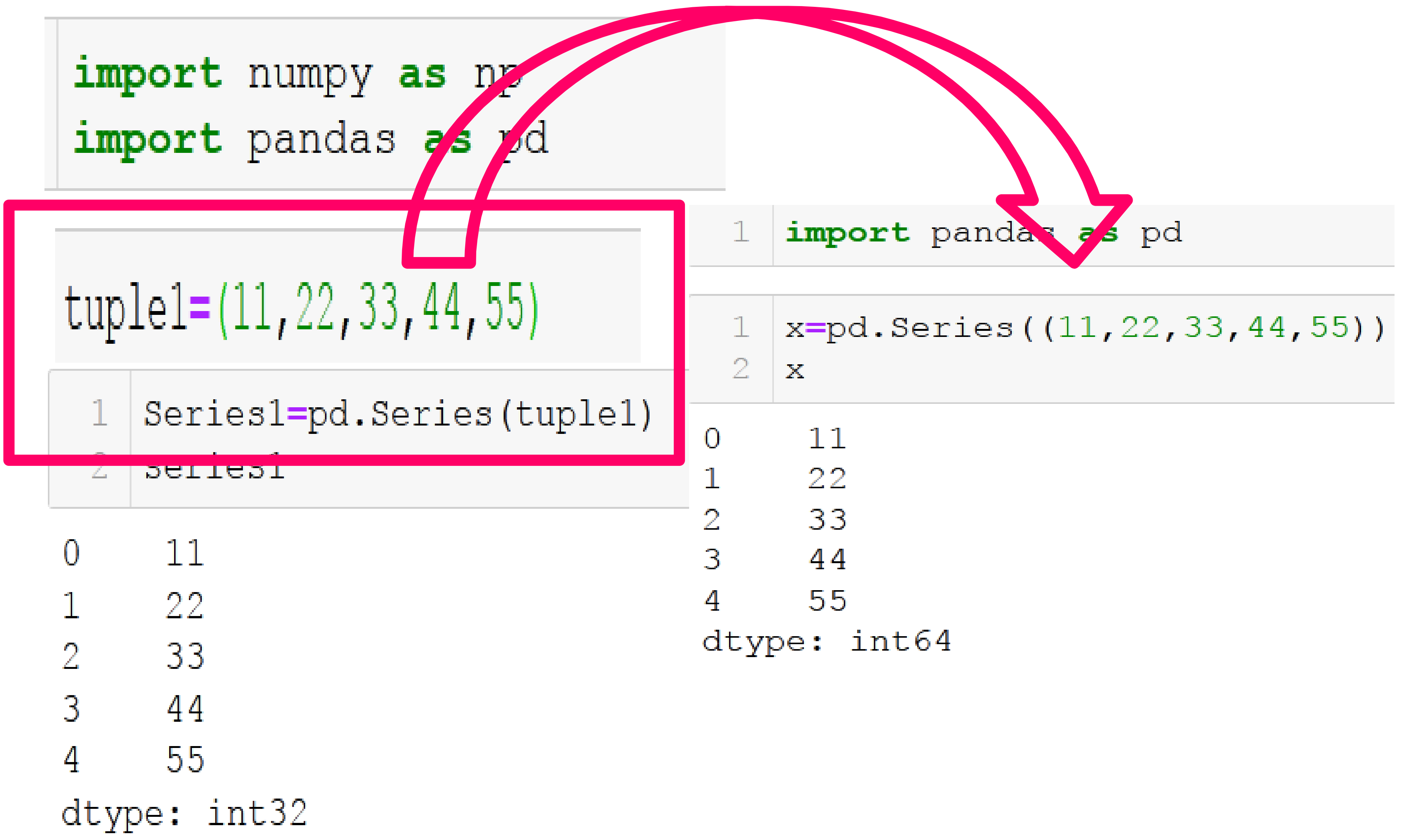
```python
series1
```

```python
import pandas as pd
```

```python
x=pd.Series([11,22,33,44,55])
x
```

```
0    11
1    22
2    33
3    44
4    55
dtype: int64
```

Pandas (Series using Tuple)



```python
import numpy as np
import pandas as pd
```

```python
tuple1=(11,22,33,44,55)
```

```python
Series1=pd.Series(tuple1)
Series1
```

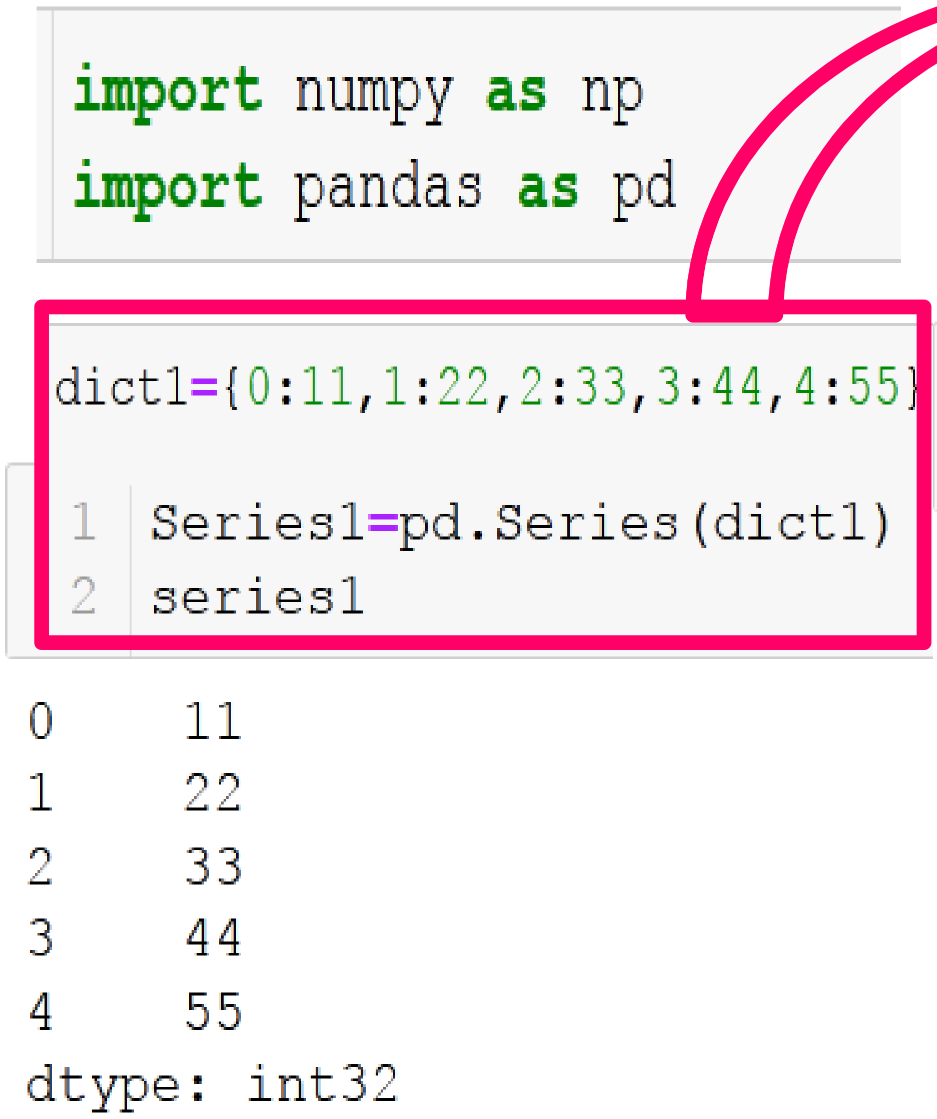
```
0     11
1     22
2     33
3     44
4     55
dtype: int32
```

```python
import pandas as pd
```

```python
x=pd.Series((11,22,33,44,55))
x
```

```
0     11
1     22
2     33
3     44
4     55
dtype: int64
```

Pandas (Series using Dictionary)

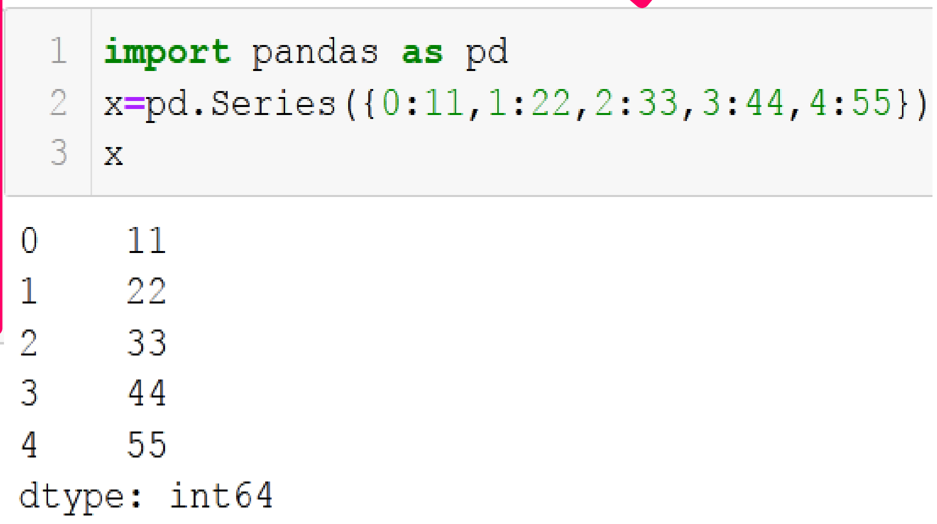


```
import numpy as np
import pandas as pd
```

```
dict1={0:11,1:22,2:33,3:44,4:55}
```

```
Series1=pd.Series(dict1)
series1
```

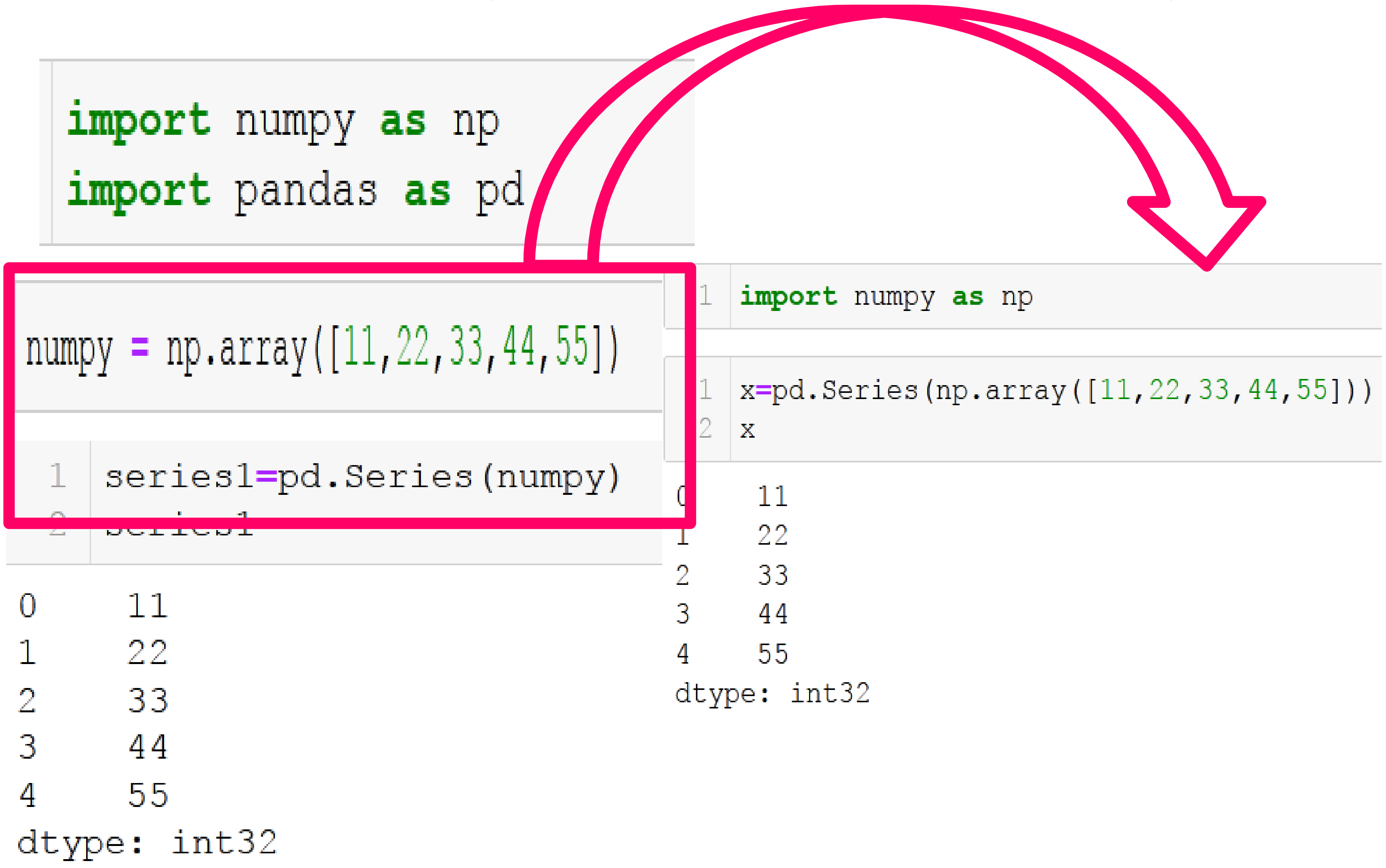
```
0      11
1      22
2      33
3      44
4      55
dtype: int32
```



```
import pandas as pd
x=pd.Series({0:11,1:22,2:33,3:44,4:55})
x
```

```
0     11
1     22
2     33
3     44
4     55
dtype: int64
```

Pandas (Series using Numpy)



```python
import numpy as np
import pandas as pd
```

```python
numpy = np.array([11,22,33,44,55])
```

```python
series1=pd.Series(numpy)
series1
```

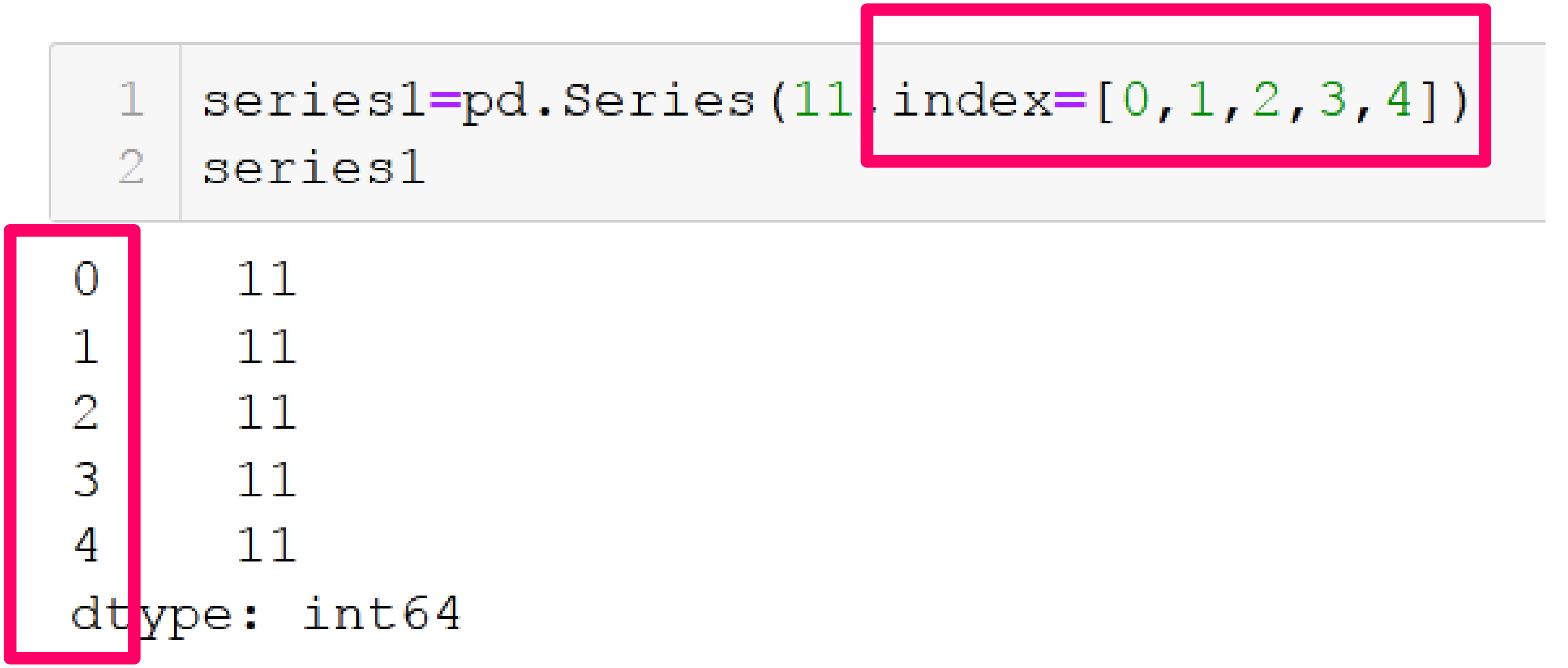
```
0      11
1      22
2      33
3      44
4      55
dtype: int32
```

```python
import numpy as np
```

```python
x=pd.Series(np.array([11,22,33,44,55]))
x
```

```
0     11
1     22
2     33
3     44
4     55
dtype: int32
```

Pandas (Series using Scalar value)



```
series1=pd.Series(11,index=[0,1,2,3,4])
series1
```

```
0     11
1     11
2     11
3     11
4     11
dtype: int64
```

Question

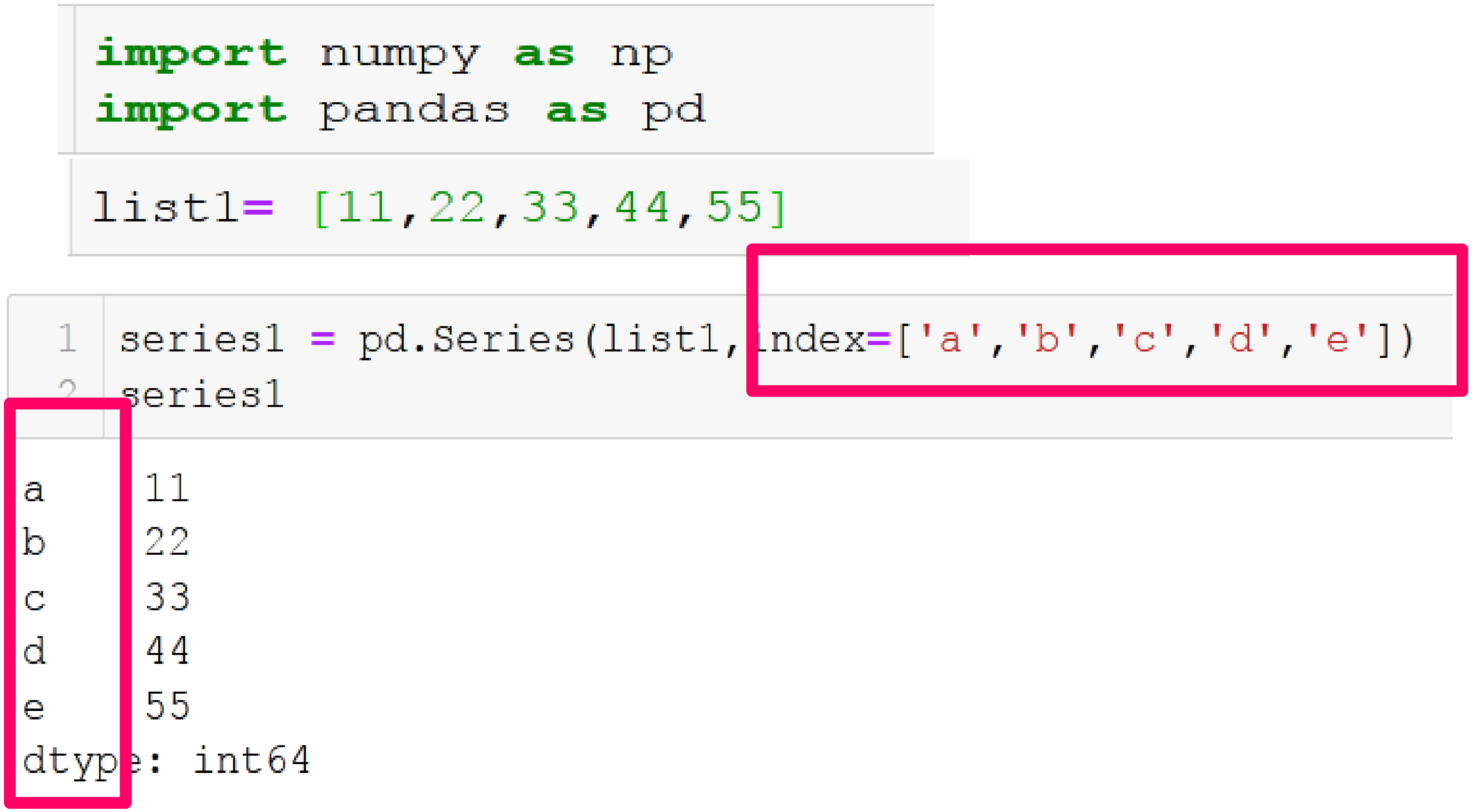
Create Pandas Series Using Python Dictionary

```
A     100
B     200
C     300
D     400
E     500
dtype: int64
```

Pandas Series

- Pandas (Series using List (with indexes))
- Pandas (Series using List (with dtype))
- Accessing the data from series
- Operations on Pandas Series

Pandas (Series using List (with indexes))



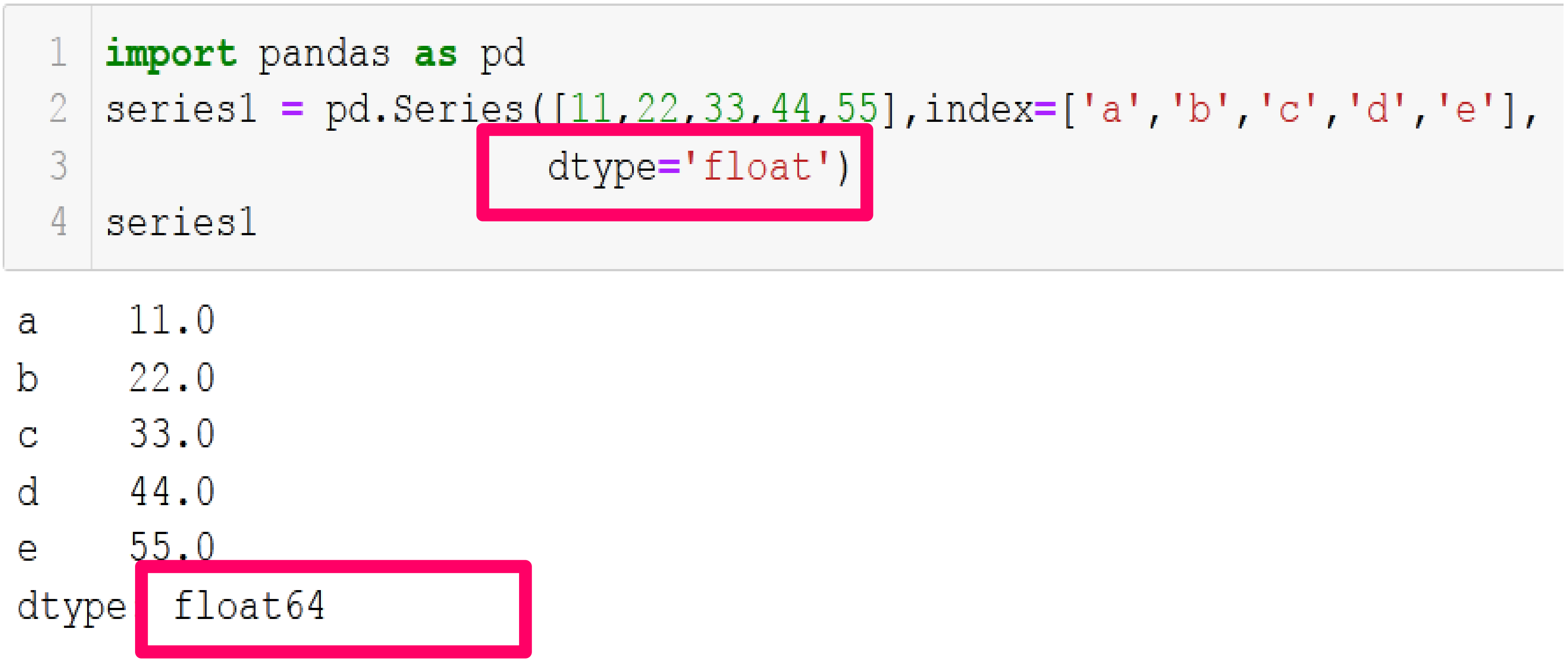
```
import numpy as np
import pandas as pd
```

```
list1= [11,22,33,44,55]
```

```
series1 = pd.Series(list1,index=['a','b','c','d','e'])
series1
```

```
a     11
b     22
c     33
d     44
e     55
dtype: int64
```

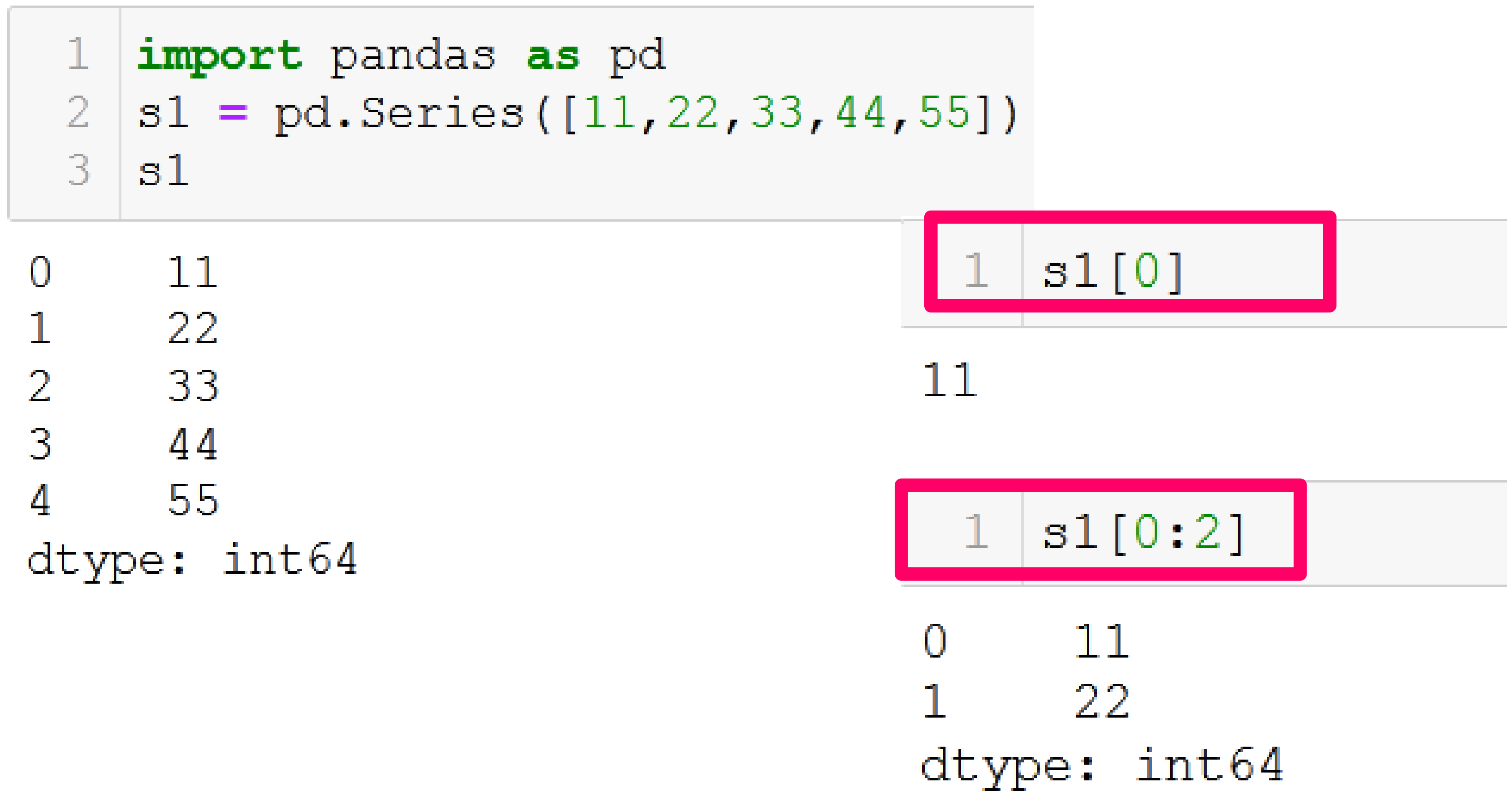
Pandas (Series using List (with dtype))



```
import pandas as pd
series1 = pd.Series([11,22,33,44,55],index=['a','b','c','d','e'],
                                  dtype='float')
series1
```

```
a     11.0
b     22.0
c     33.0
d     44.0
e     55.0
dtype  float64
```

Accessing the data from series



```python
import pandas as pd
s1 = pd.Series([11,22,33,44,55])
s1
```

```
0     11
1     22
2     33
3     44
4     55
dtype: int64
```

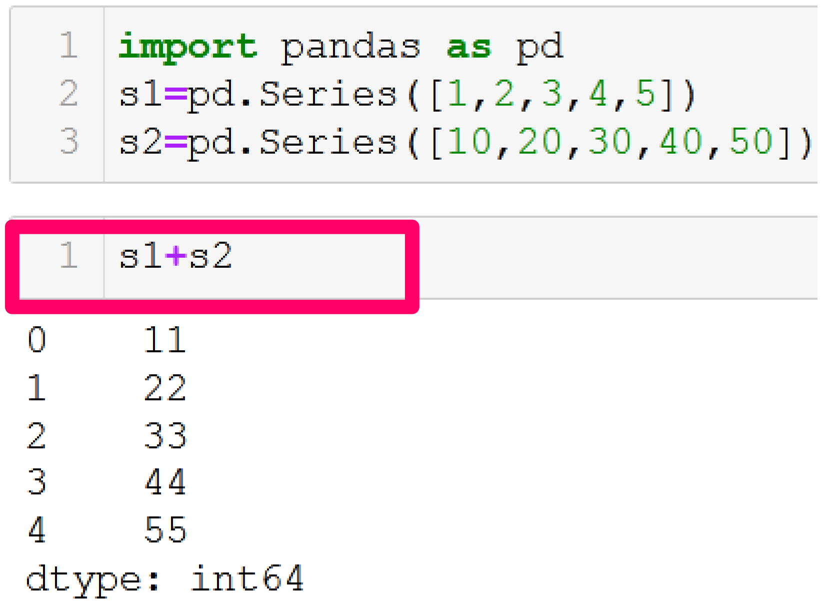
```python
s1[0]
```

```
11
```

```python
s1[0:2]
```

```
0     11
1     22
dtype: int64
```

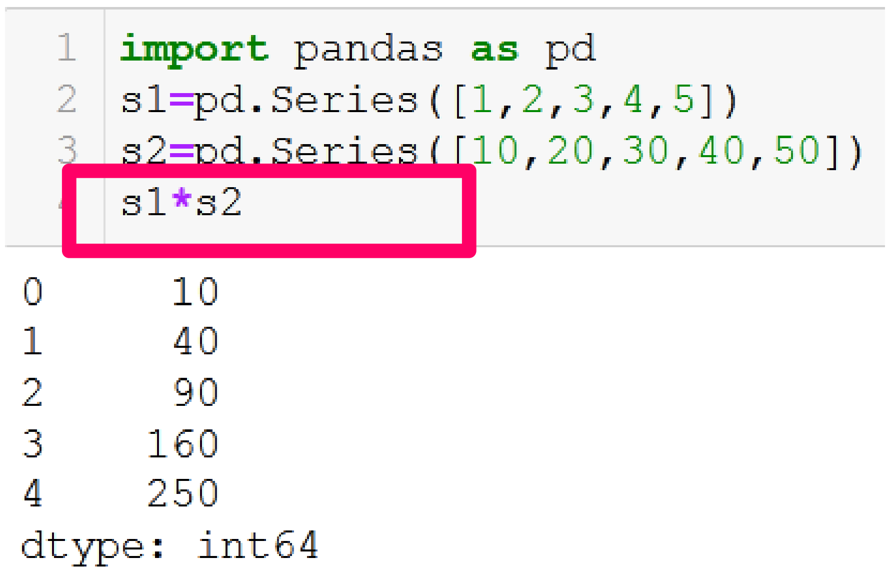
Operations on Pandas Series



```python
import pandas as pd
s1=pd.Series([1,2,3,4,5])
s2=pd.Series([10,20,30,40,50])
```

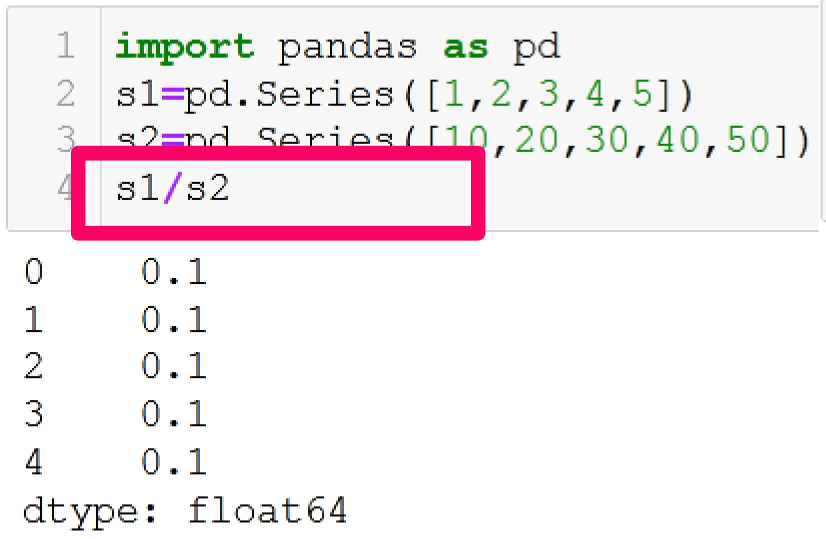
```python
s1+s2
```

```
0     11
1     22
2     33
3     44
4     55
dtype: int64
```



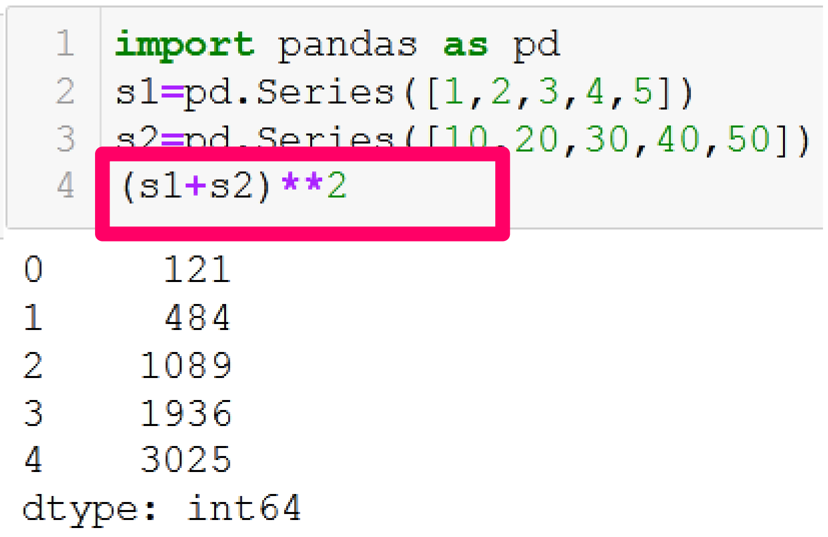
```python
import pandas as pd
s1=pd.Series([1,2,3,4,5])
s2=pd.Series([10,20,30,40,50])
s1*s2
```

```
0      10
1      40
2      90
3     160
4     250
dtype: int64
```



```python
import pandas as pd
s1=pd.Series([1,2,3,4,5])
s2=pd.Series([10,20,30,40,50])
s1/s2
```

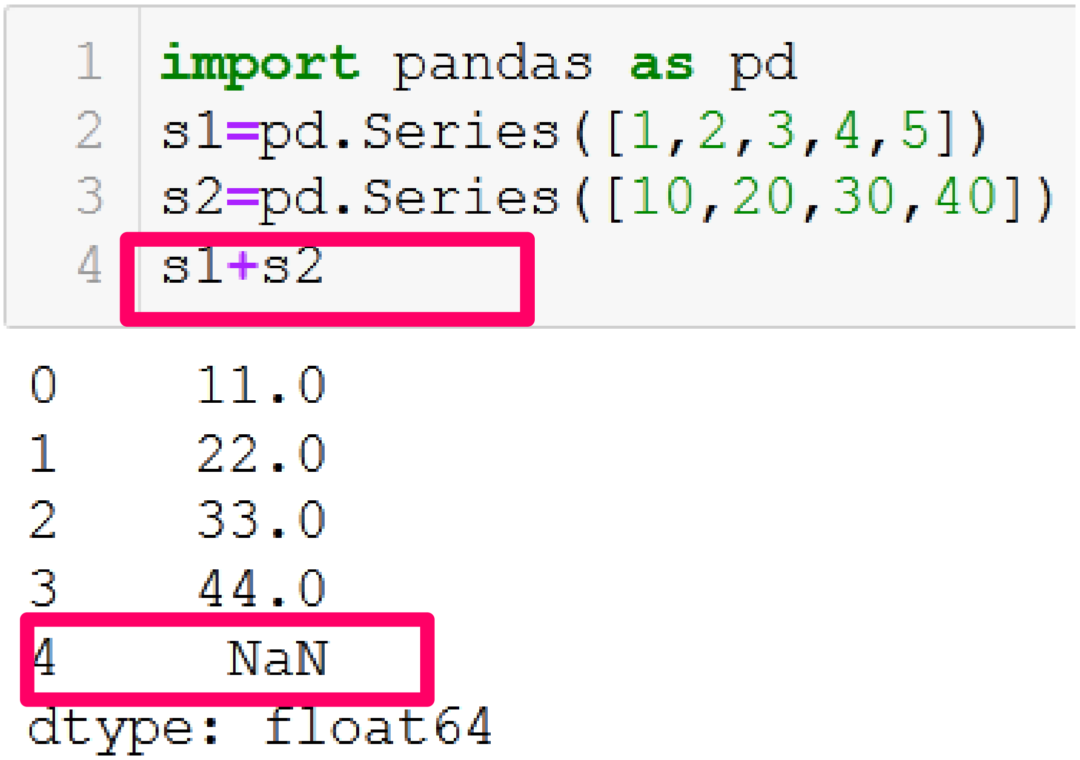
```
0     0.1
1     0.1
2     0.1
3     0.1
4     0.1
dtype: float64
```



```python
import pandas as pd
s1=pd.Series([1,2,3,4,5])
s2=pd.Series([10,20,30,40,50])
(s1+s2)**2
```

```
0      121
1      484
2     1089
3     1936
4     3025
dtype: int64
```

Operations on Pandas Series



```python
import pandas as pd
s1=pd.Series([1,2,3,4,5])
s2=pd.Series([10,20,30,40])
s1+s2
```

```
0     11.0
1     22.0
2     33.0
3     44.0
4      NaN
dtype: float64
```

NaN : Not a Number

Pandas (Data Frame)

- **Data Frame :** 2D data structure , Holds a data into table-like format.
- Data frames consist of the row, column and data

Name	ID Number	Marks
Priyang	123	98
Aadhya	124	99
Parshv	125	97
Vedant	126	99
Krisha	127	88

Pandas (Data Frame)

Name
Priyang
Aadhya
Parshv
Vedant
Krisha

ID Number
123
124
125
126
127

Marks
98
99
97
99
88

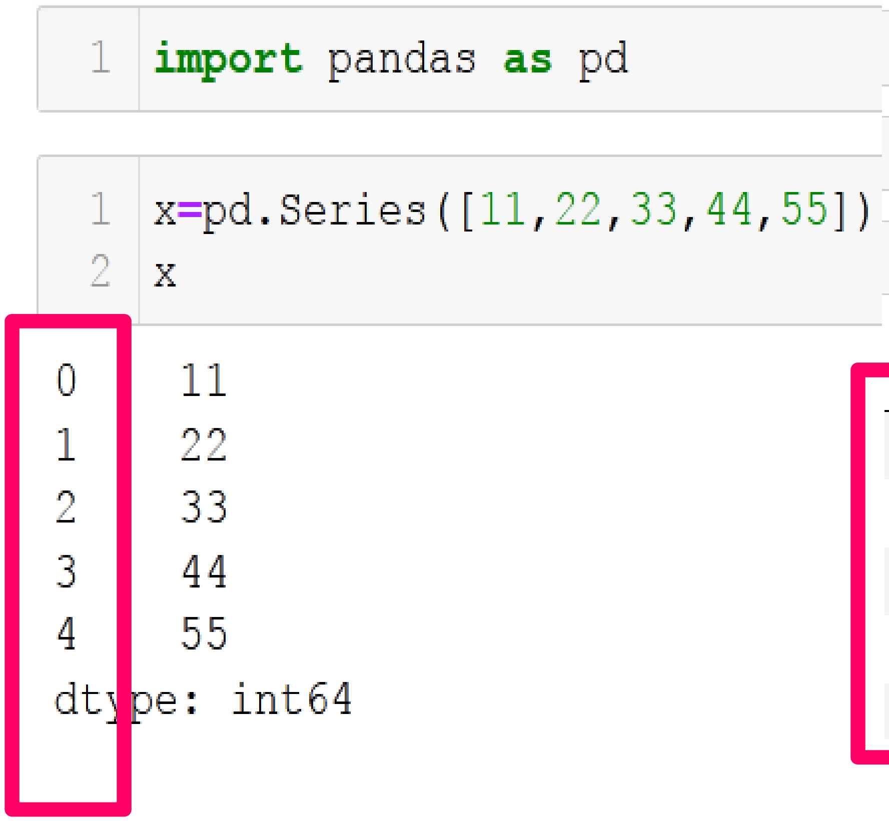
Name	ID Number	Marks
Priyang	123	98
Aadhya	124	99
Parshv	125	97
Vedant	126	99
Krisha	127	88

Pandas (Series) Vs. Pandas (Data Frame)

- Series : 1D labeled array
- Series can hold heterogeneous values (int, float, string etc.)
- Series == Column

- Data Frame : 2D data structure , Holds a data into table-like format.
- Data Frame can hold heterogeneous values (int, float, string etc.)
- Data frames consist of the row, column and data

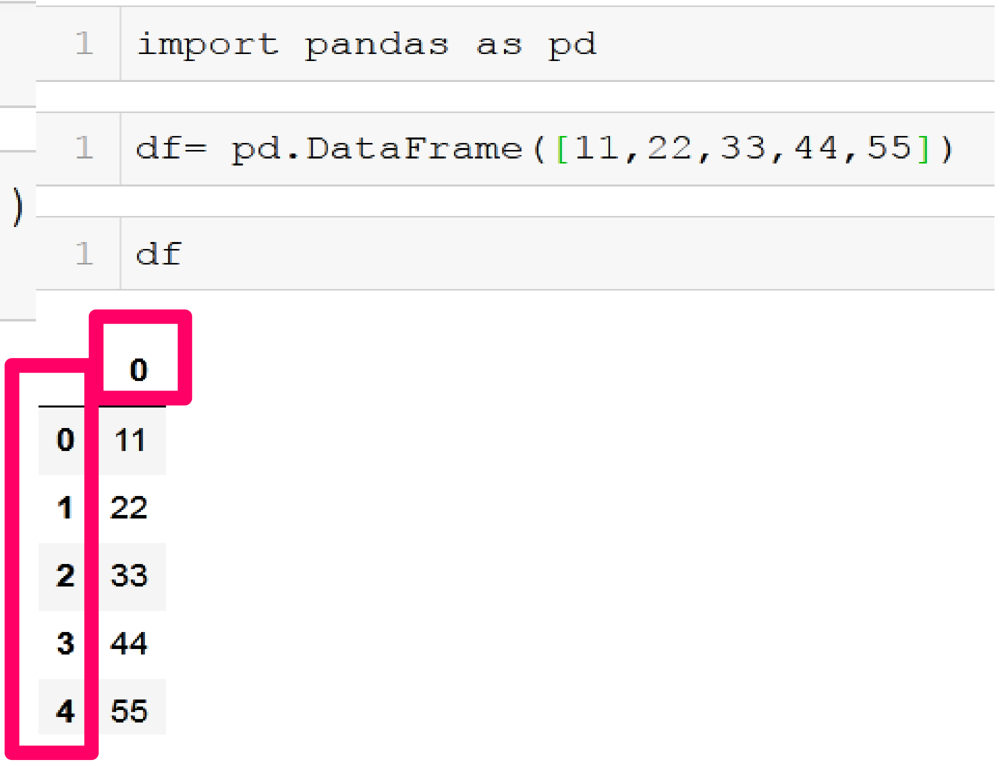
Pandas (Series) Vs. Pandas (Data Frame)



```python
import pandas as pd
```

```python
x=pd.Series([11,22,33,44,55])
x
```

```
0    11
1    22
2    33
3    44
4    55
dtype: int64
```



```python
import pandas as pd
```

```python
df= pd.DataFrame([11,22,33,44,55])
```

```python
df
```

```
    0
0  11
1  22
2  33
3  44
4  55
```

Pandas Data Frame can be created using

List

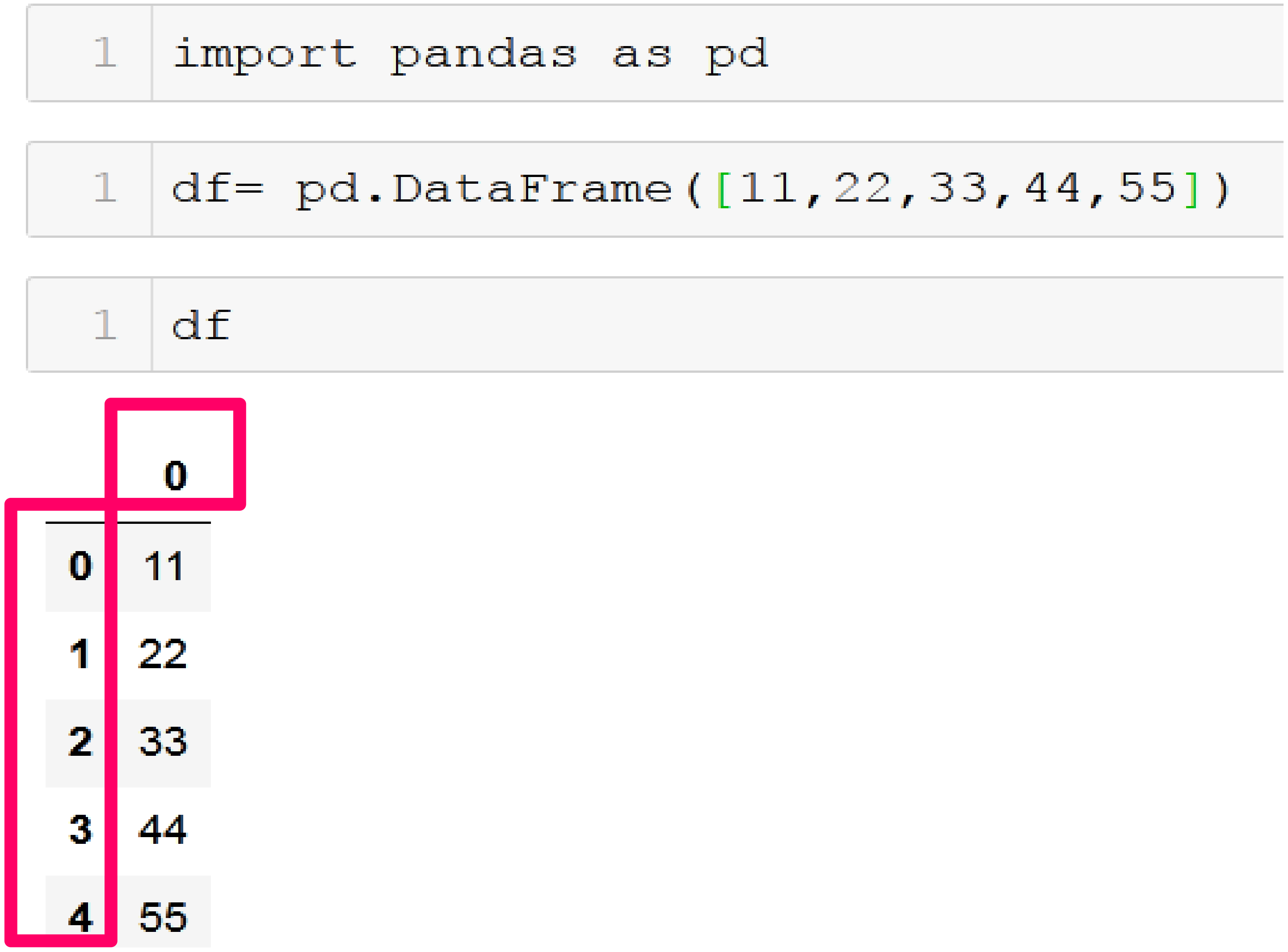
Nested List

Dictionary

Numpy array

Series

Create a Data Frame from List



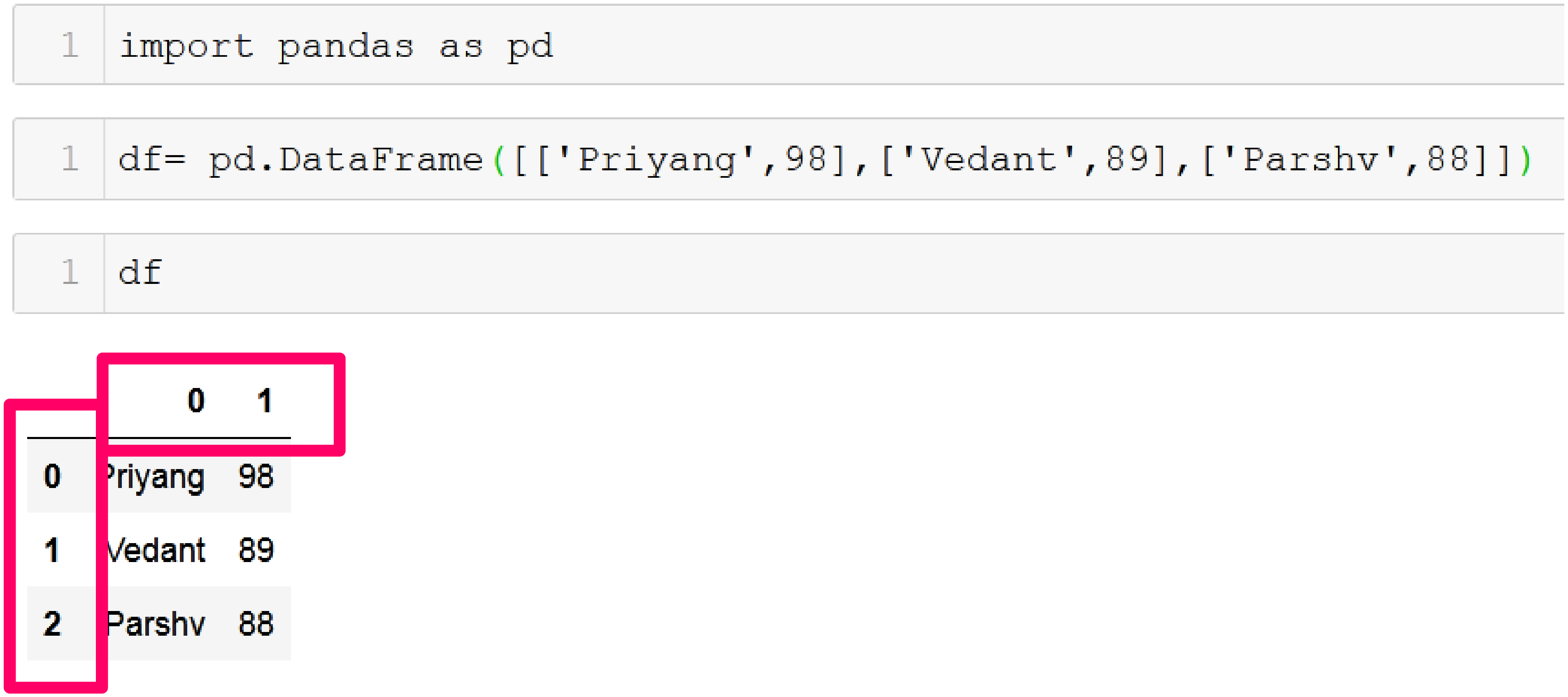
```python
import pandas as pd
```

```python
df= pd.DataFrame([11,22,33,44,55])
```

```python
df
```

```
    0
0  11
1  22
2  33
3  44
4  55
```

Create a Data Frame from Nested List



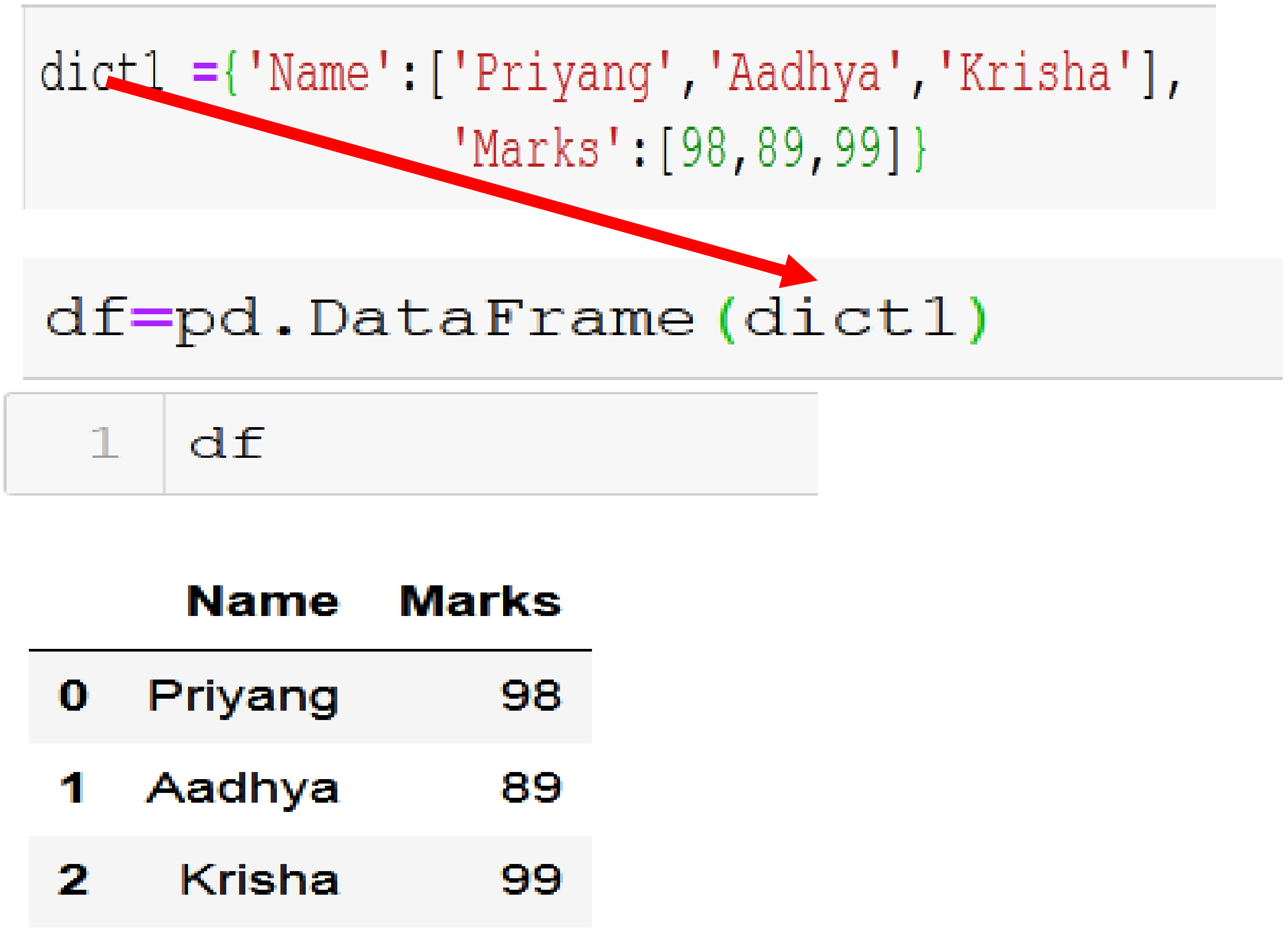
```
import pandas as pd
```

```
df= pd.DataFrame([['Priyang',98],['Vedant',89],['Parshv',88]])
```

```
df
```

```
         0   1
0  Priyang  98
1   Vedant  89
2   Parshv  88
```

Create a Data Frame from Dictionary



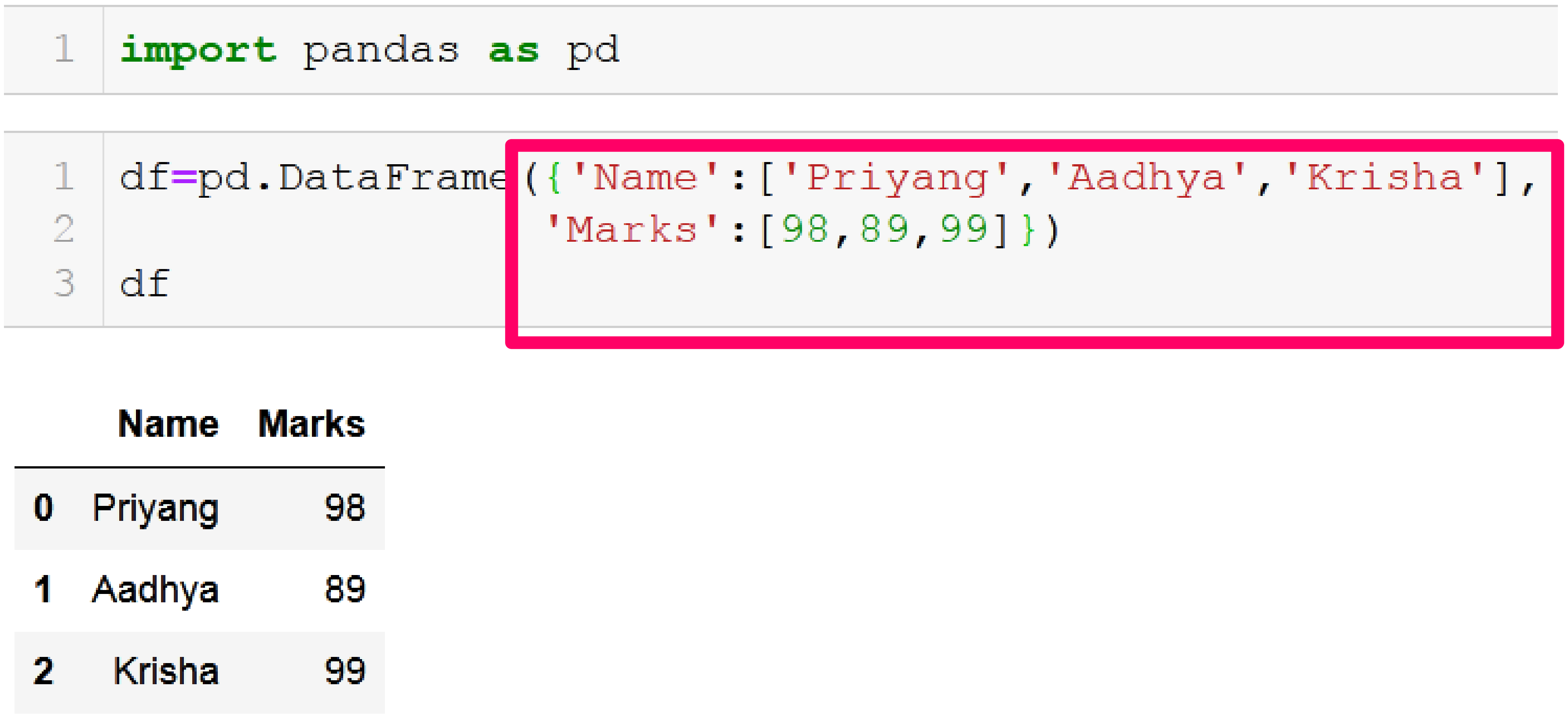
```python
dict1 ={'Name':['Priyang','Aadhya','Krisha'],
                 'Marks':[98,89,99]}
```

```python
df=pd.DataFrame(dict1)
```

```python
df
```

```
      Name  Marks
0  Priyang     98
1   Aadhya     89
2   Krisha     99
```

Create a Data Frame from Dictionary

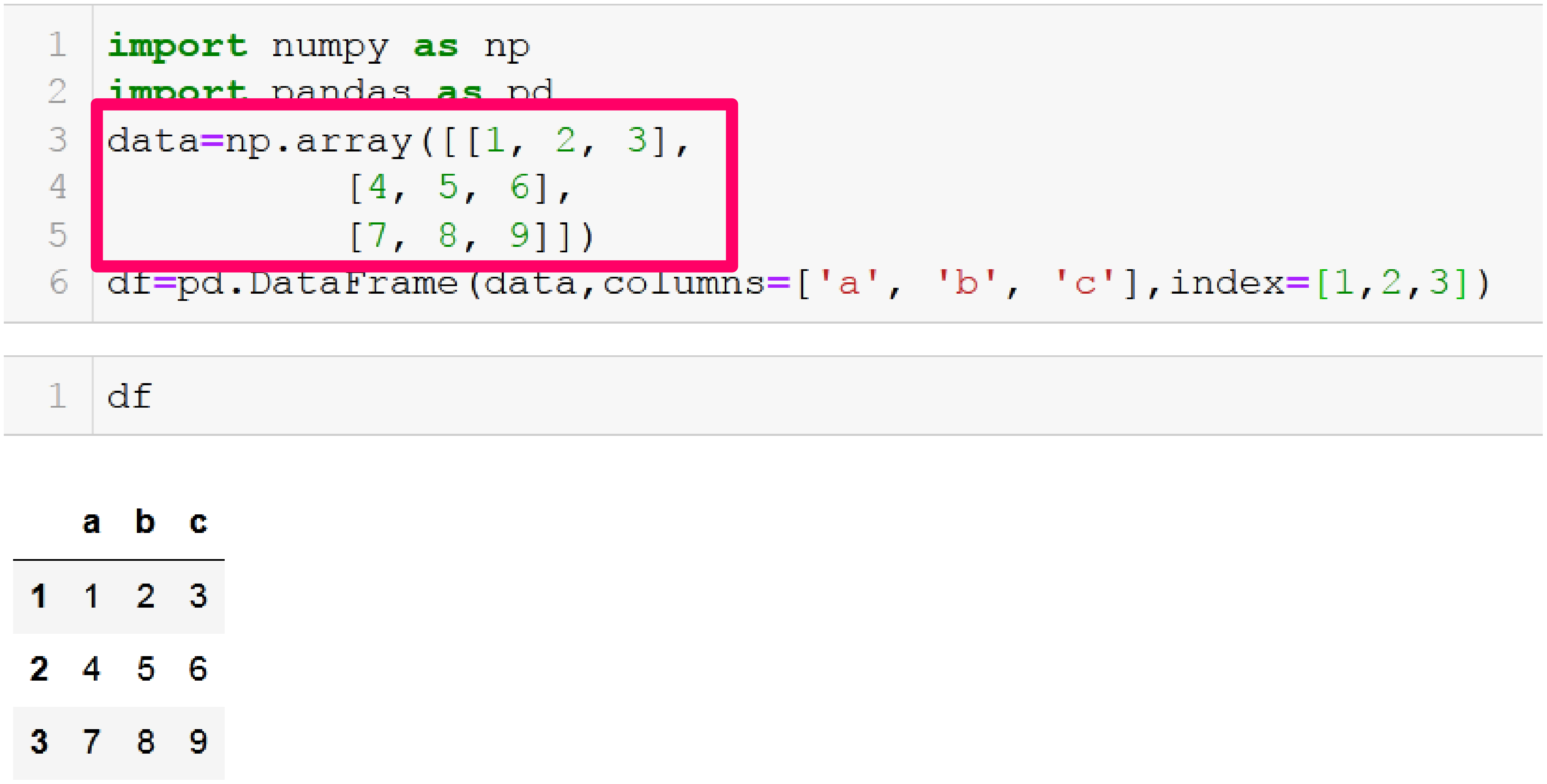


```
import pandas as pd
```

```
df=pd.DataFrame({'Name':['Priyang','Aadhya','Krisha'],
'Marks':[98,89,99]})
df
```

	Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99

Create a Data Frame using Numpy



```
import numpy as np
import pandas as pd
data=np.array([[1, 2, 3],
              [4, 5, 6],
              [7, 8, 9]])
df=pd.DataFrame(data,columns=['a', 'b', 'c'],index=[1,2,3])
```

```
df
```

```
   a  b  c
1  1  2  3
2  4  5  6
3  7  8  9
```

Pandas (Data Frame)

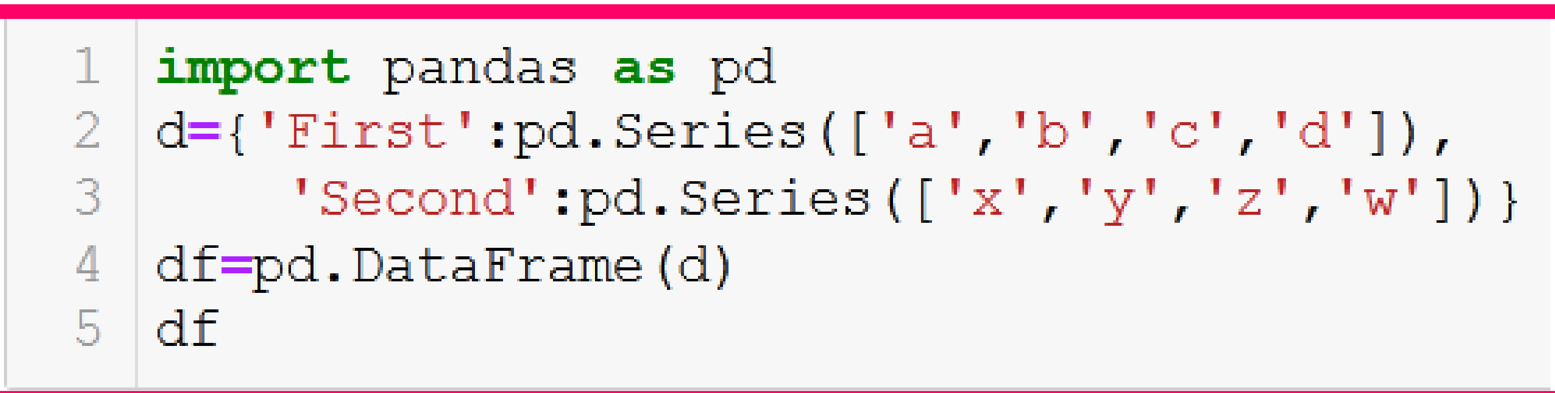
Name
Priyang
Aadhya
Parshv
Vedant
Krisha

ID Number
123
124
125
126
127

Marks
98
99
97
99
88

Name	ID Number	Marks
Priyang	123	98
Aadhya	124	99
Parshv	125	97
Vedant	126	99
Krisha	127	88

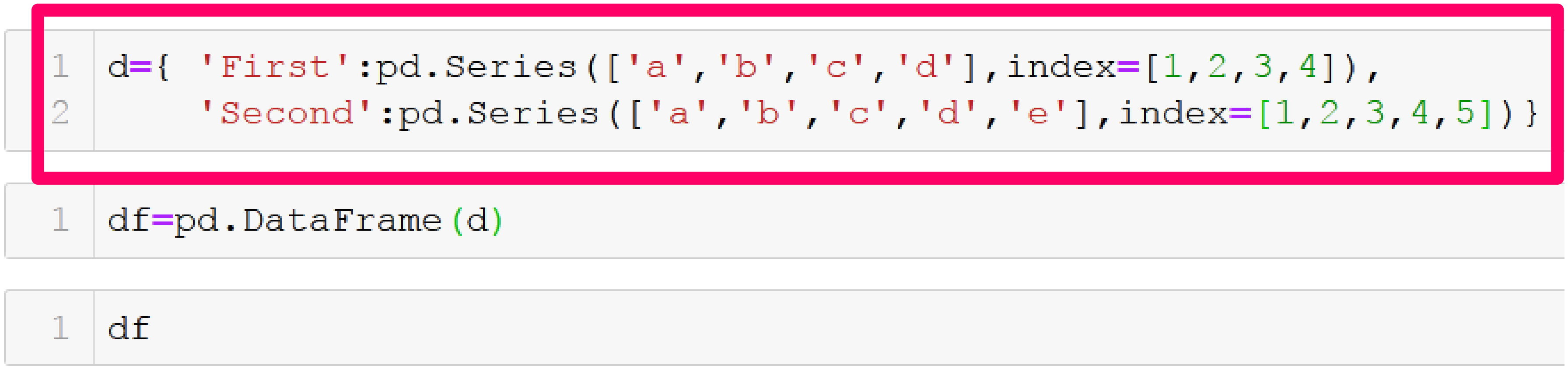
Create a Data Frame using Series



```python
import pandas as pd
d={'First':pd.Series(['a','b','c','d']),
    'Second':pd.Series(['x','y','z','w'])}
df=pd.DataFrame(d)
df
```

	First	Second
0	a	x
1	b	y
2	c	z
3	d	w

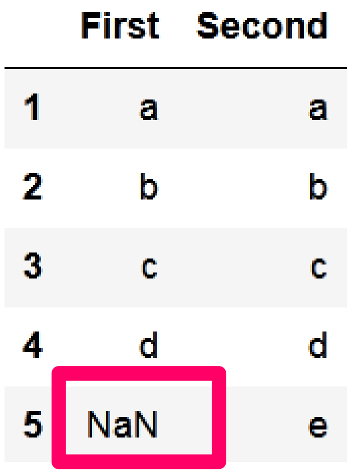
Create a Data Frame using Series



```python
d={ 'First':pd.Series(['a','b','c','d'],index=[1,2,3,4]),
    'Second':pd.Series(['a','b','c','d','e'],index=[1,2,3,4,5])}
```

```python
df=pd.DataFrame(d)
```

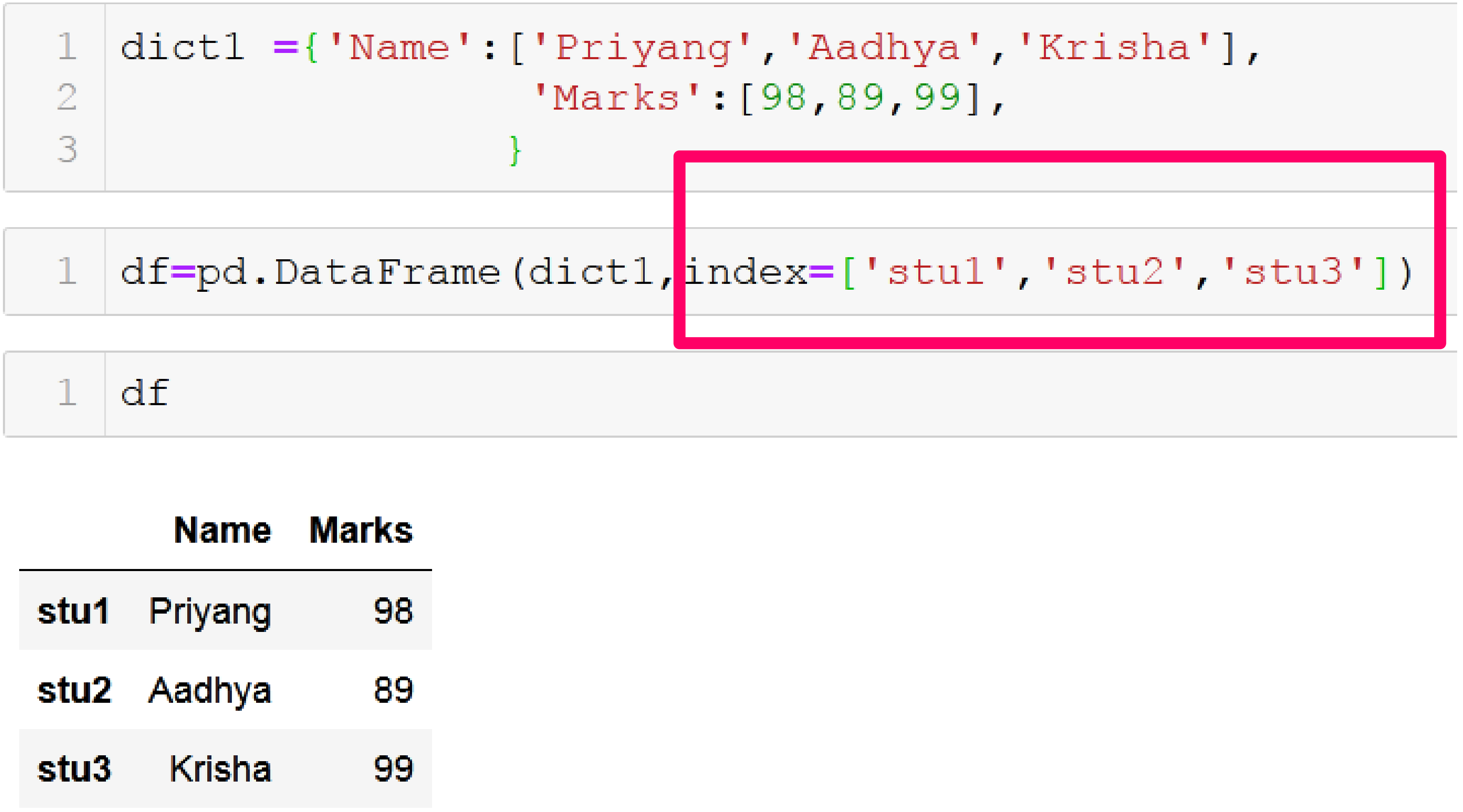
```python
df
```



	First	Second
1	a	a
2	b	b
3	c	c
4	d	d
5	NaN	e

NaN : Not a Number

Create a Data Frame from Dictionary (with Index)



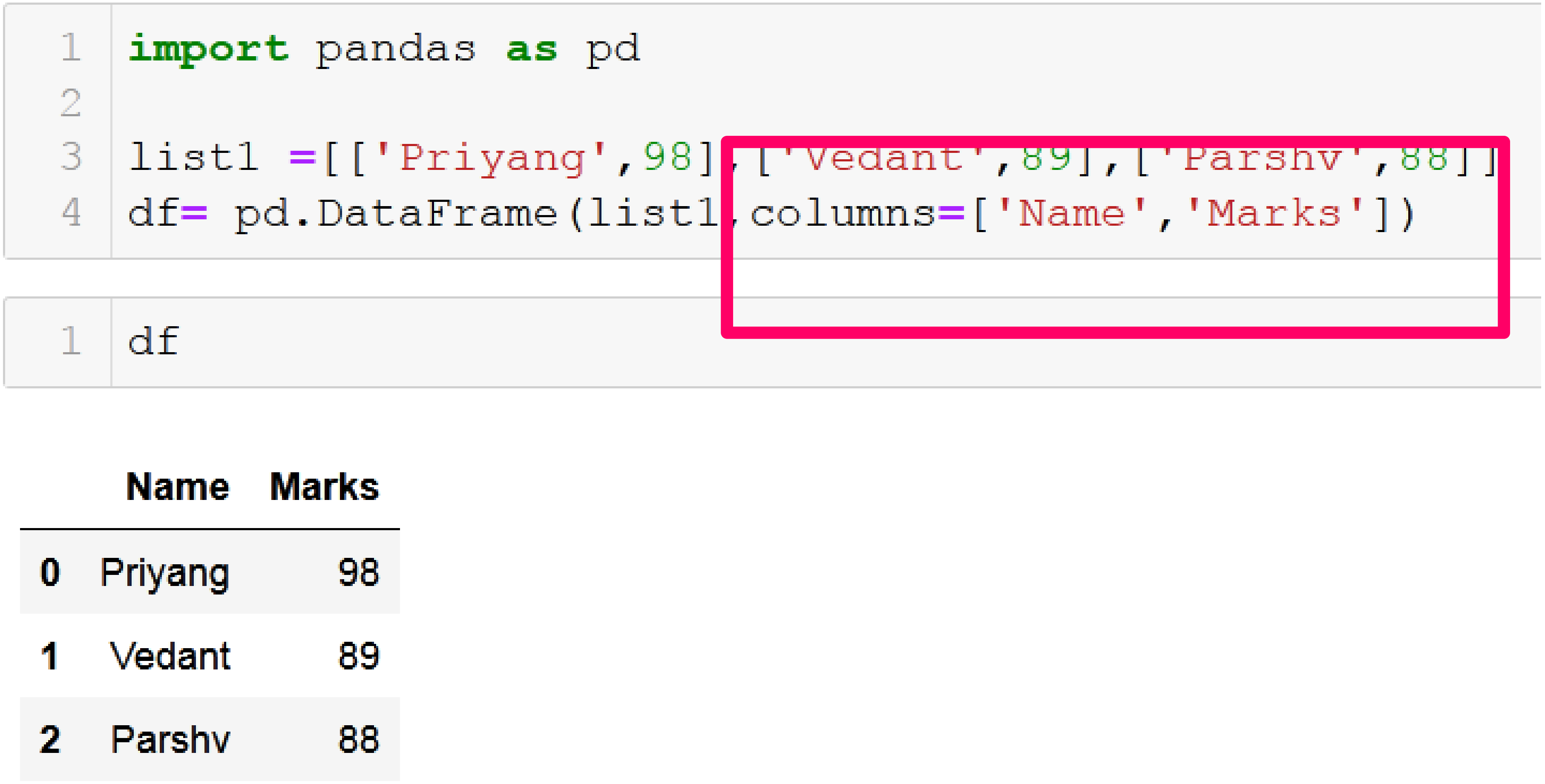
```python
dict1 ={'Name':['Priyang','Aadhya','Krisha'],
                  'Marks':[98,89,99],
                }
```

```python
df=pd.DataFrame(dict1,index=['stu1','stu2','stu3'])
```

```python
df
```

	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99

Create a Data Frame from Dictionary (with Columns)

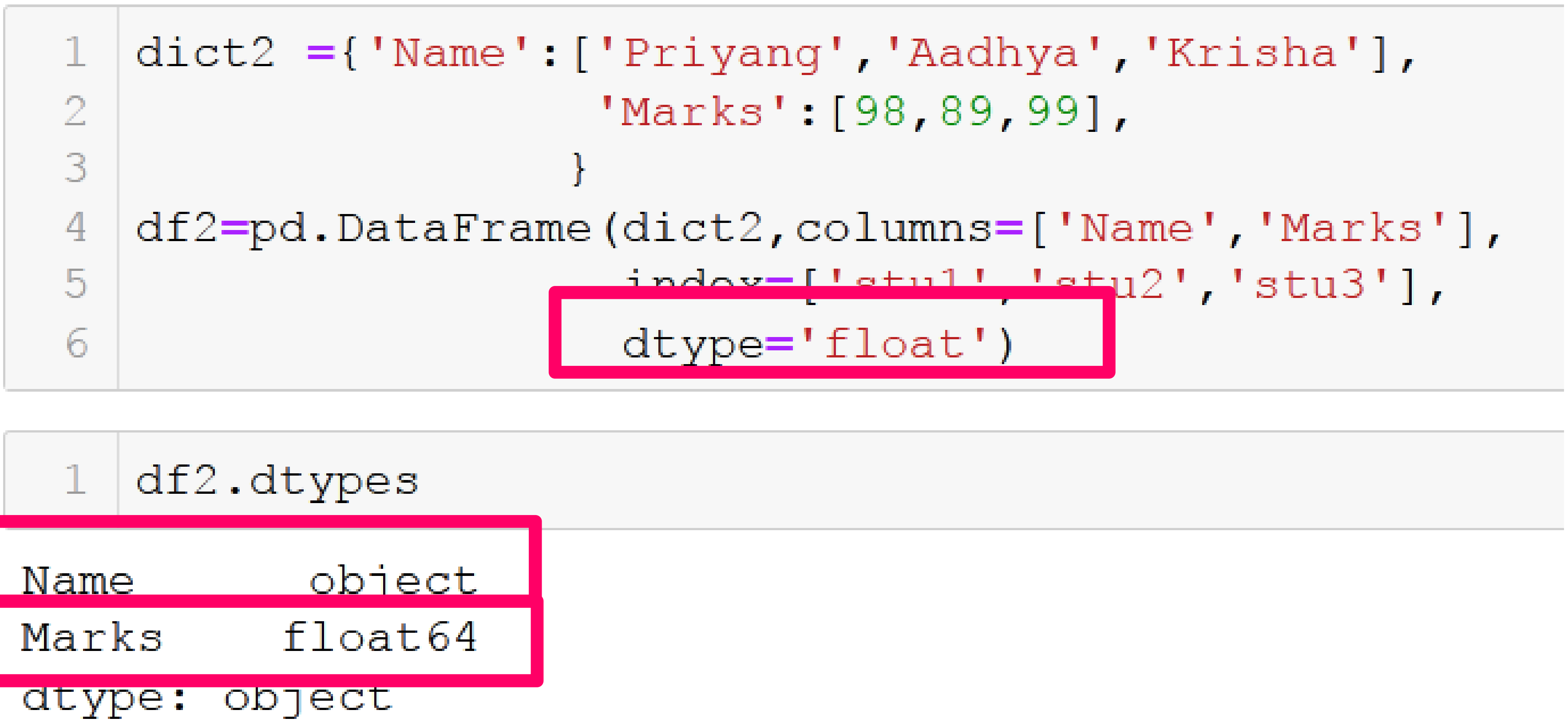


```python
import pandas as pd

list1 =[['Priyang',98],['Vedant',89],['Parshv',88]]
df= pd.DataFrame(list1,columns=['Name','Marks'])
```

```python
df
```

```
      Name  Marks
0  Priyang     98
1   Vedant     89
2   Parshv     88
```

Create a Data Frame from Dictionary (with dtype)

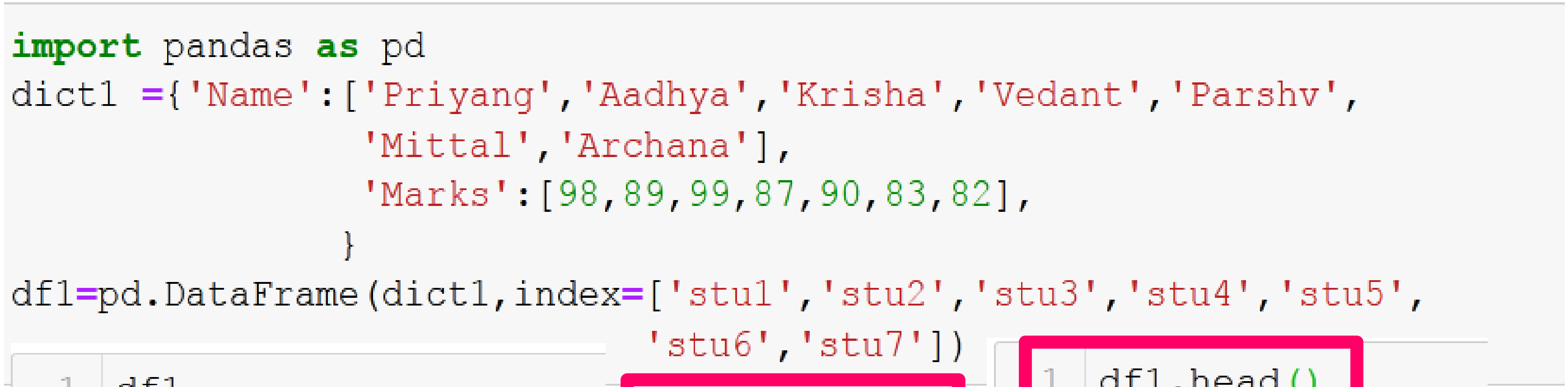


```python
dict2 ={'Name':['Priyang','Aadhya','Krisha'],
                'Marks':[98,89,99],
               }
df2=pd.DataFrame(dict2,columns=['Name','Marks'],
                 index=['stu1','stu2','stu3'],
                 dtype='float')
```

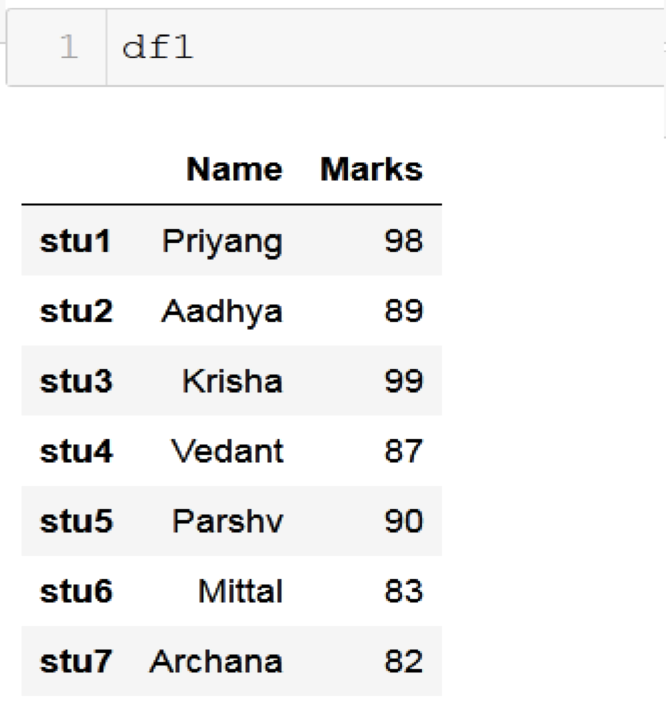
```python
df2.dtypes
```

```
Name      object
Marks    float64
dtype: object
```

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)

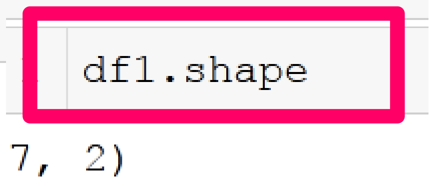


```
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                               'stu6','stu7'])
```



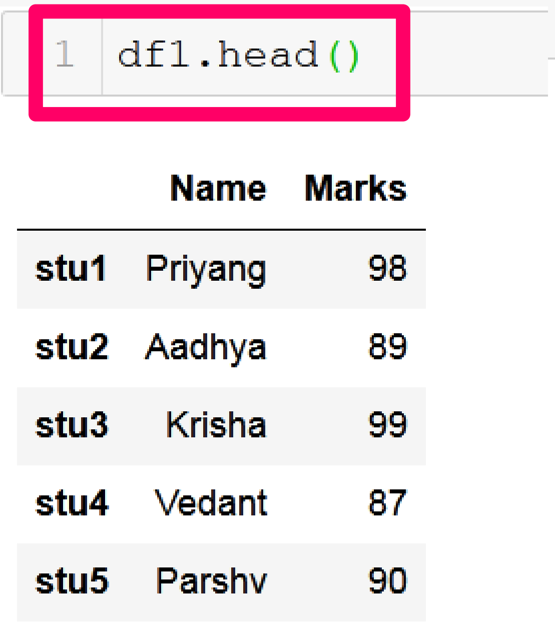
```
df1
```

	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82



```
df1.shape
```

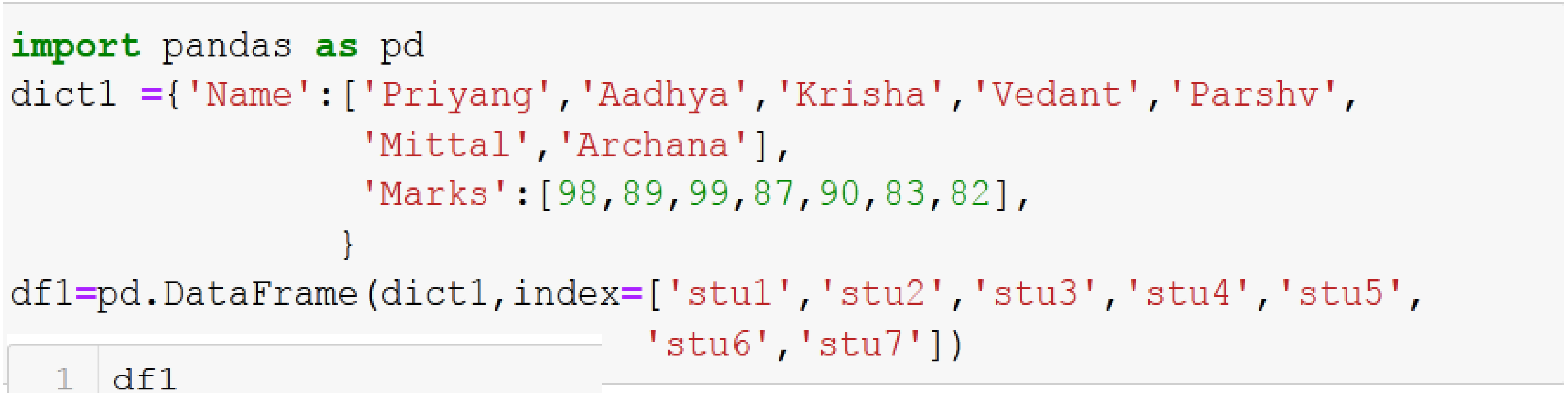
```
7, 2)
```



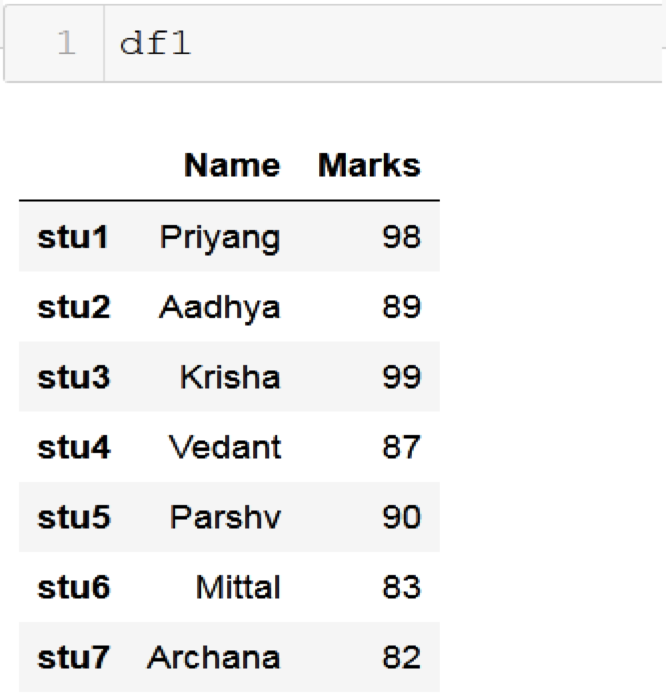
```
df1.head()
```

	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)

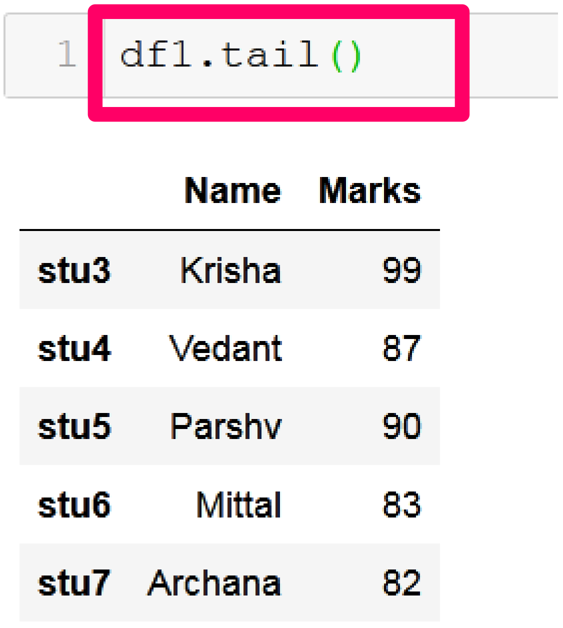


```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                              'stu6','stu7'])
```



```python
df1
```

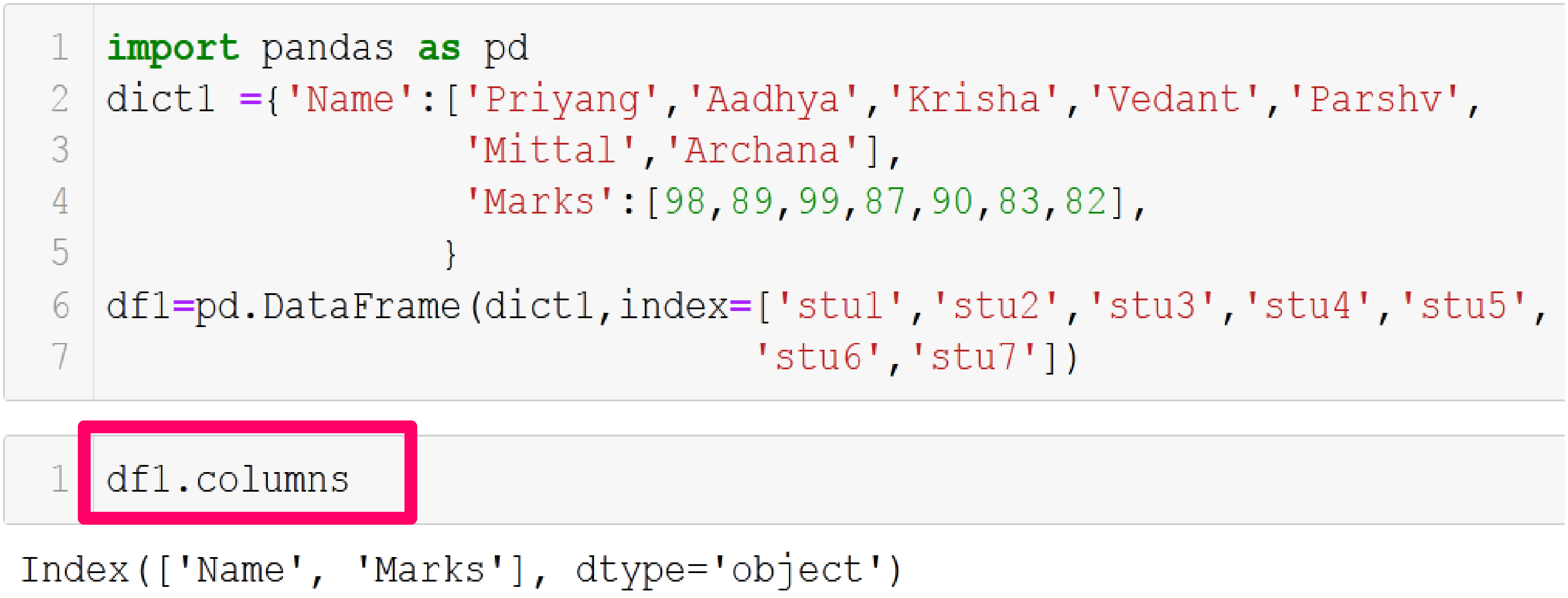
	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82



```python
df1.tail()
```

	Name	Marks
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)

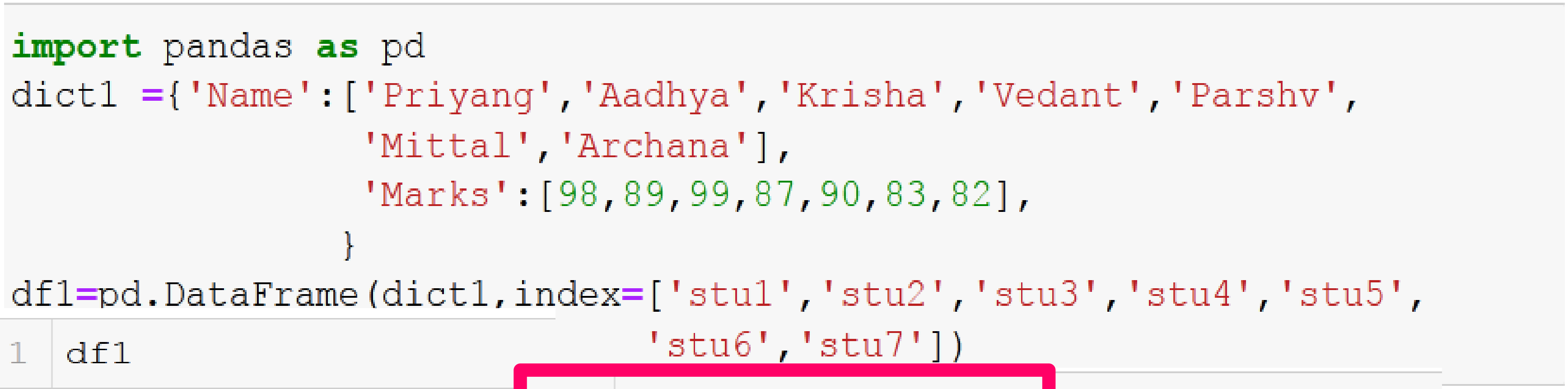


```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                             'stu6','stu7'])
```

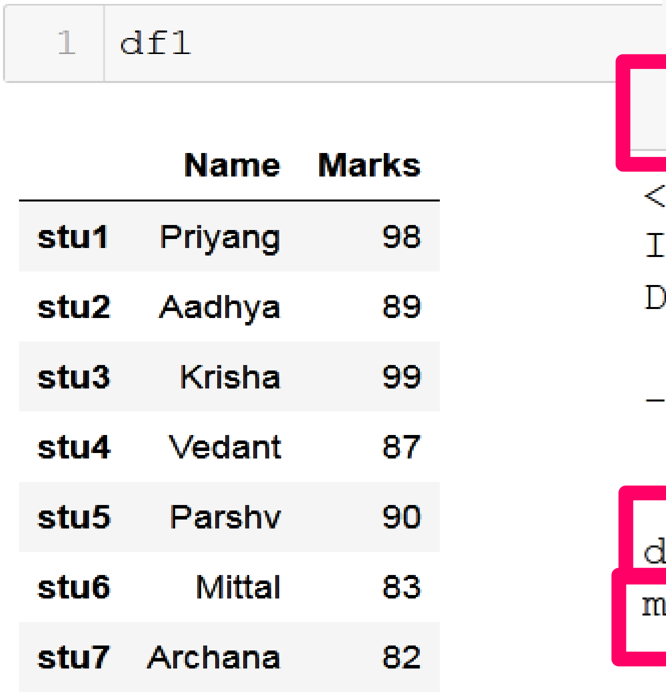
```python
df1.columns
```

```
Index(['Name', 'Marks'], dtype='object')
```

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)

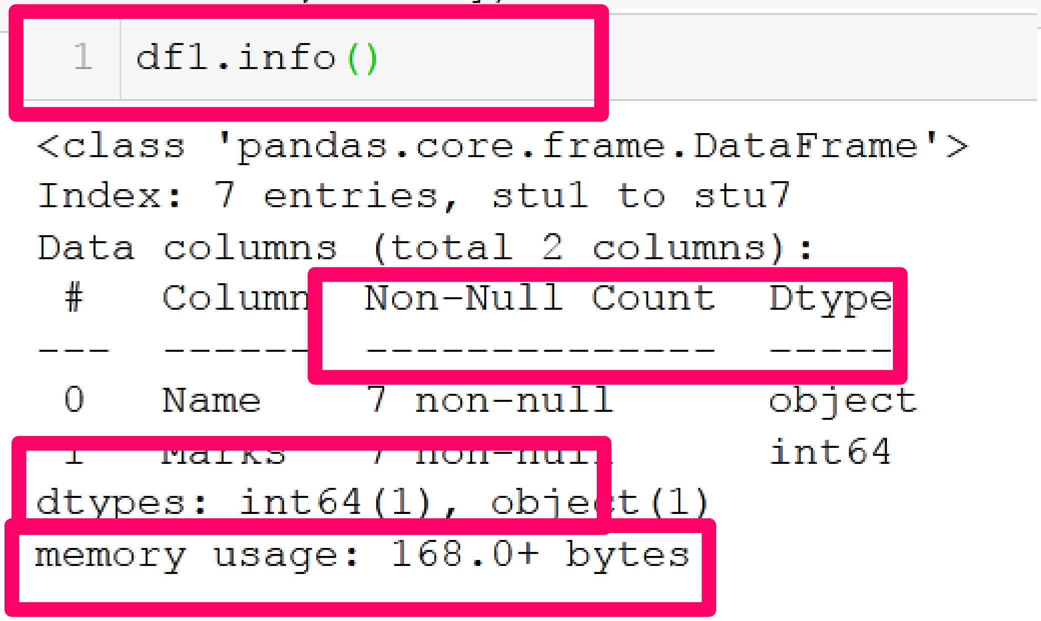


```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                              'stu6','stu7'])
```



```python
df1
```

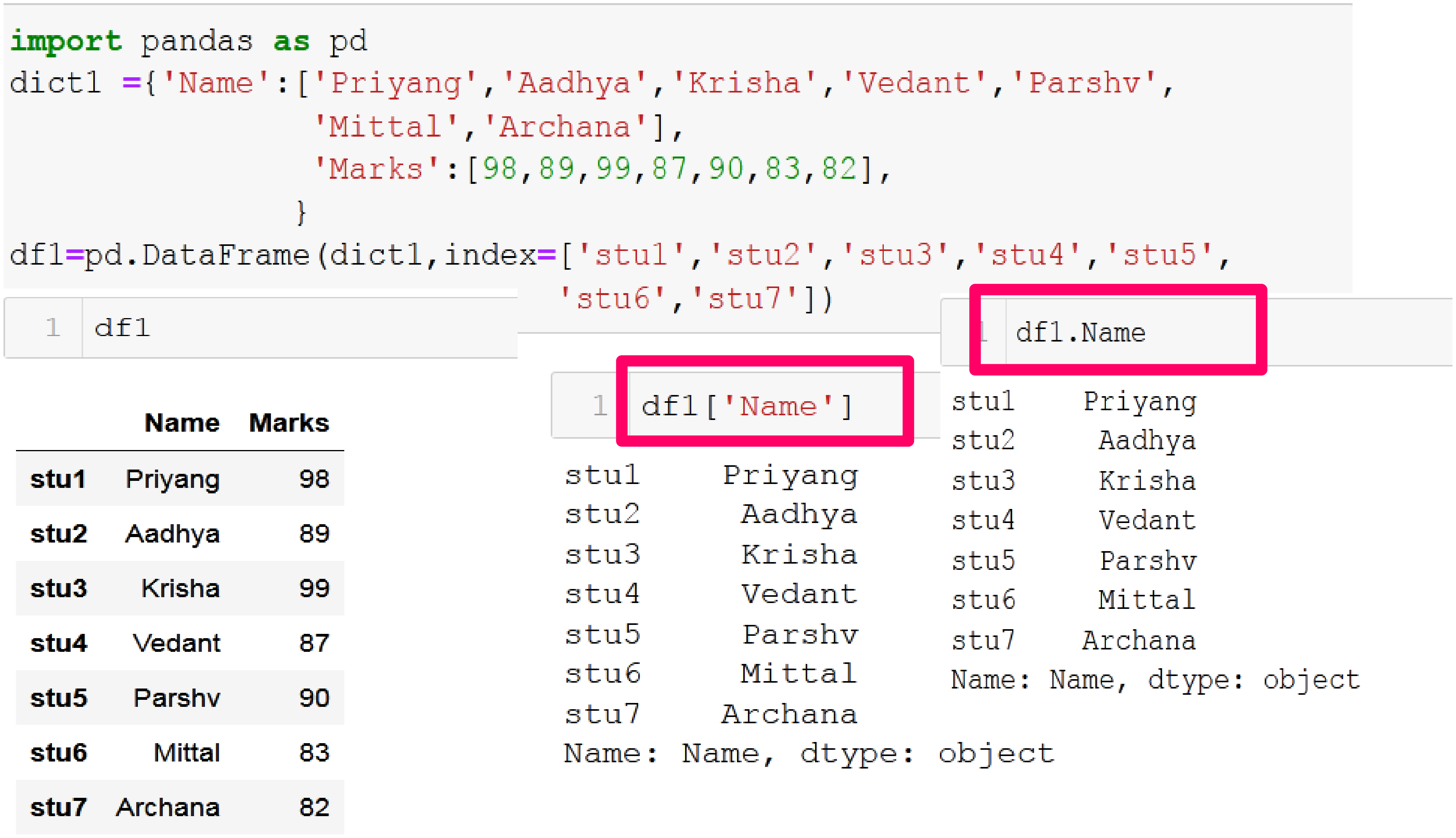
	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82



```python
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 7 entries, stu1 to stu7
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Name    7 non-null      object
 1   Marks   7 non-null      int64 
dtypes: int64(1), object(1)
memory usage: 168.0+ bytes
```

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)



```python
import pandas as pd
dictl ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dictl,index=['stu1','stu2','stu3','stu4','stu5',
                              'stu6','stu7'])
```

```python
df1
```

	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82

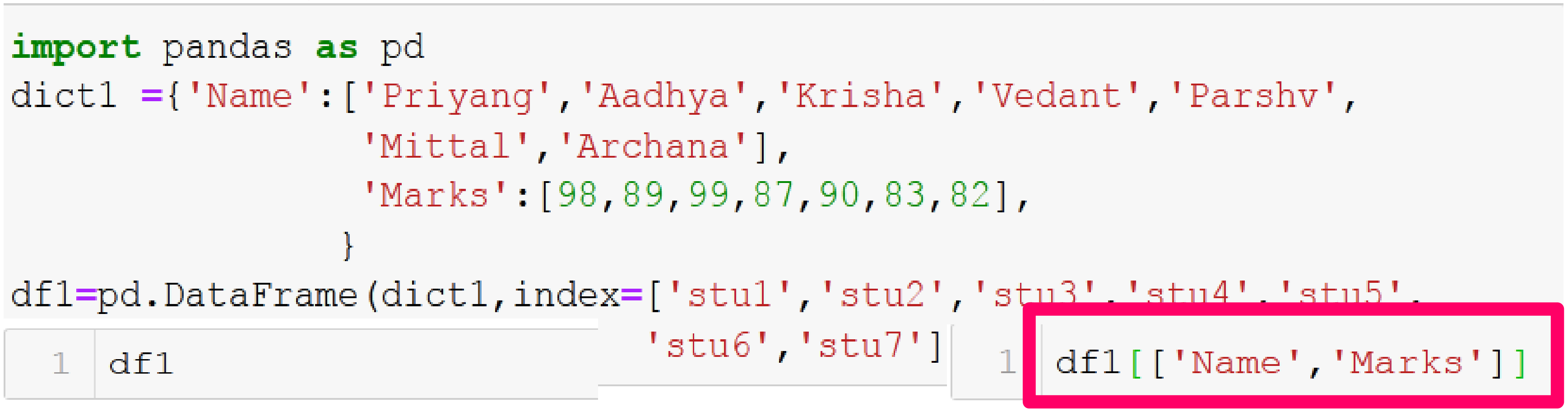
```python
df1['Name']
```

```
stu1     Priyang
stu2      Aadhya
stu3      Krisha
stu4      Vedant
stu5      Parshv
stu6      Mittal
stu7     Archana
Name: Name, dtype: object
```

```python
df1.Name
```

```
stu1     Priyang
stu2      Aadhya
stu3      Krisha
stu4      Vedant
stu5      Parshv
stu6      Mittal
stu7     Archana
Name: Name, dtype: object
```

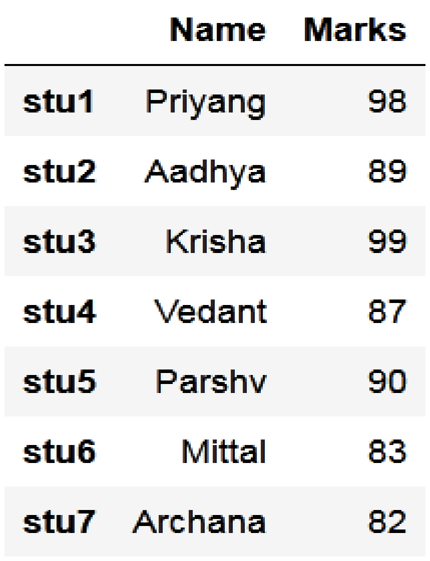
Viewing/Inspecting Data (Pandas Data Frame)(Accessing Columns)



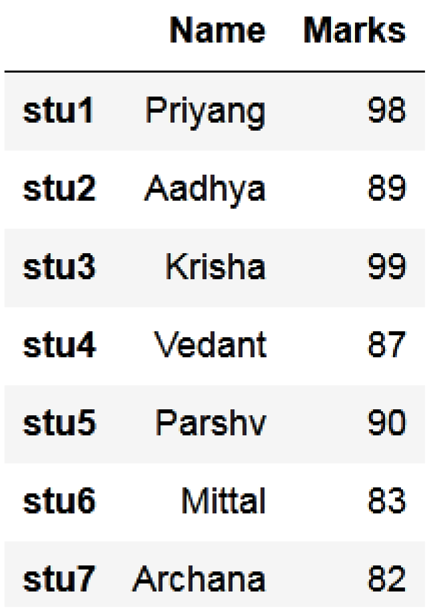
```
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                             'stu6','stu7'])
```

```
df1
```

```
df1[['Name','Marks']]
```

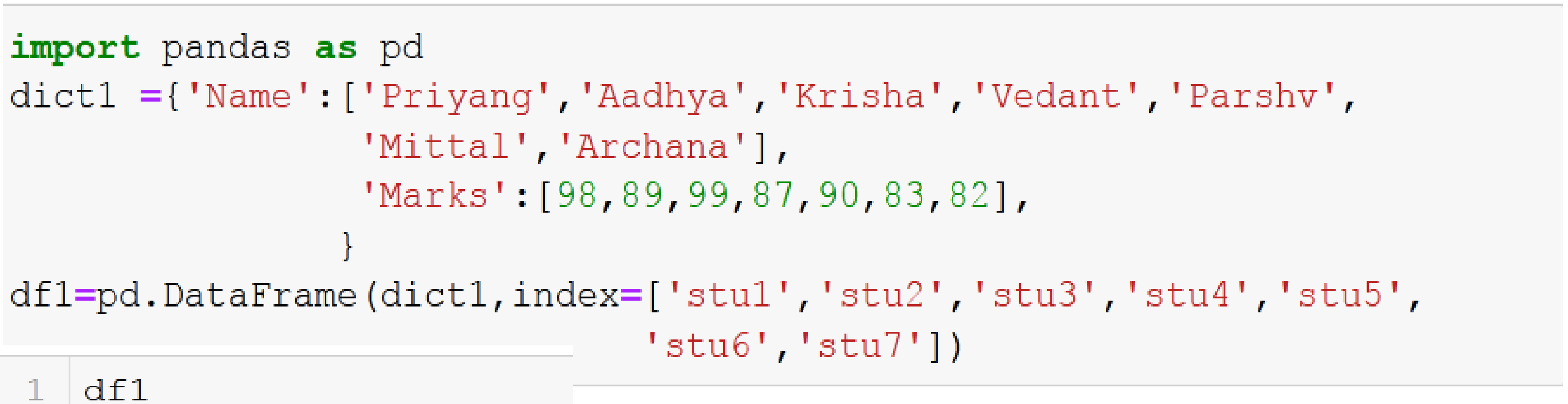


	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82

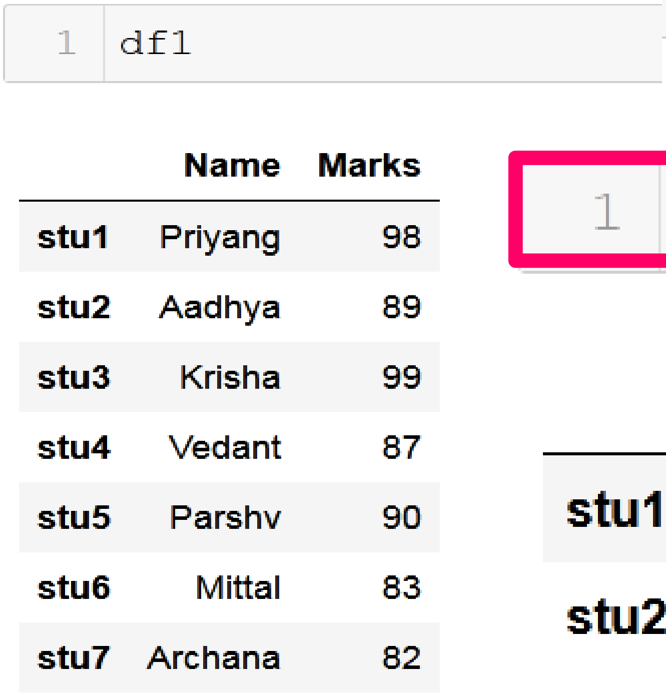


	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)(Accessing Rows)

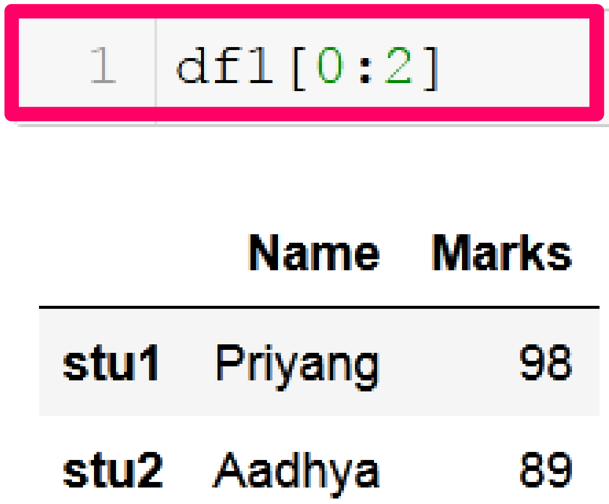


```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                              'stu6','stu7'])
```



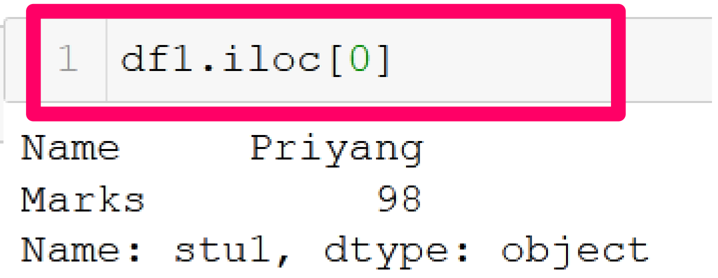
```python
df1
```

	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82



```python
df1[0:2]
```

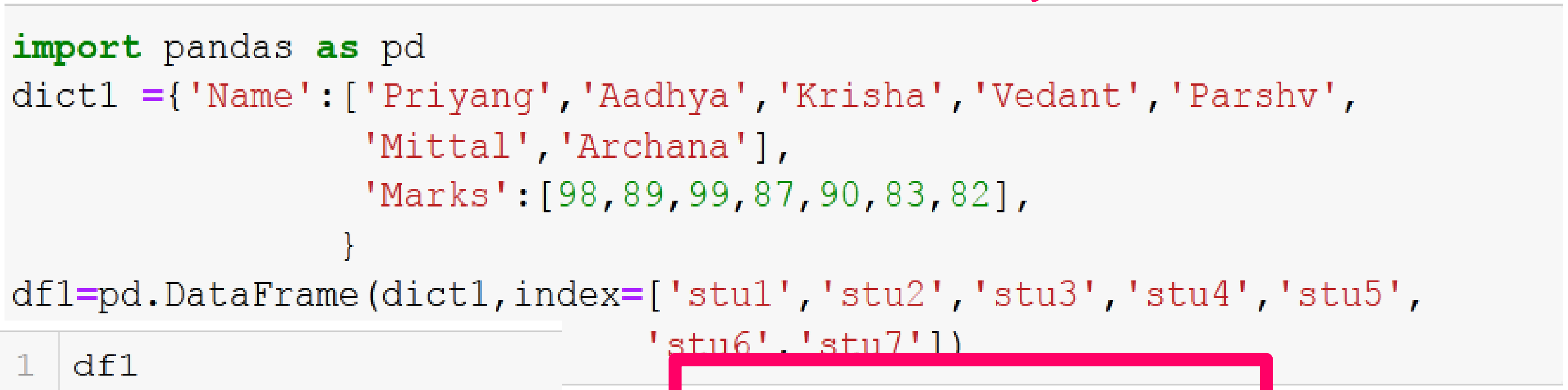
	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89



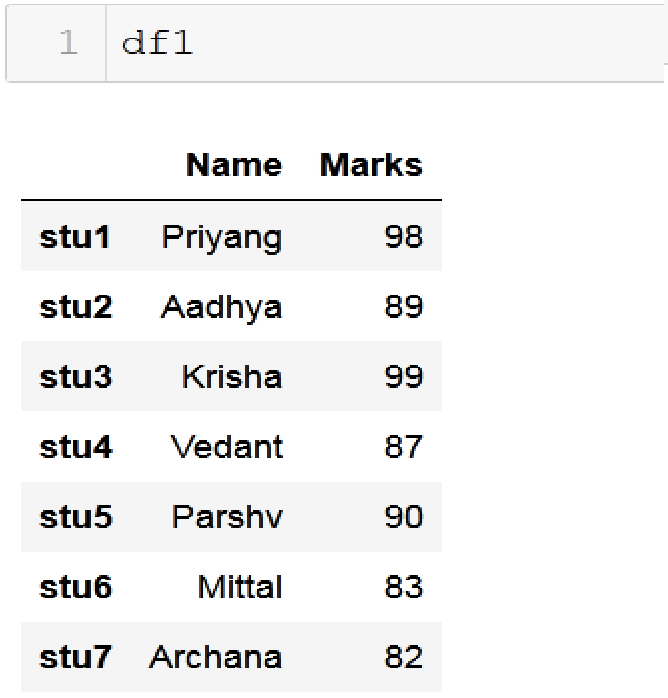
```python
df1.iloc[0]
```

```
Name       Priyang
Marks           98
Name: stu1, dtype: object
```

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)

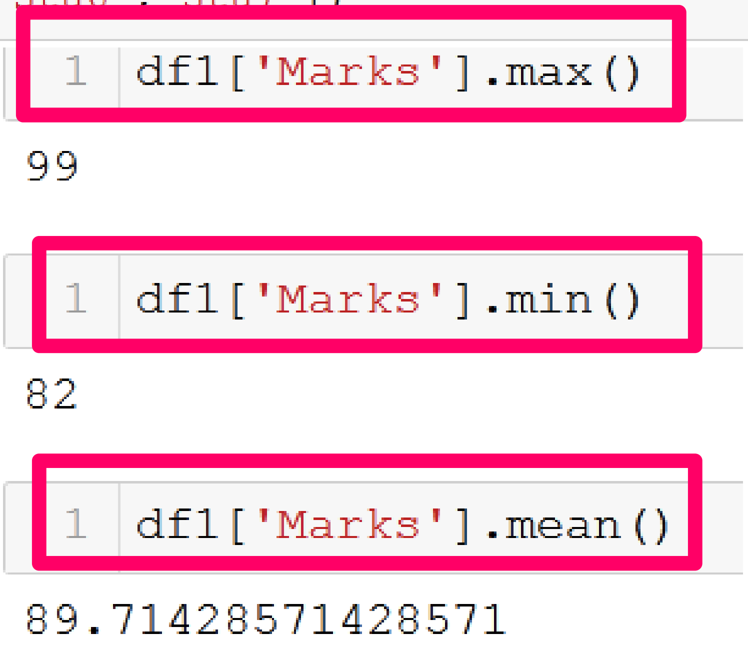


```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                              'stu6','stu7'])
```



```python
df1
```

	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82



```python
df1['Marks'].max()
```

```
99
```

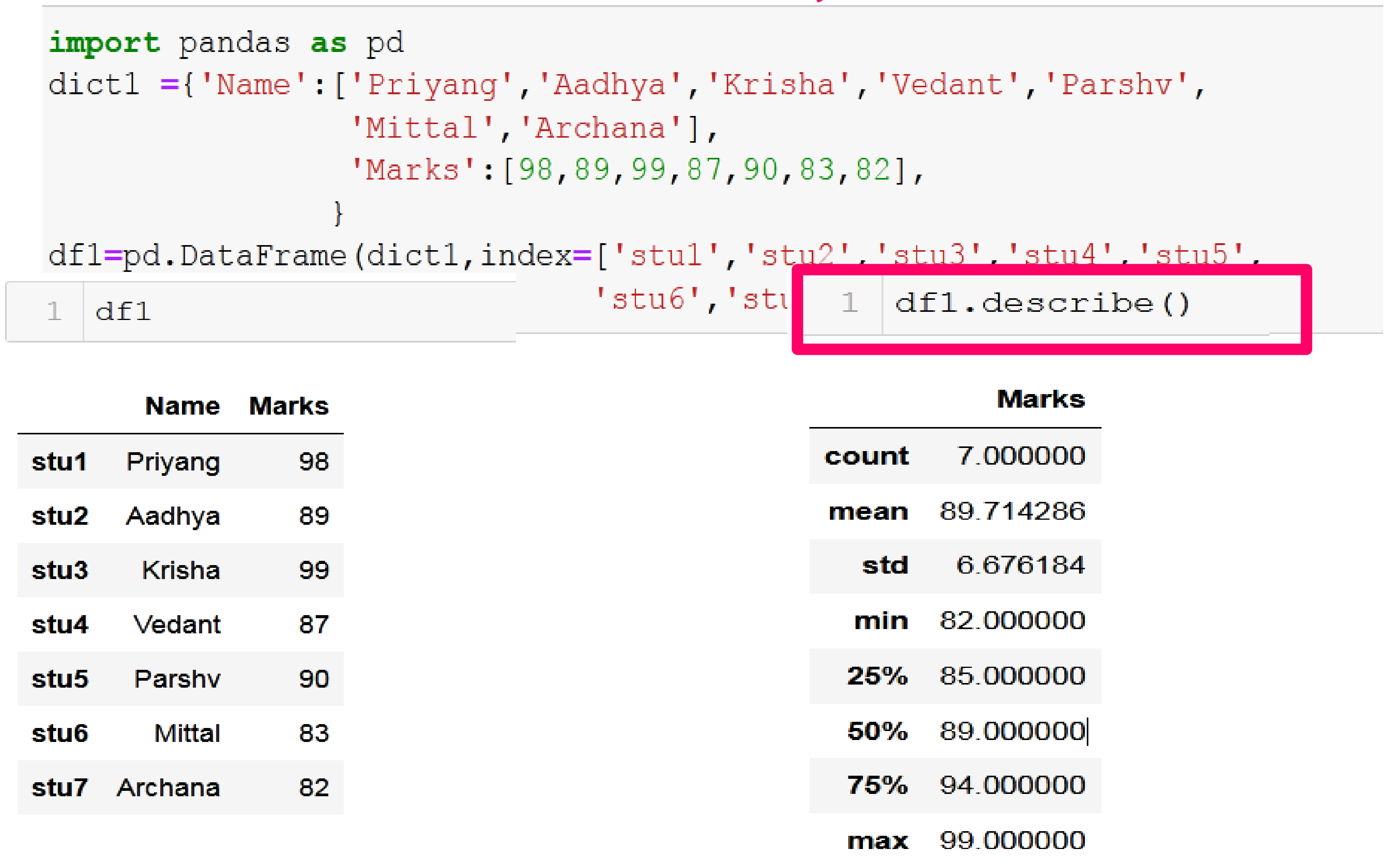
```python
df1['Marks'].min()
```

```
82
```

```python
df1['Marks'].mean()
```

```
89.71428571428571
```

Viewing/Inspecting Data (Pandas Data Frame)



```
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                              'stu6','stu
```

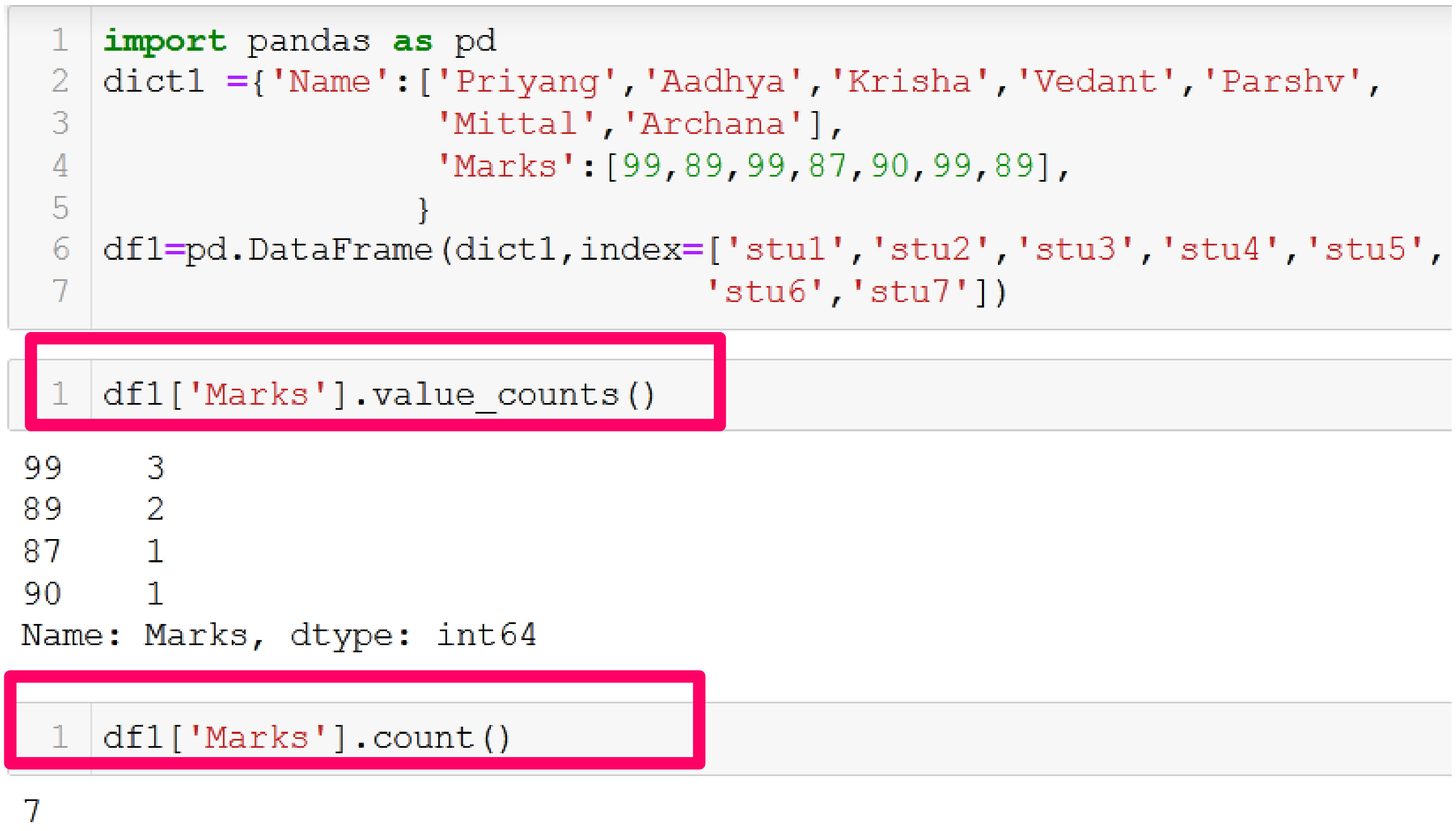
```
df1
```

```
         Name  Marks
stu1  Priyang     98
stu2   Aadhya     89
stu3   Krisha     99
stu4   Vedant     87
stu5   Parshv     90
stu6   Mittal     83
stu7  Archana     82
```

```
df1.describe()
```

```
           Marks
count   7.000000
mean   89.714286
std     6.676184
min    82.000000
25%    85.000000
50%    89.000000
75%    94.000000
max    99.000000
```

VIEWING/INSPECTING DATA (PANDAS DATA FRAME)



```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                 'Mittal','Archana'],
                 'Marks':[99,89,99,87,90,99,89],
                }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                             'stu6','stu7'])
```

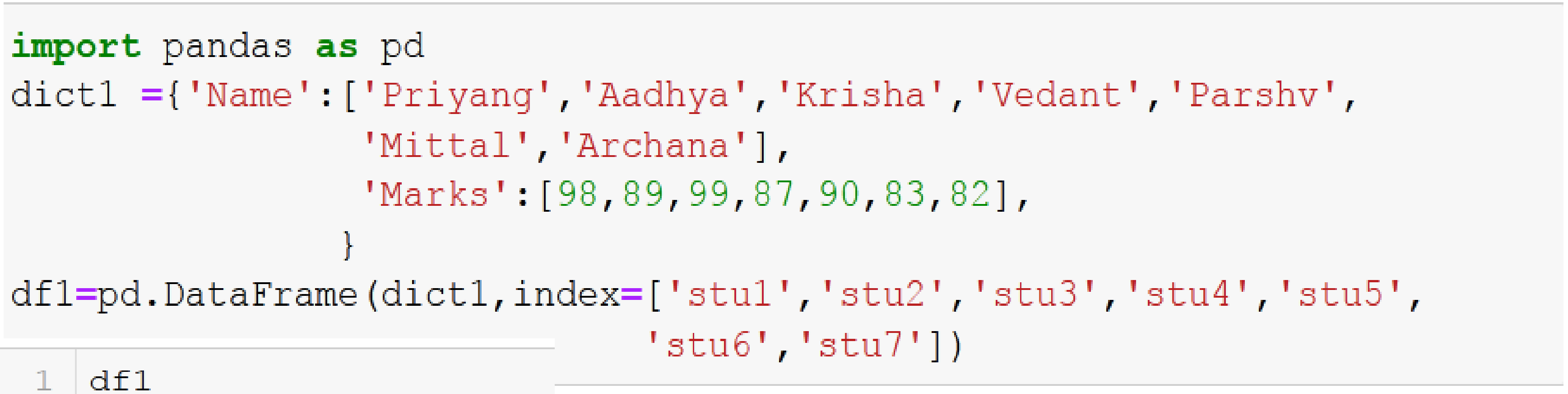
```python
df1['Marks'].value_counts()
```

```
99    3
89    2
87    1
90    1
Name: Marks, dtype: int64
```

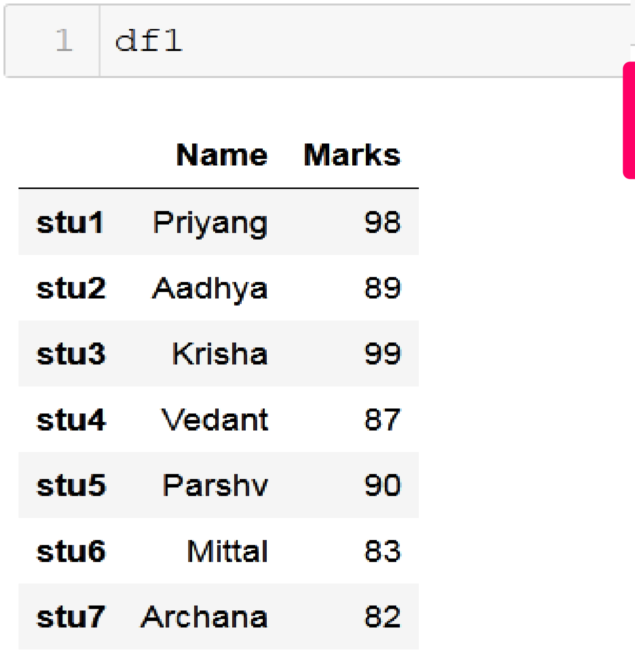
```python
df1['Marks'].count()
```

```
7
```

Bracket ([ ]) Vs. Dot(.) Selection of Column (Pandas Data Frame)

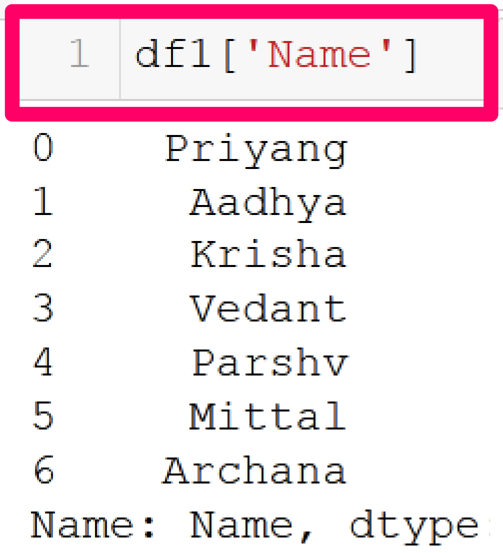


```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1,index=['stu1','stu2','stu3','stu4','stu5',
                              'stu6','stu7'])
```



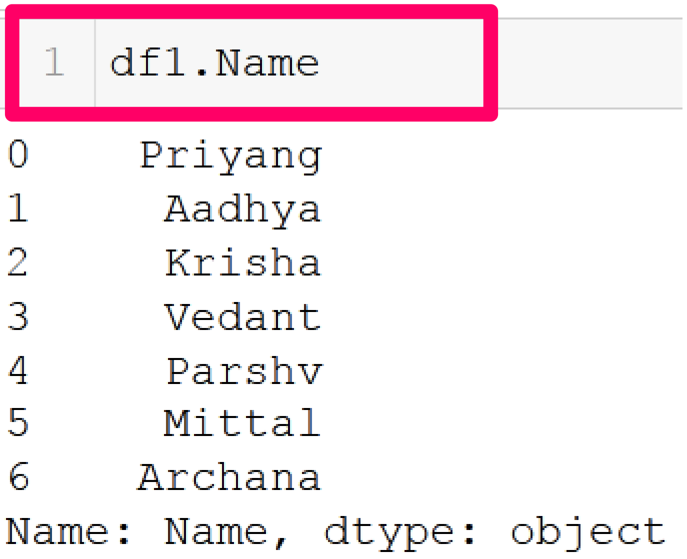
```python
df1
```

	Name	Marks
stu1	Priyang	98
stu2	Aadhya	89
stu3	Krisha	99
stu4	Vedant	87
stu5	Parshv	90
stu6	Mittal	83
stu7	Archana	82



```python
df1['Name']
```

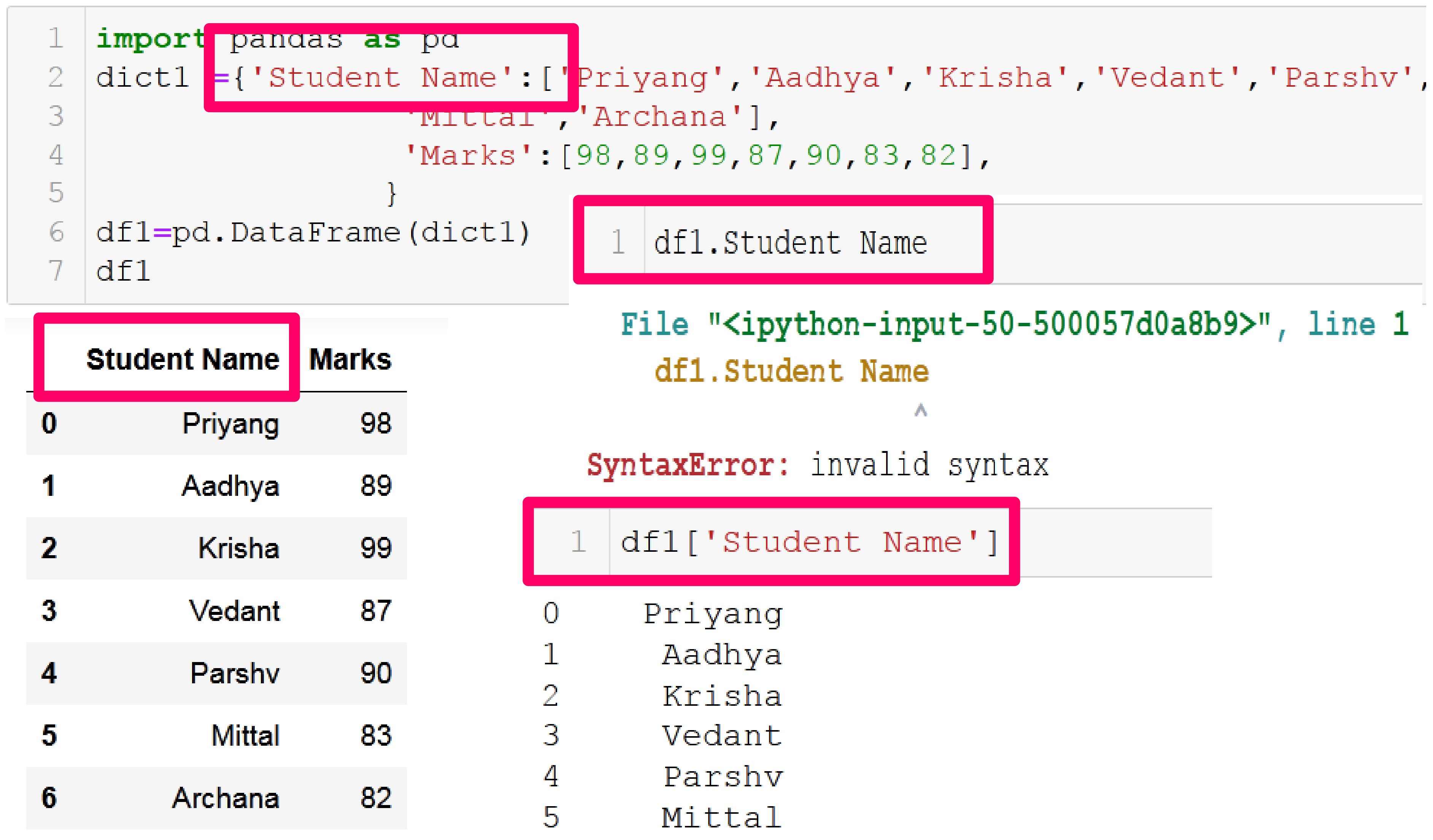
```
0     Priyang
1      Aadhya
2      Krisha
3      Vedant
4      Parshv
5      Mittal
6     Archana
Name: Name, dtype
```



```python
df1.Name
```

```
0     Priyang
1      Aadhya
2      Krisha
3      Vedant
4      Parshv
5      Mittal
6     Archana
Name: Name, dtype: object
```

1. Single Column Selection



```
import pandas as pd
dict1 ={'Student Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                        'Mittal','Archana'],
                        'Marks':[98,89,99,87,90,83,82],
                       }
df1=pd.DataFrame(dict1)
df1
```

	Student Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99
3	Vedant	87
4	Parshv	90
5	Mittal	83
6	Archana	82

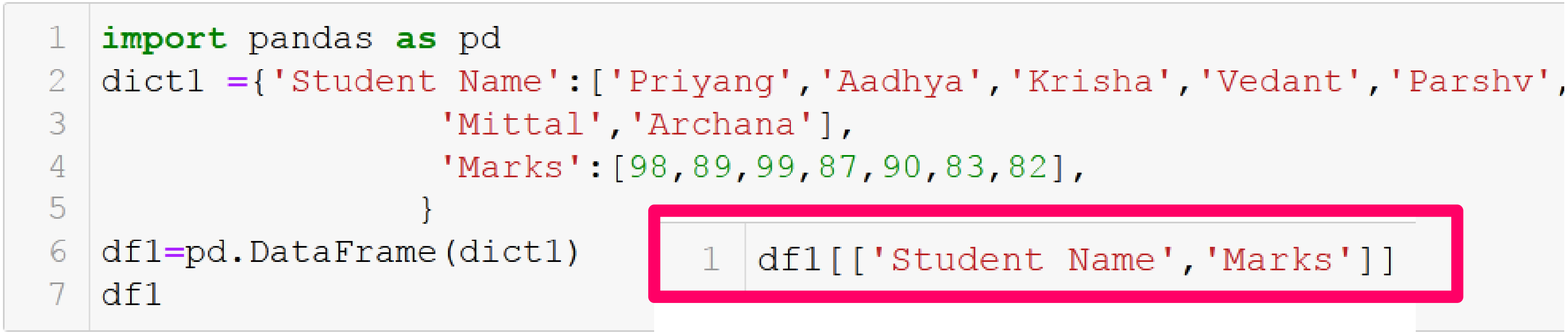
```
df1.Student Name
```

```
  File "<ipython-input-50-500057d0a8b9>", line 1
    df1.Student Name
                   ^

SyntaxError: invalid syntax
```

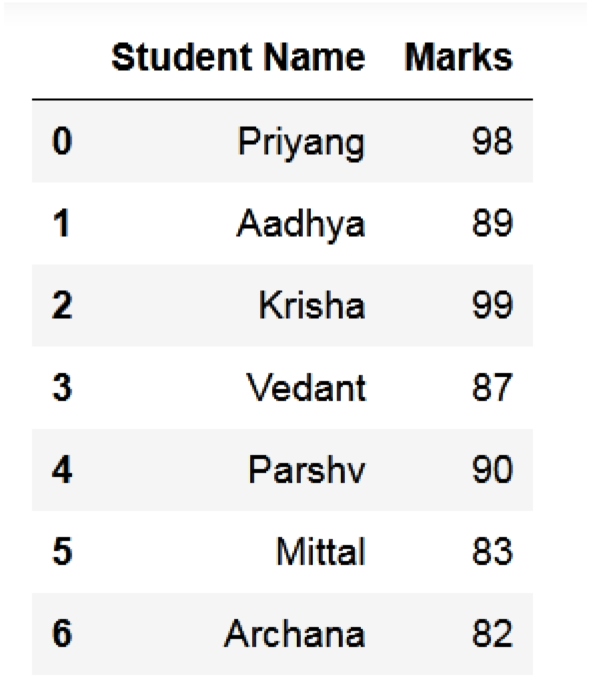
```
df1['Student Name']
```

```
0     Priyang
1      Aadhya
2      Krisha
3      Vedant
4      Parshv
5      Mittal
```

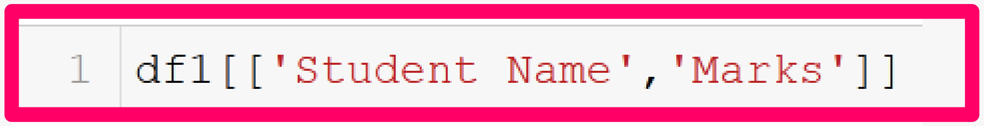
2. Multiple Columns Selection



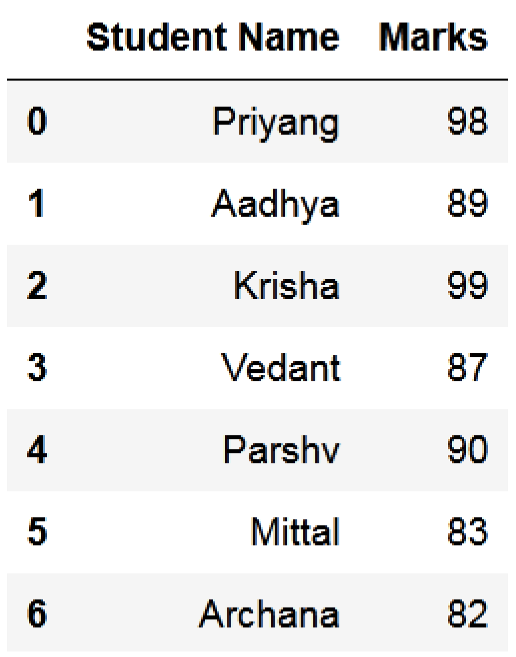
```python
import pandas as pd
dict1 ={'Student Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1)
df1
```



	Student Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99
3	Vedant	87
4	Parshv	90
5	Mittal	83
6	Archana	82

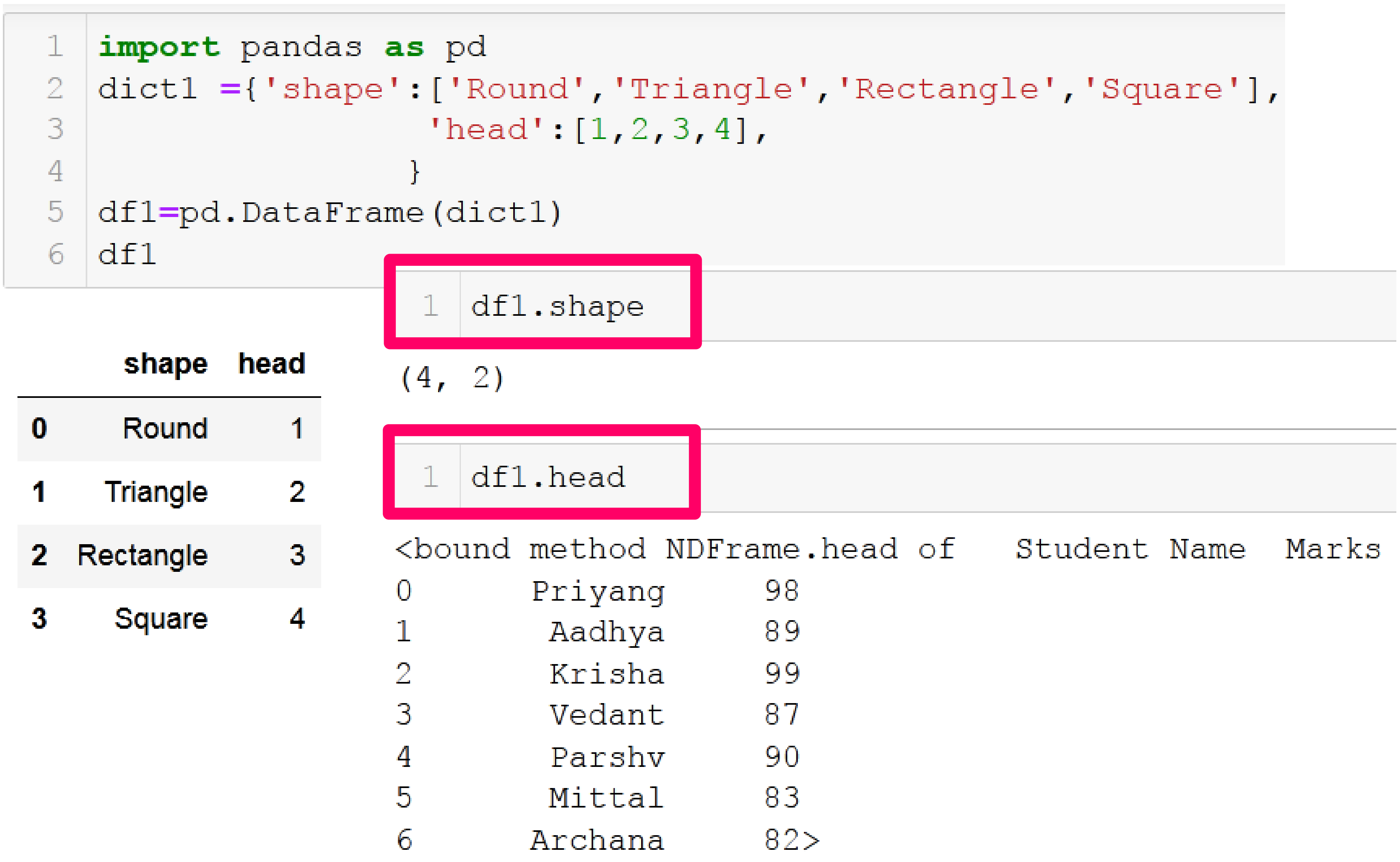


```python
df1[['Student Name','Marks']]
```



	Student Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99
3	Vedant	87
4	Parshv	90
5	Mittal	83
6	Archana	82

3. Columns with attribute Name



```python
import pandas as pd
dict1 ={'shape':['Round','Triangle','Rectangle','Square'],
                'head':[1,2,3,4],
               }
df1=pd.DataFrame(dict1)
df1
```

	shape	head
0	Round	1
1	Triangle	2
2	Rectangle	3
3	Square	4

```python
df1.shape
```

```
(4, 2)
```

```python
df1.head
```

```
<bound method NDFrame.head of   Student Name  Marks
0      Priyang     98
1       Aadhya     89
2       Krisha     99
3       Vedant     87
4       Parshv     90
5       Mittal     83
6      Archana     82>
```

Pandas Data Frame (Accessing Rows & Columns) Using loc & iloc

loc

- DataFrame.loc : Access a group of rows and columns by label(s).
- loc is inclusive both the sides [Inclusive, Inclusive]

iloc

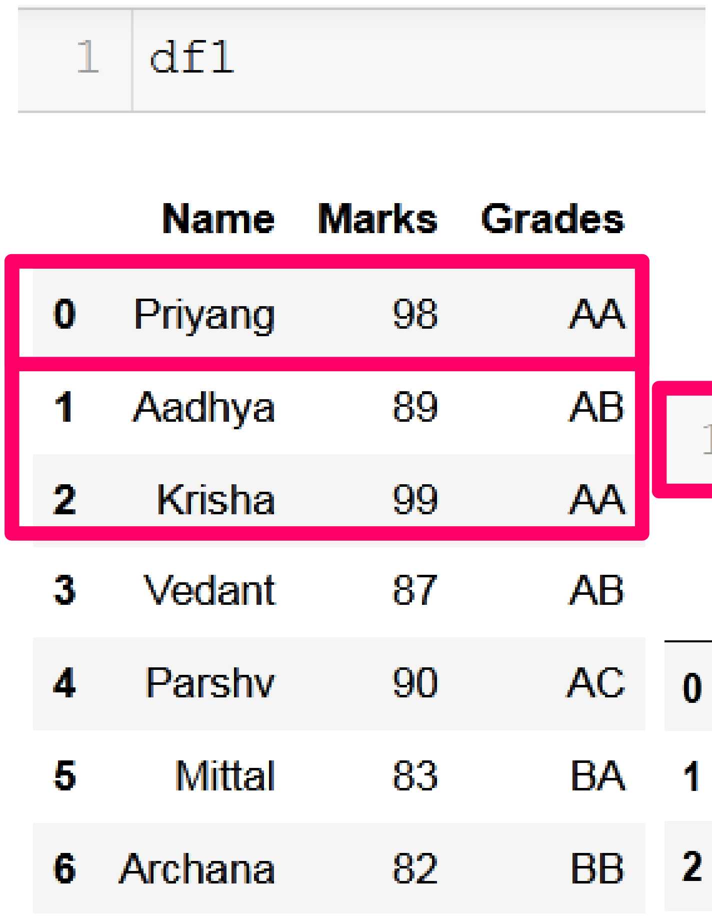
- DataFrame.iloc : Access a group of rows and columns by integer position(s).
- [First Inclusive , Last Exclusive]

Pandas Data Frame (Accessing Rows & Columns) Using loc & iloc

dataframe.loc[Row,Column]

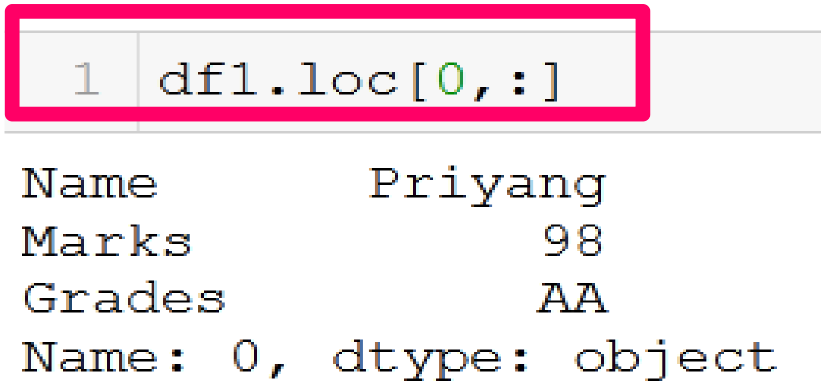
dataframe.loc [Row(s) I want,Column(s) I want]

Pandas Data Frame (Accessing Rows & Columns) Using loc & iloc



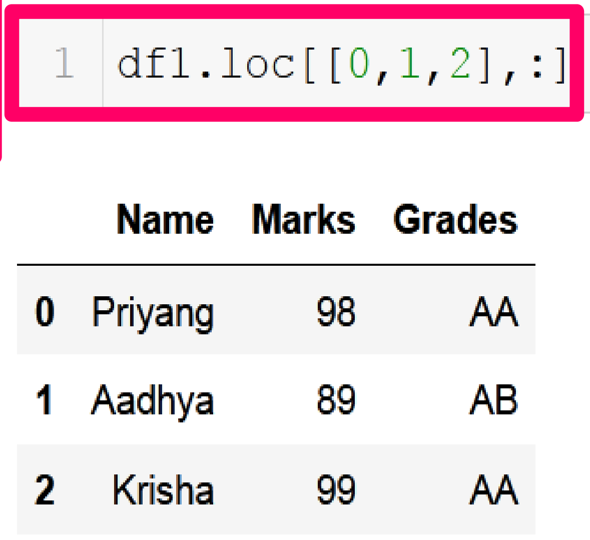
```
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB



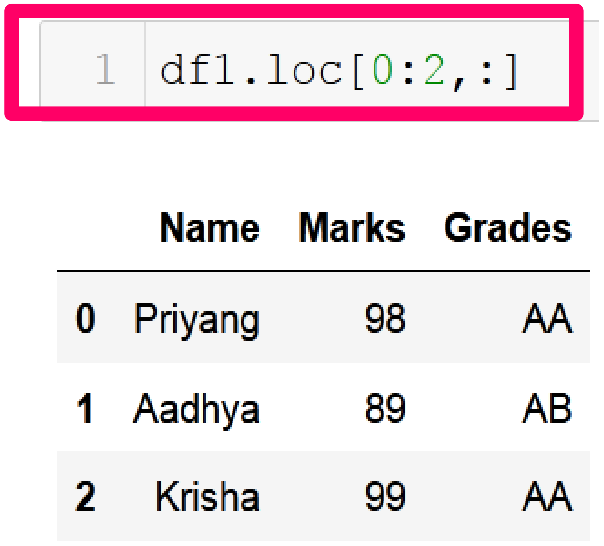
```
df1.loc[0,:]
```

```
Name       Priyang
Marks           98
Grades          AA
Name: 0, dtype: object
```



```
df1.loc[[0,1,2],:]
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA

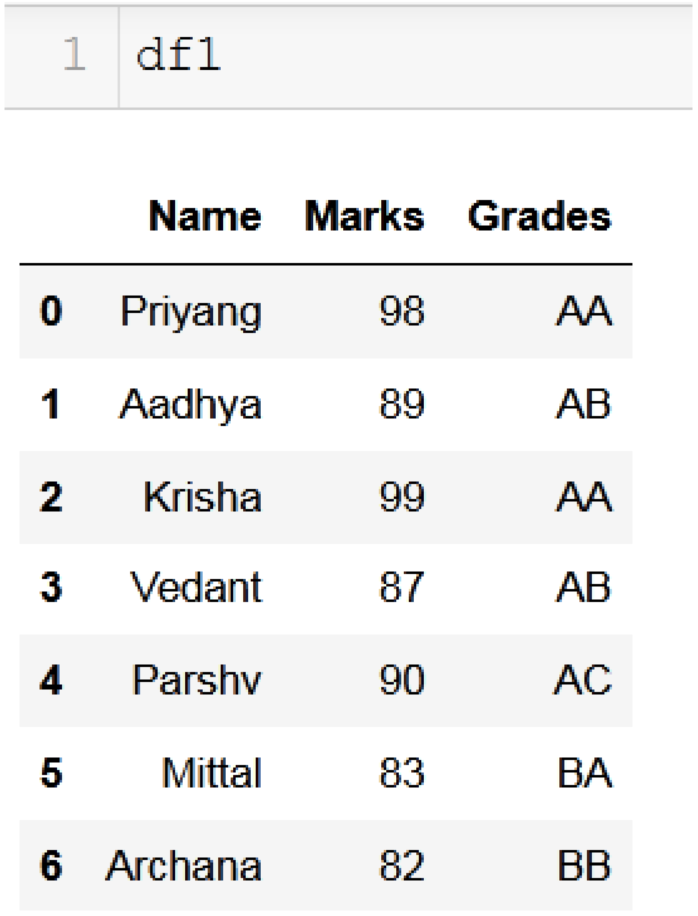


```
df1.loc[0:2,:]
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA

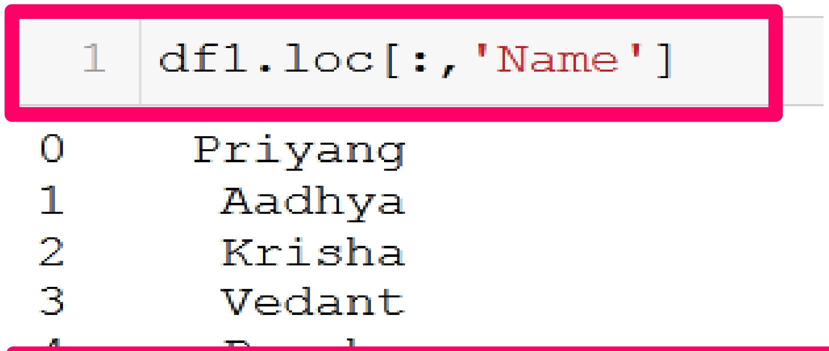
Note: loc is inclusive both the sides

Pandas Data Frame (Accessing Rows & Columns) Using loc & iloc



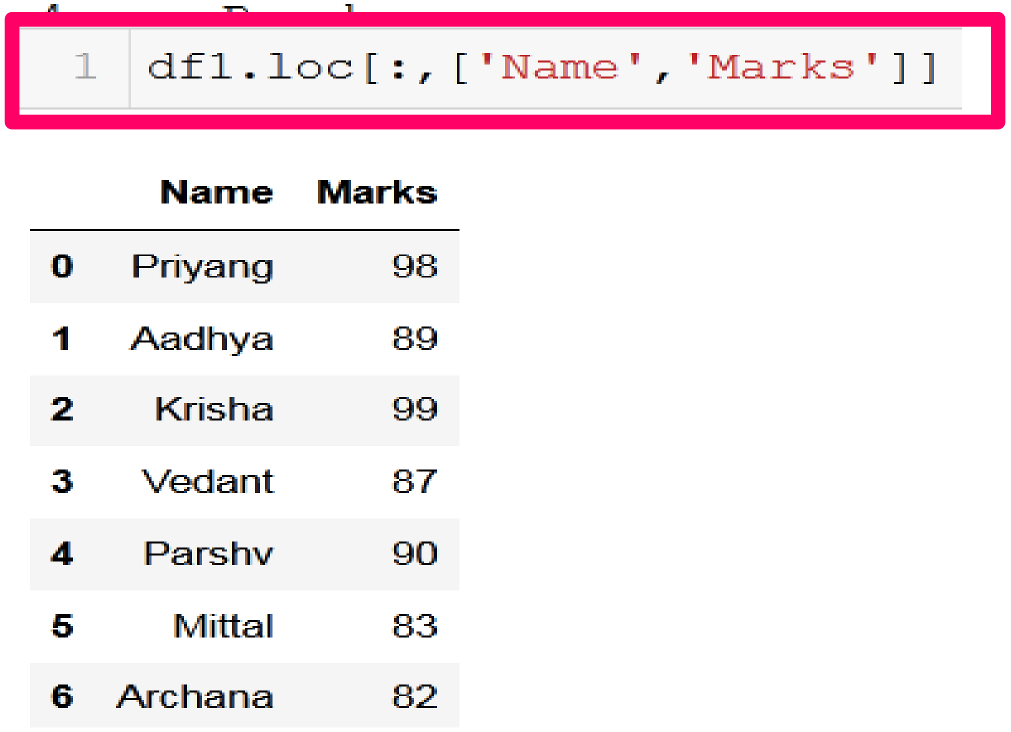
```
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB



```
df1.loc[:,'Name']
```

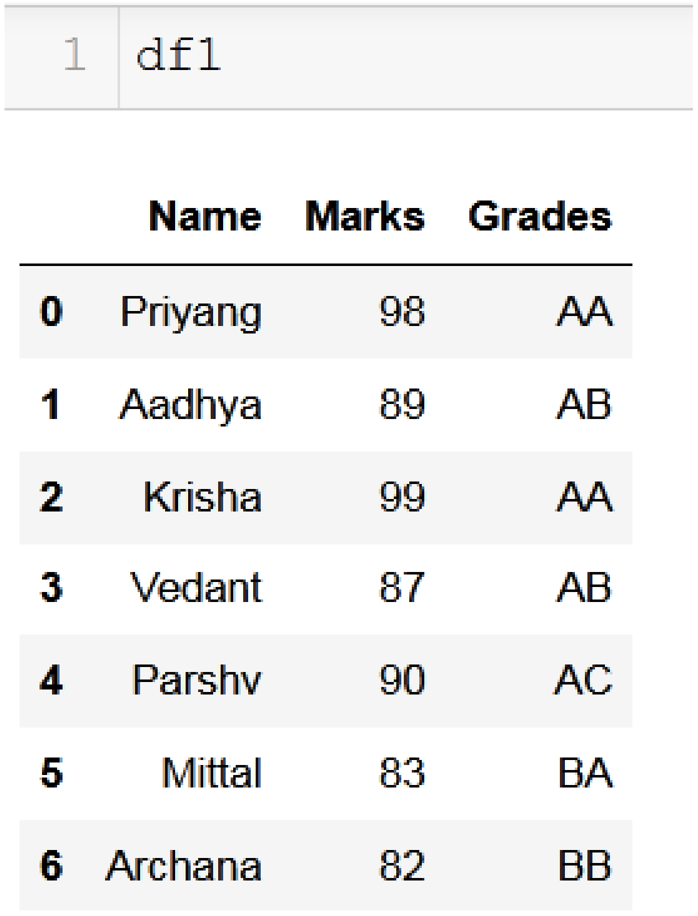
```
0      Priyang
1       Aadhya
2       Krisha
3       Vedant
```



```
df1.loc[:,['Name','Marks']]
```

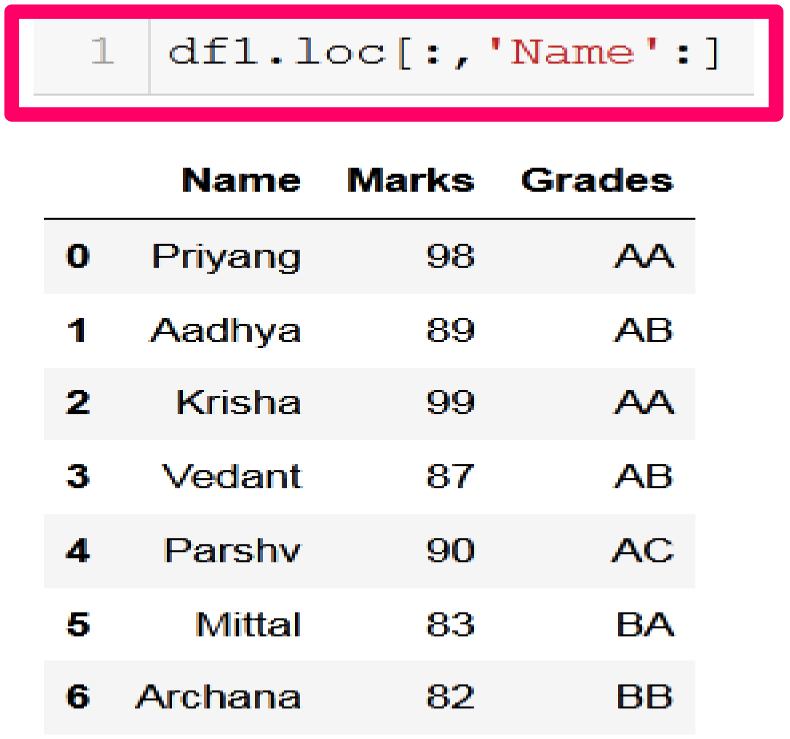
	Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99
3	Vedant	87
4	Parshv	90
5	Mittal	83
6	Archana	82

Pandas Data Frame (Accessing Rows & Columns) Using loc & iloc



```python
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB



```python
df1.loc[:,'Name':]
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

Pandas Data Frame (Accessing Rows & Columns) Using loc & iloc

- **DataFrame.iloc : Access a group of rows and columns by integer position(s). [First Inclusive , Last Exclusive]**

```
df1
```

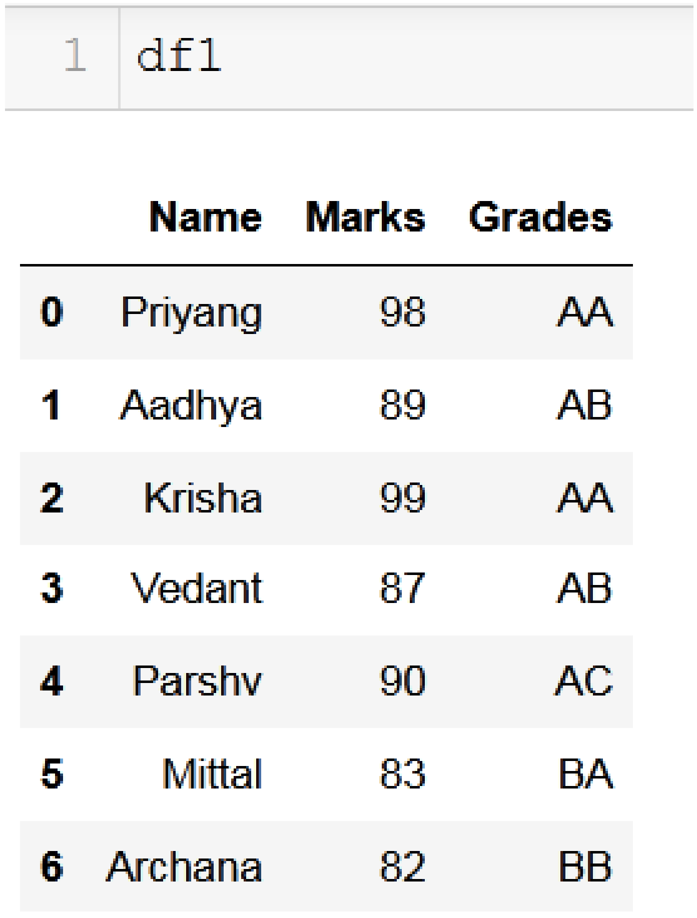
	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

```
df1.iloc[:,0]
```

```
0      Priyang
1       Aadhya
2       Krisha
3       Vedant
4       Parshv
5       Mittal
6      Archana
Name: Name, dtype: object
```

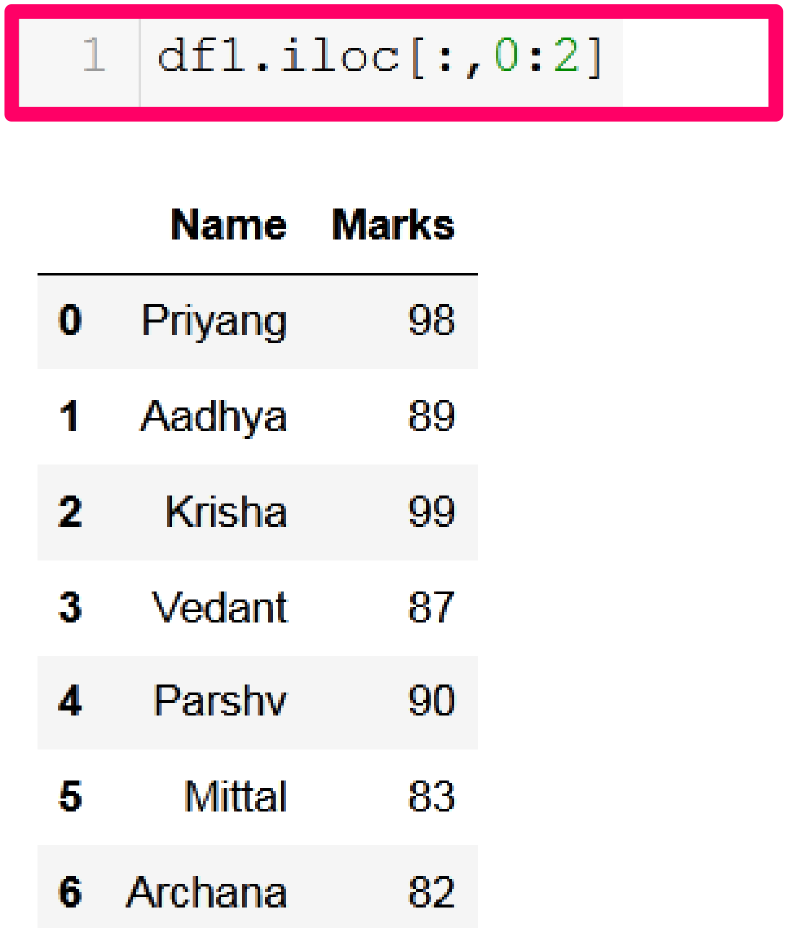
Pandas Data Frame (Accessing Rows & Columns) Using loc & iloc

- **DataFrame.iloc : Access a group of rows and columns by integer position(s). [First Inclusive , Last Exclusive]**



```
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

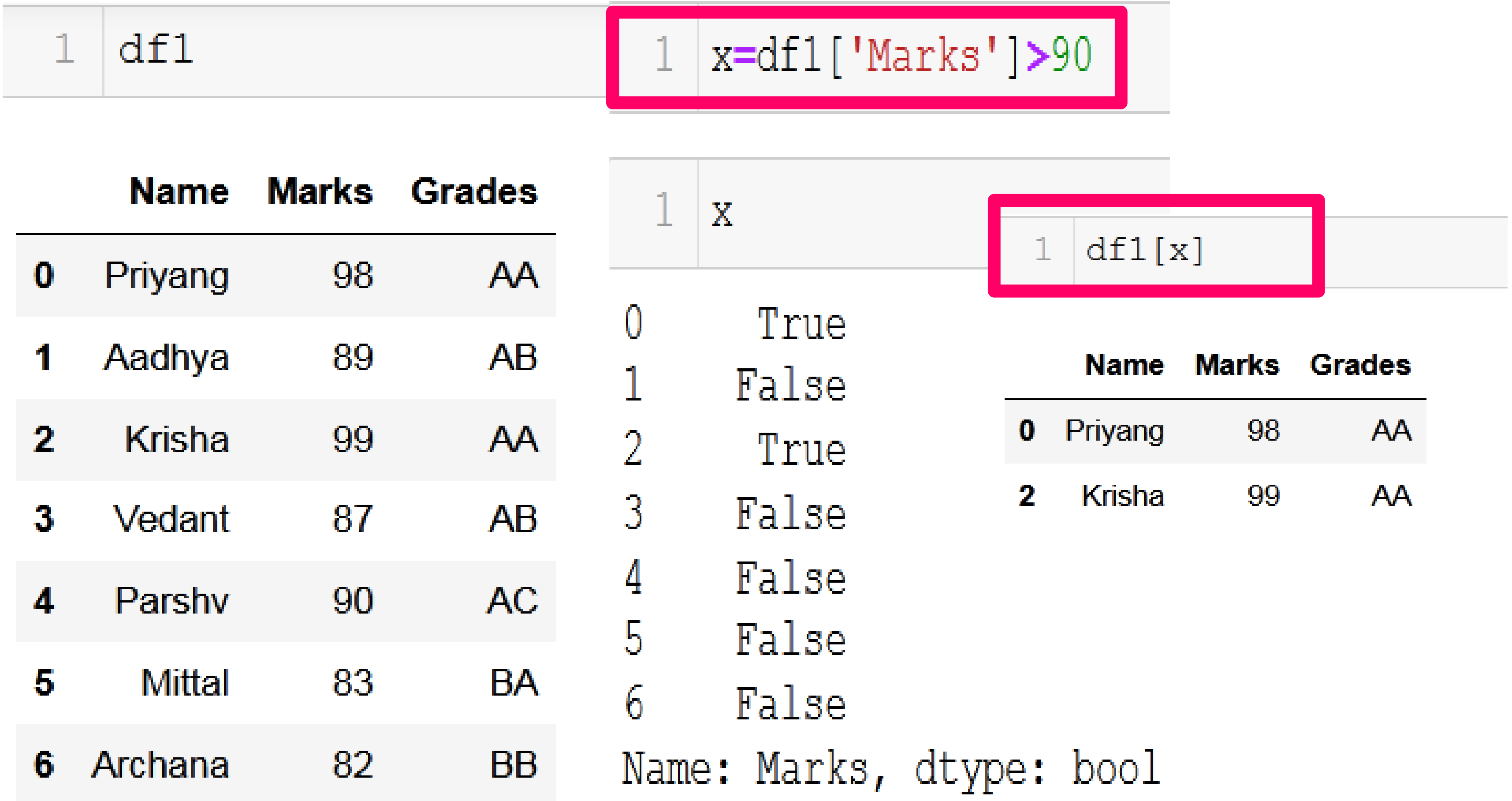


```
df1.iloc[:,0:2]
```

	Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99
3	Vedant	87
4	Parshv	90
5	Mittal	83
6	Archana	82

Conditional Selection (Pandas Data Frame)

- Display record of the students who have scored more than 90



```
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

```
x=df1['Marks']>90
```

```
x
```

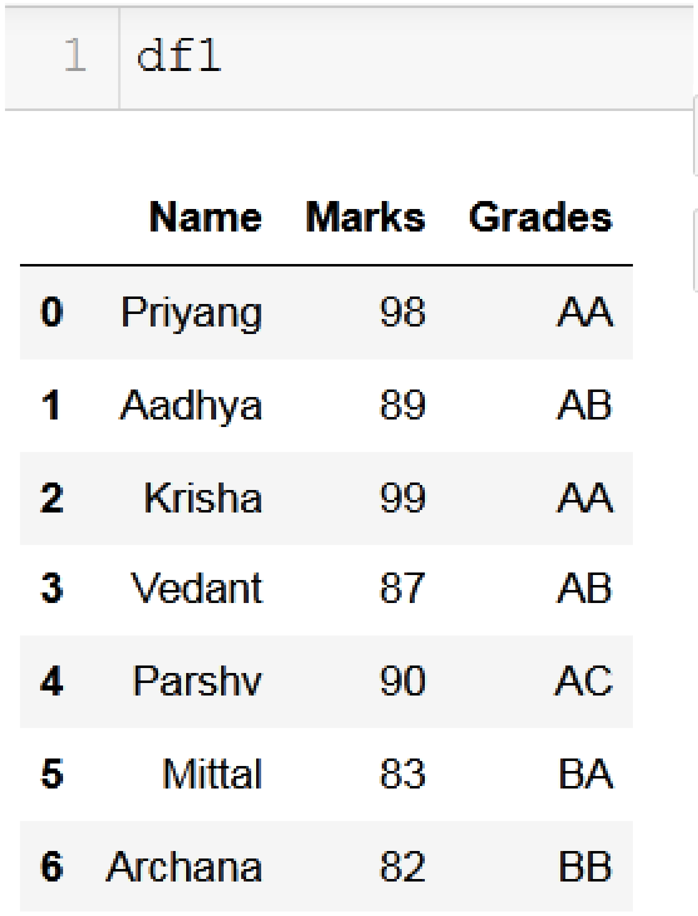
```
0     True
1    False
2     True
3    False
4    False
5    False
6    False
Name: Marks, dtype: bool
```

```
df1[x]
```

	Name	Marks	Grades
0	Priyang	98	AA
2	Krisha	99	AA

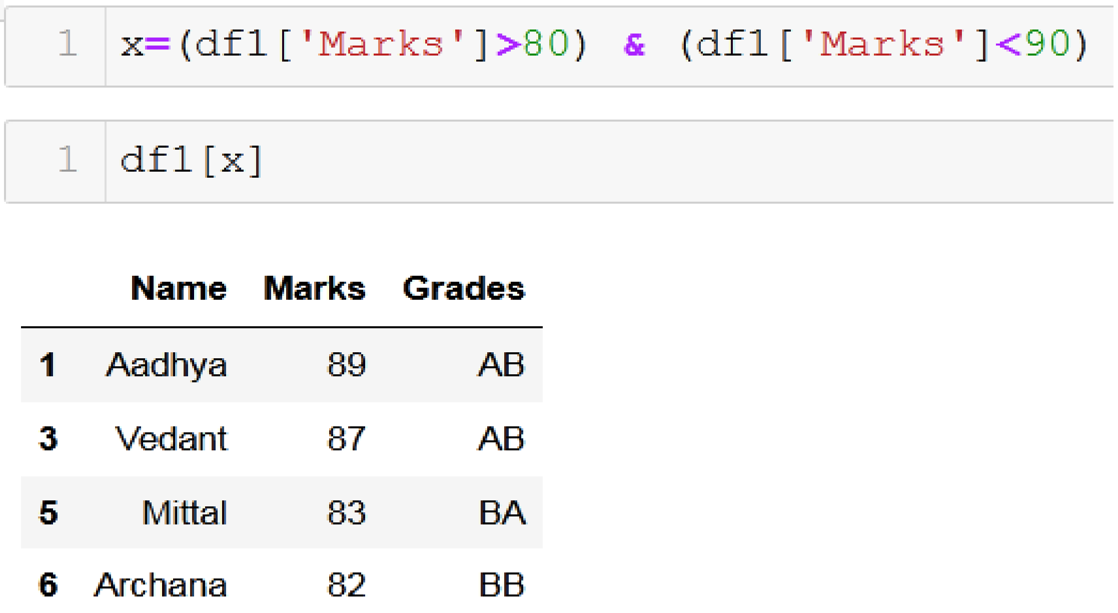
Conditional Selection (Pandas Data Frame)

- Display record of the students who have scored greater than 80 but less than 90



```
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB



```
x=(df1['Marks']>80) & (df1['Marks']<90)
```

```
df1[x]
```

	Name	Marks	Grades
1	Aadhya	89	AB
3	Vedant	87	AB
5	Mittal	83	BA
6	Archana	82	BB

Adding Column to Data Frame

```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
               }
df1=pd.DataFrame(dict1)
```

```python
df1
```

	Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99
3	Vedant	87
4	Parshv	90
5	Mittal	83
6	Archana	82

```python
s1=pd.Series(['AA','AB','AA','AB','AC','BA','BB'])
```

```python
df1['Grades']=s1
```

```python
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC

Inserting Column to Data Frame

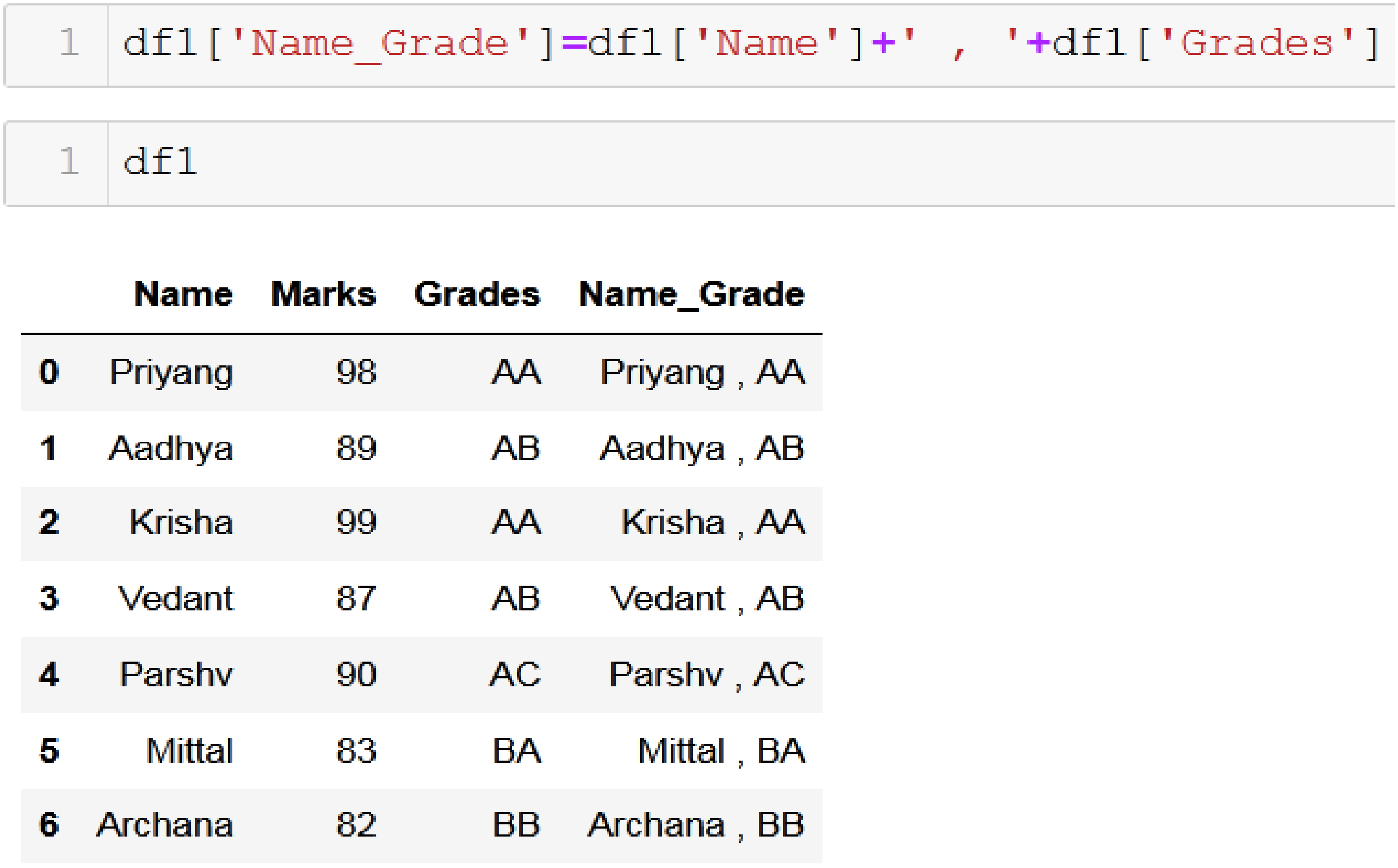
```
s1=pd.Series(['AA','AB','AA','AB','AC','BA','BB'])
```

```
df1.insert(1,'Grade',s1)   # loc,column, value
```

```
df1
```

	Name	Grade	Marks
0	Priyang	AA	98
1	Aadhya	AB	89
2	Krisha	AA	99
3	Vedant	AB	87
4	Parshv	AC	90
5	Mittal	BA	83
6	Archana	BB	82

Adding Column from existing columns to Data Frame

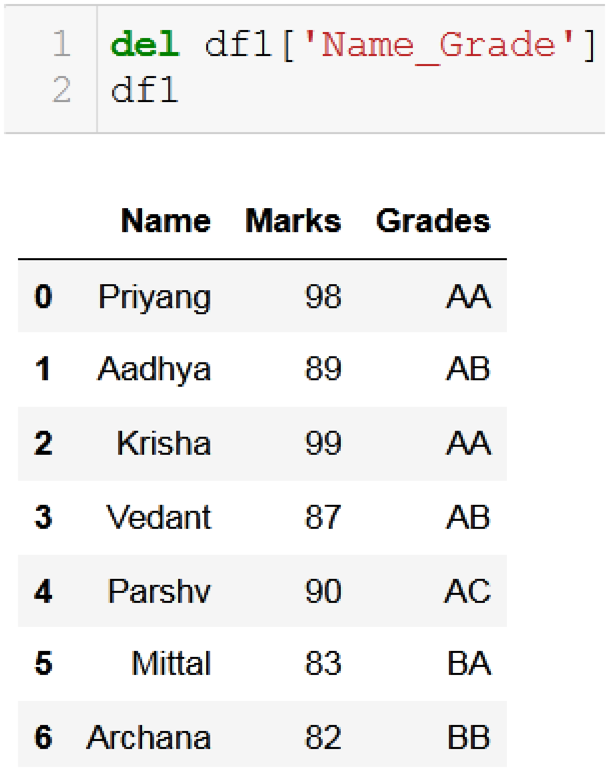


```python
df1['Name_Grade']=df1['Name']+' , '+df1['Grades']
```

```python
df1
```

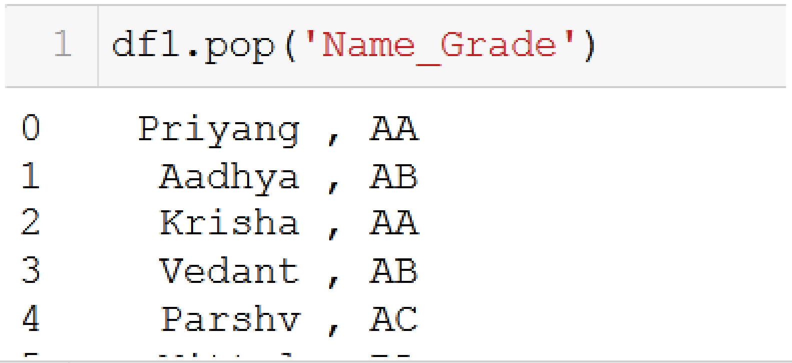
	Name	Marks	Grades	Name_Grade
0	Priyang	98	AA	Priyang , AA
1	Aadhya	89	AB	Aadhya , AB
2	Krisha	99	AA	Krisha , AA
3	Vedant	87	AB	Vedant , AB
4	Parshv	90	AC	Parshv , AC
5	Mittal	83	BA	Mittal , BA
6	Archana	82	BB	Archana , BB

Delete Column From Data Frame



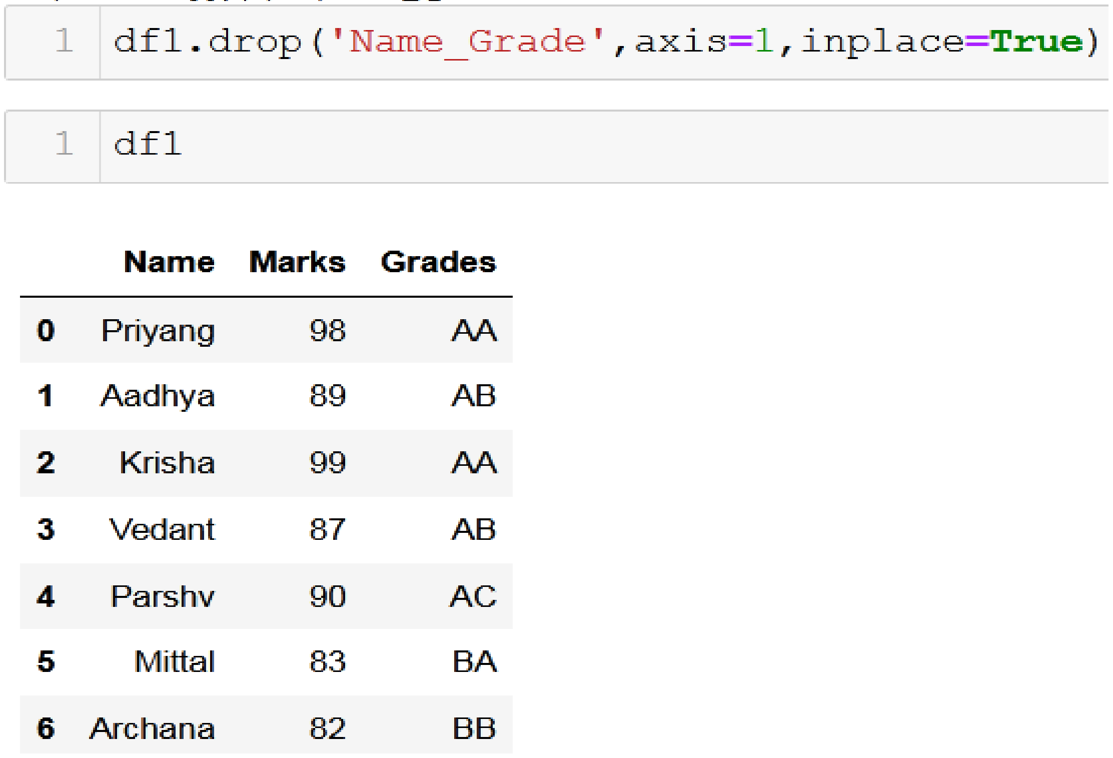
```python
del df1['Name_Grade']
df1
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB



```python
df1.pop('Name_Grade')
```

```
0     Priyang , AA
1      Aadhya , AB
2      Krisha , AA
3      Vedant , AB
4      Parshv , AC
```

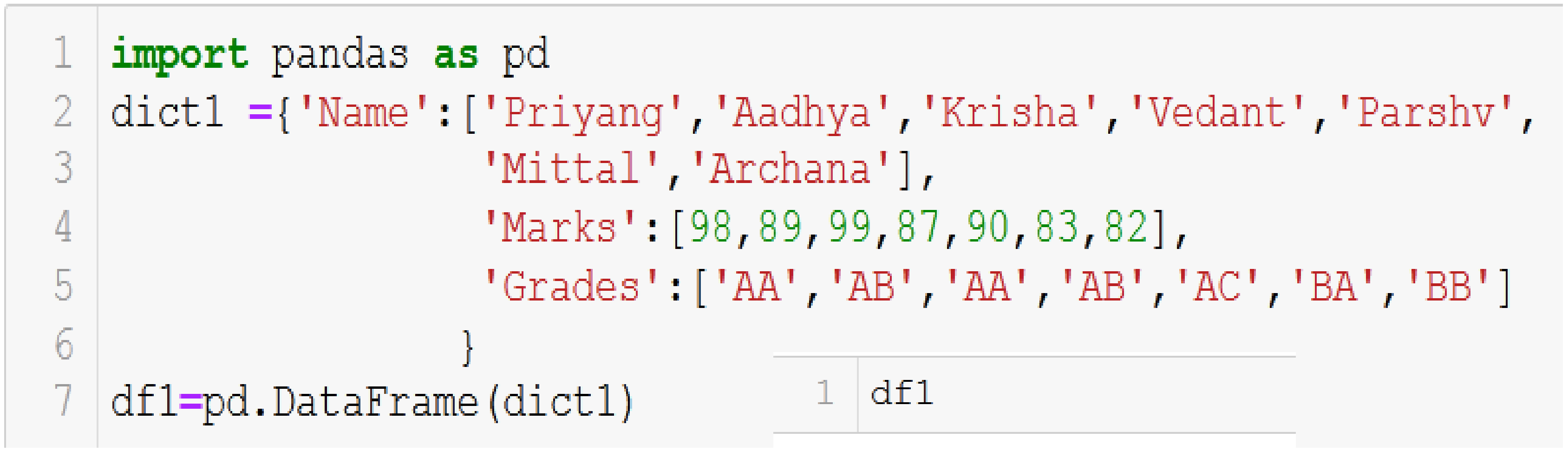


```python
df1.drop('Name_Grade',axis=1,inplace=True)
```

```python
df1
```

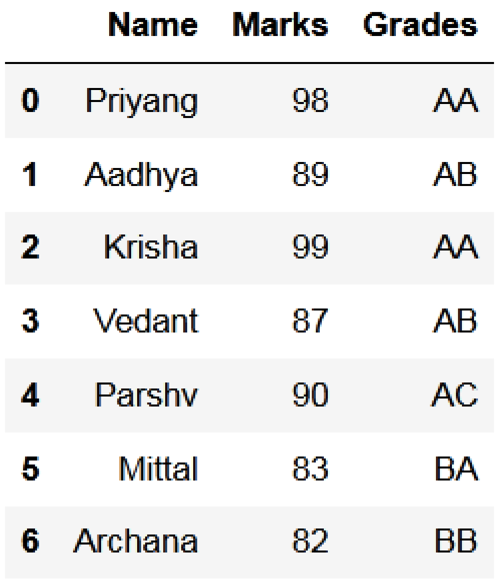
	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

Create CSV file from Pandas Data Frame



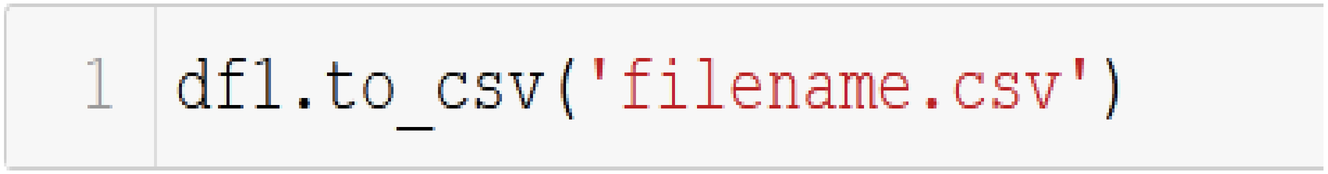
```python
import pandas as pd
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,89,99,87,90,83,82],
                'Grades':['AA','AB','AA','AB','AC','BA','BB']
               }
df1=pd.DataFrame(dict1)
```

```python
df1
```



```
      Name  Marks  Grades
0  Priyang     98      AA
1   Aadhya     89      AB
2   Krisha     99      AA
3   Vedant     87      AB
4   Parshv     90      AC
5   Mittal     83      BA
6  Archana     82      BB
```

Create a CSV File From Pandas Data Frame



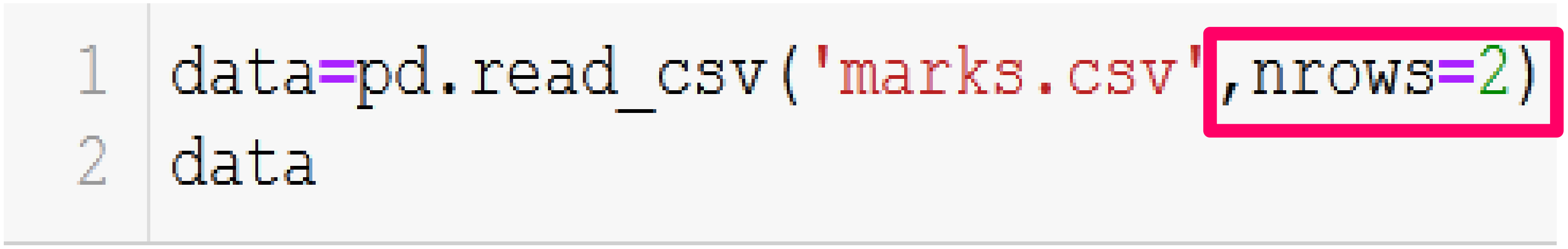
```
df1.to_csv('filename.csv')
```

Pandas read_csv()

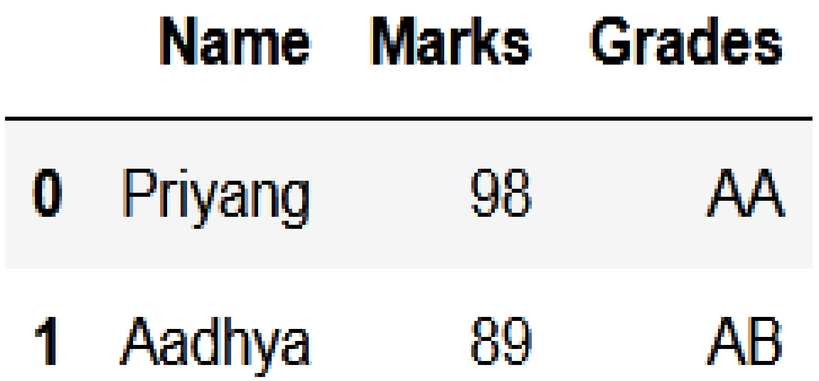
- Read a comma-separated values (CSV) file into Data Frame.

Parameters	Description
nrows	Number of rows of file to read. Useful for reading pieces of large files.
usecols	load specific columns into data frame.
skiprows	Line numbers to skip while reading csv.
index_col	Column(s) to use as the row labels of the DataFrame.
header	Row number(s) to use as the column names
names	List of column names to use.
prefix	Prefix to add to column numbers when no header, e.g. 'X' for X0, X1, ...
dtype	Data type for data or columns

Pandas read_csv() : nrows

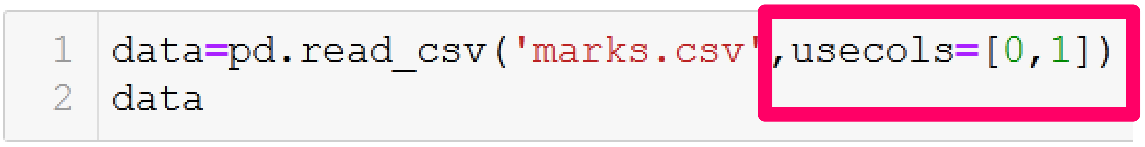


```python
data=pd.read_csv('marks.csv',nrows=2)
data
```

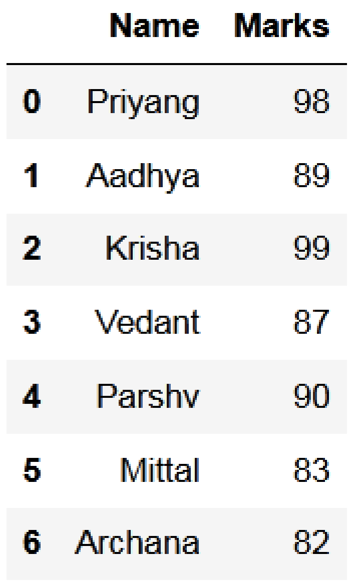


	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB

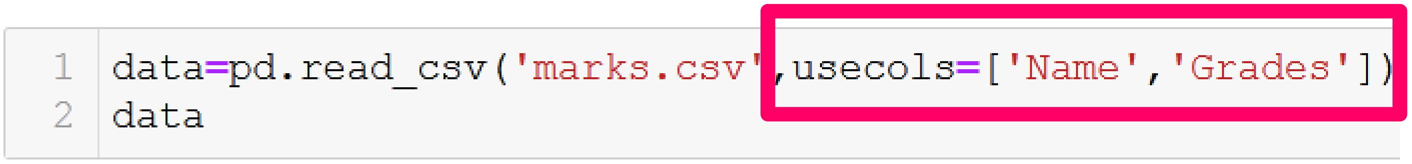
Pandas read_csv() : usecols



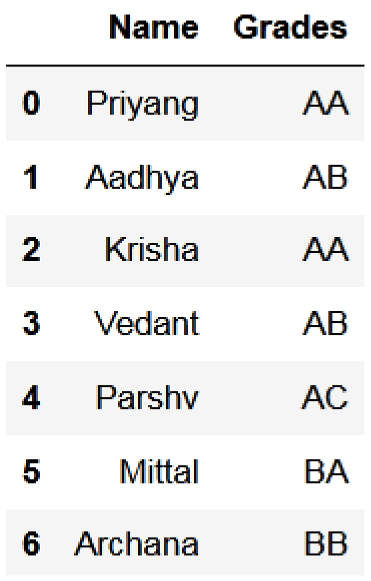
```python
data=pd.read_csv('marks.csv',usecols=[0,1])
data
```



	Name	Marks
0	Priyang	98
1	Aadhya	89
2	Krisha	99
3	Vedant	87
4	Parshv	90
5	Mittal	83
6	Archana	82

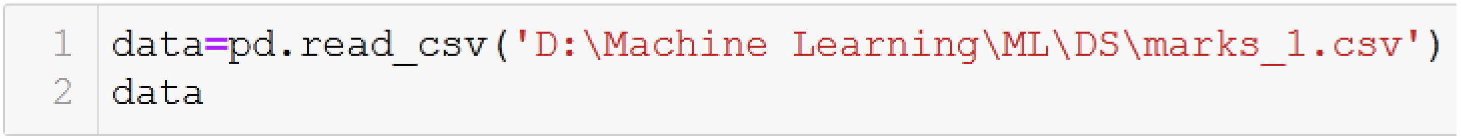


```python
data=pd.read_csv('marks.csv',usecols=['Name','Grades'])
data
```

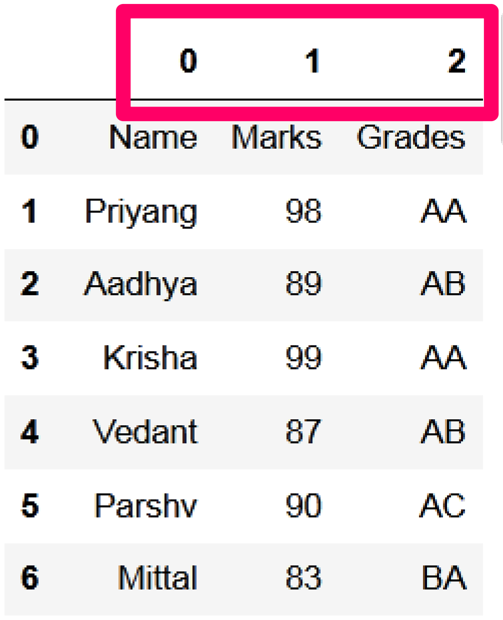


	Name	Grades
0	Priyang	AA
1	Aadhya	AB
2	Krisha	AA
3	Vedant	AB
4	Parshv	AC
5	Mittal	BA
6	Archana	BB

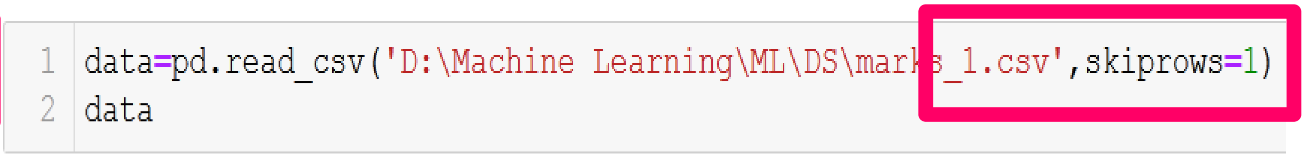
Pandas read_csv() : skiprows



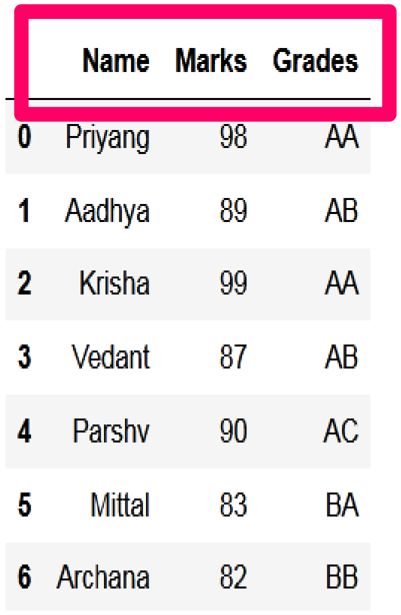
```python
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_1.csv')
data
```



	0	1	2
0	Name	Marks	Grades
1	Priyang	98	AA
2	Aadhya	89	AB
3	Krisha	99	AA
4	Vedant	87	AB
5	Parshv	90	AC
6	Mittal	83	BA

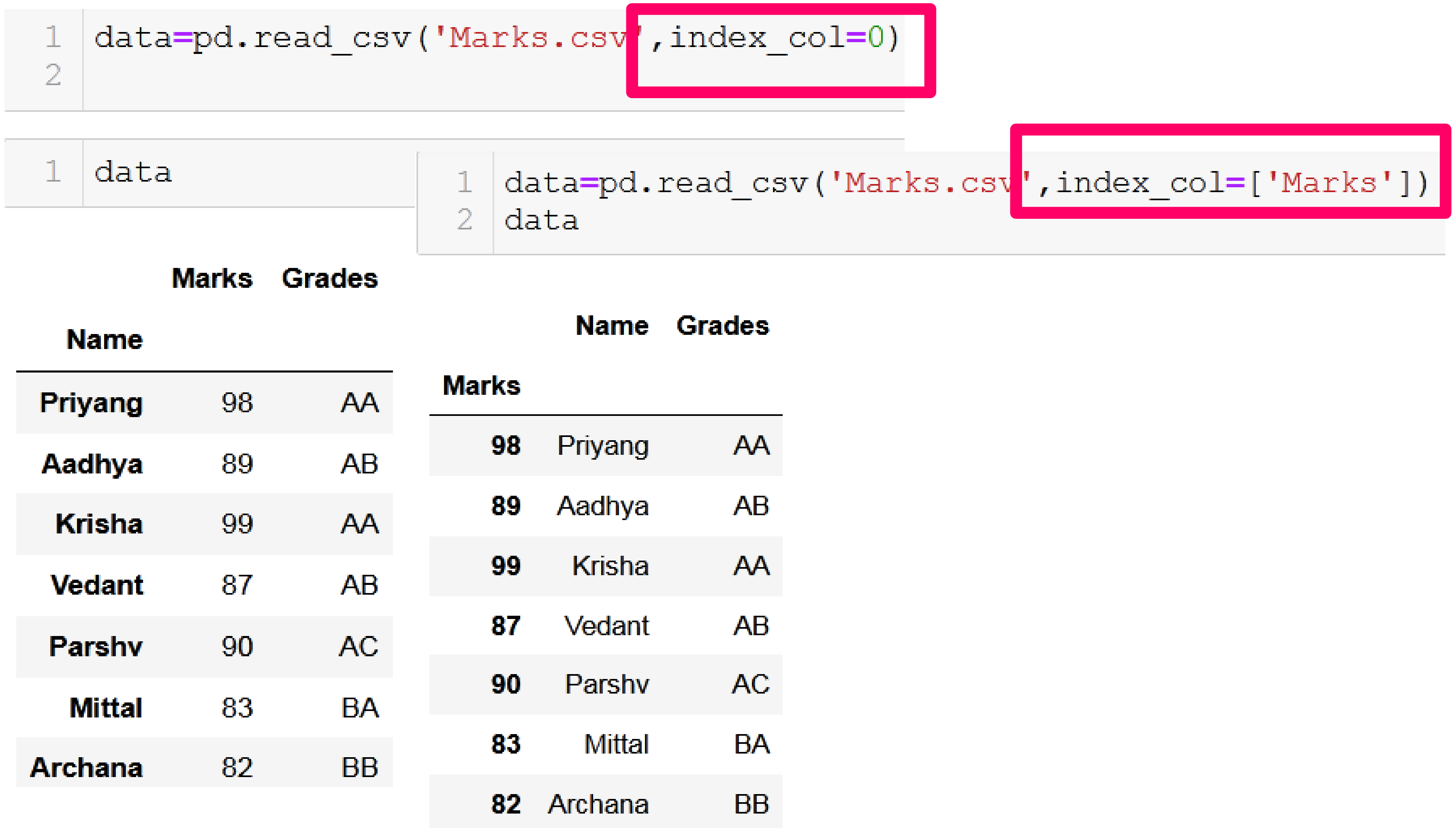


```python
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_1.csv',skiprows=1)
data
```



	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

Pandas read_csv() : index_col



```python
data=pd.read_csv('Marks.csv',index_col=0)

```

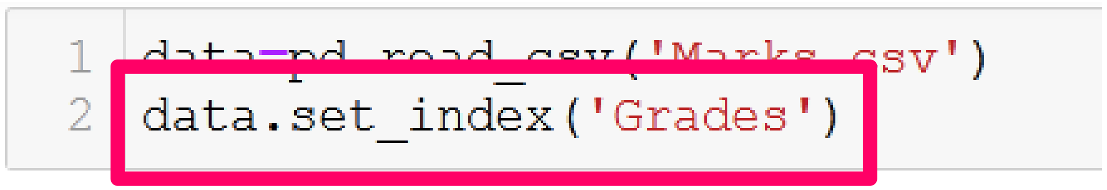
```python
data
```

	Marks	Grades
Name		
Priyang	98	AA
Aadhya	89	AB
Krisha	99	AA
Vedant	87	AB
Parshv	90	AC
Mittal	83	BA
Archana	82	BB

```python
data=pd.read_csv('Marks.csv',index_col=['Marks'])
data
```

	Name	Grades
Marks		
98	Priyang	AA
89	Aadhya	AB
99	Krisha	AA
87	Vedant	AB
90	Parshv	AC
83	Mittal	BA
82	Archana	BB

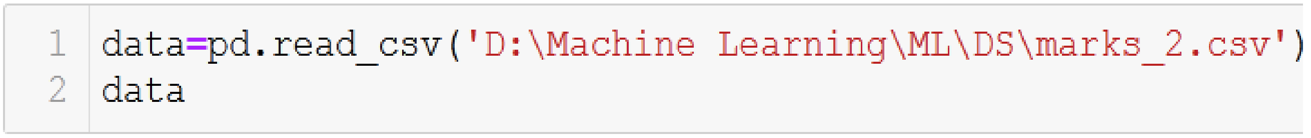
Pandas read_csv() : set_index()



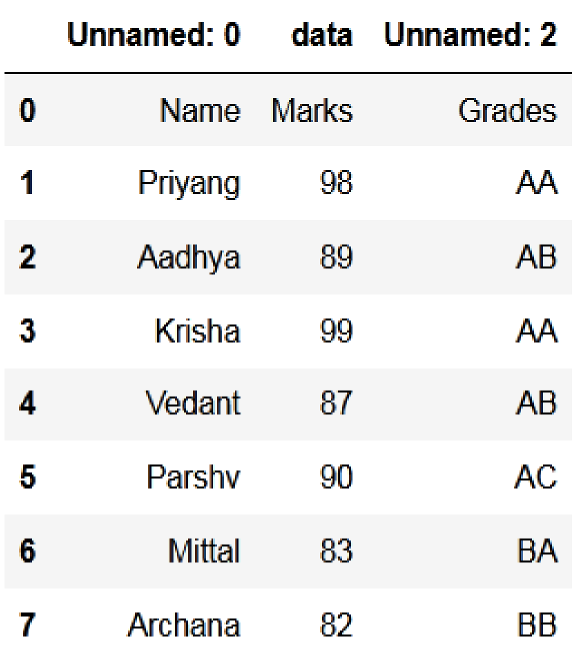
```
data=pd.read_csv('Marks.csv')
data.set_index('Grades')
```

	Name	Marks
Grades		
AA	Priyang	98
AB	Aadhya	89
AA	Krisha	99
AB	Vedant	87
AC	Parshv	90
BA	Mittal	83
BB	Archana	82

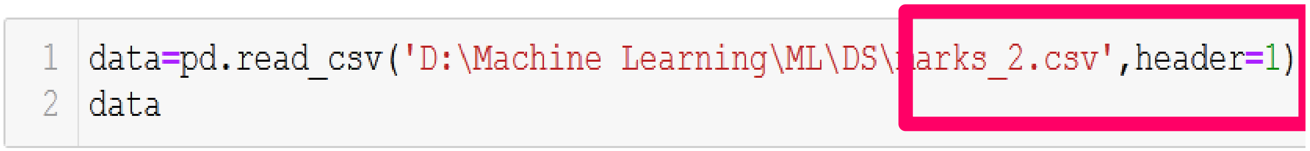
Pandas read_csv() : header



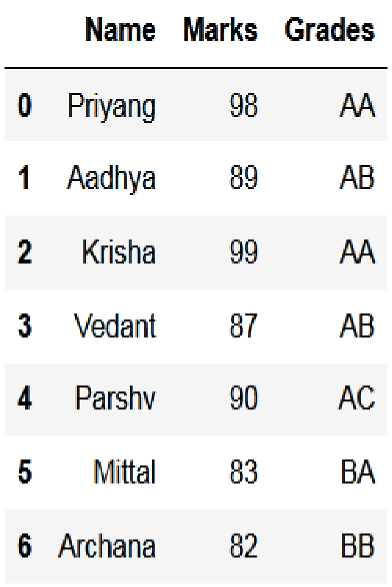
```
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2.csv')
data
```



	Unnamed: 0	data	Unnamed: 2
0	Name	Marks	Grades
1	Priyang	98	AA
2	Aadhya	89	AB
3	Krisha	99	AA
4	Vedant	87	AB
5	Parshv	90	AC
6	Mittal	83	BA
7	Archana	82	BB

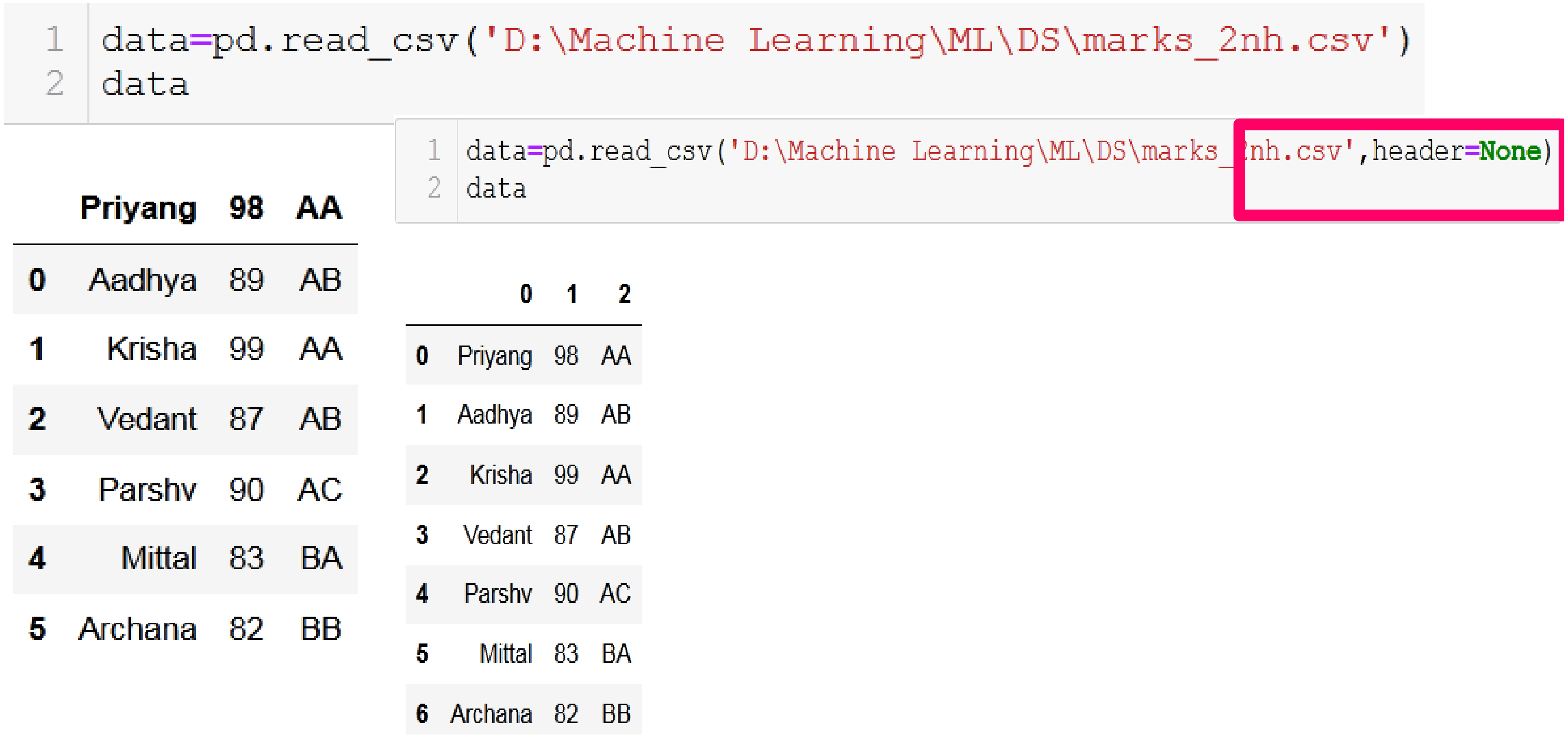


```
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2.csv',header=1)
data
```



	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

Pandas read_csv() : header



```python
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2nh.csv')
data
```

	Priyang	98	AA
0	Aadhya	89	AB
1	Krisha	99	AA
2	Vedant	87	AB
3	Parshv	90	AC
4	Mittal	83	BA
5	Archana	82	BB

```python
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2nh.csv',header=None)
data
```

	0	1	2
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

Pandas read_csv() : names

```python
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2nh.csv')
data
```

	Priyang	98	AA
0	Aadhya	89	AB
1	Krisha	99	AA
2	Vedant	87	AB
3	Parshv	90	AC
4	Mittal	83	BA
5	Archana	82	BB

```python
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2nh.csv',header=None,
                 names=['Name','Marks','Grades'])
data
```

	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

Pandas read_csv() : prefix

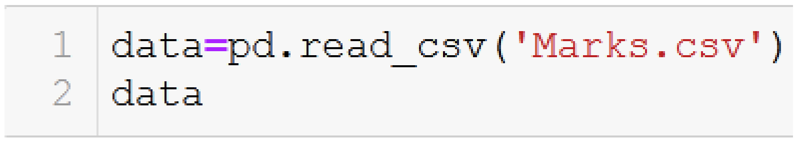
```
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2nh.csv')
data
```

	Priyang	98	AA
0	Aadhya	89	AB
1	Krisha	99	AA
2	Vedant	87	AB
3	Parshv	90	AC
4	Mittal	83	BA
5	Archana	82	BB

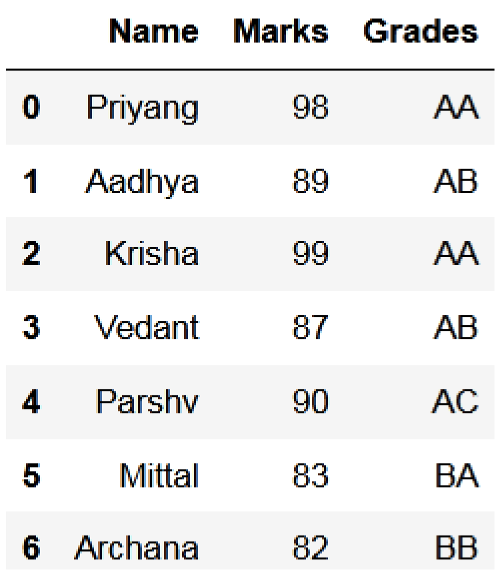
```
data=pd.read_csv('D:\Machine Learning\ML\DS\marks_2nh.csv',
                 header=None,prefix='data')
data
```

	data0	data1	data2
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

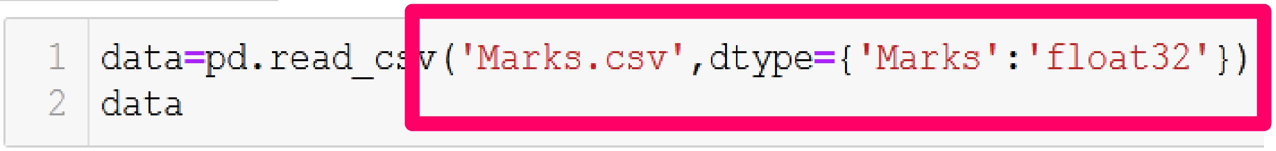
Pandas read_csv() : dtype



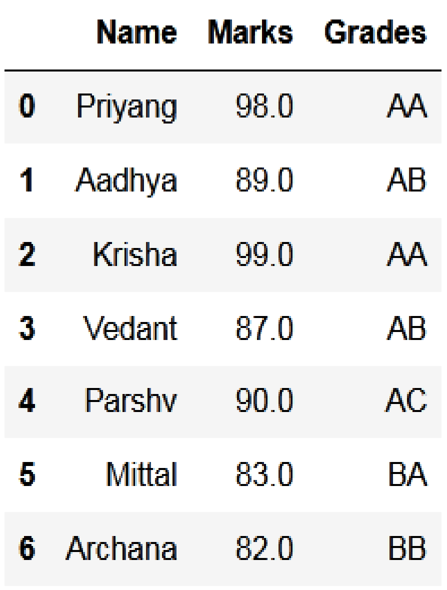
```python
data=pd.read_csv('Marks.csv')
data
```



	Name	Marks	Grades
0	Priyang	98	AA
1	Aadhya	89	AB
2	Krisha	99	AA
3	Vedant	87	AB
4	Parshv	90	AC
5	Mittal	83	BA
6	Archana	82	BB

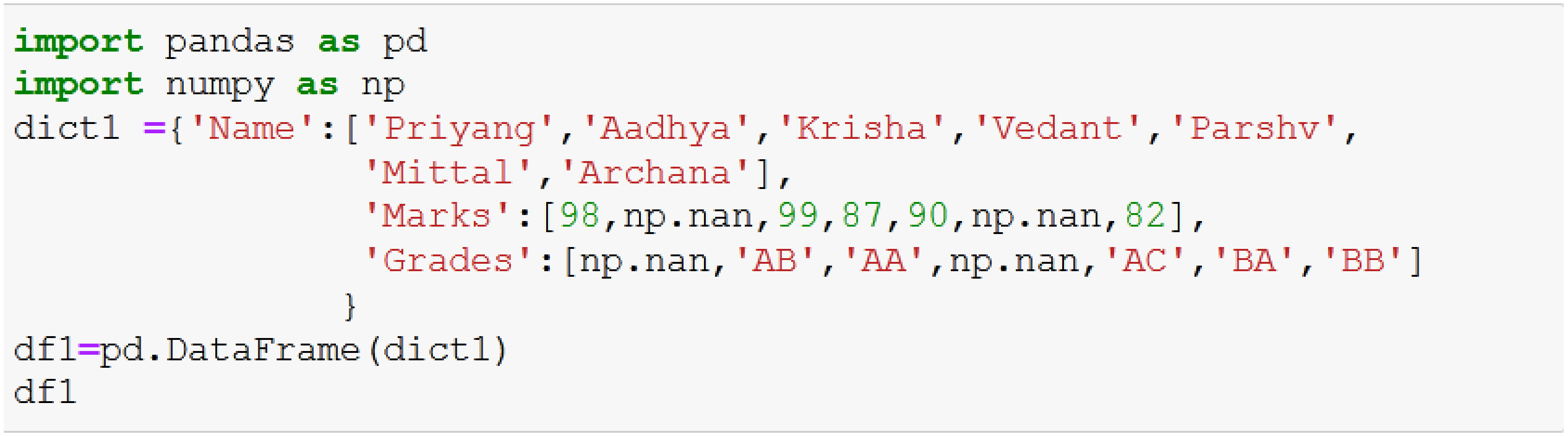


```python
data=pd.read_csv('Marks.csv',dtype={'Marks':'float32'})
data
```

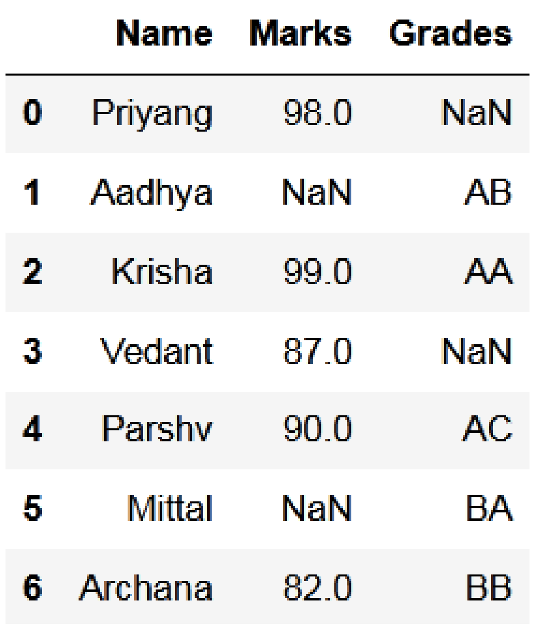


	Name	Marks	Grades
0	Priyang	98.0	AA
1	Aadhya	89.0	AB
2	Krisha	99.0	AA
3	Vedant	87.0	AB
4	Parshv	90.0	AC
5	Mittal	83.0	BA
6	Archana	82.0	BB

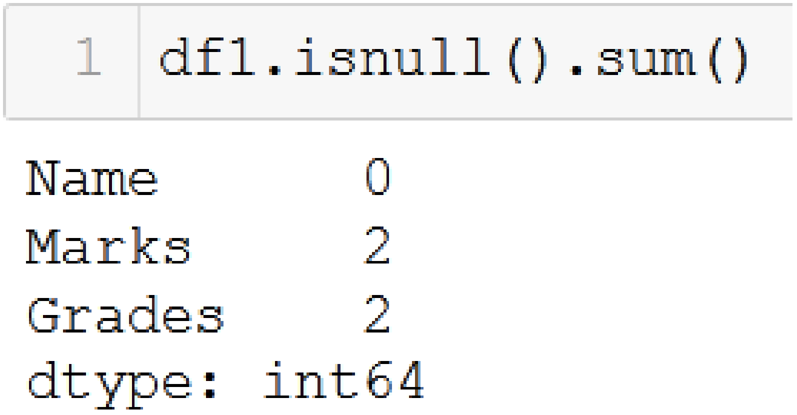
Handling Missing Values



```python
import pandas as pd
import numpy as np
dict1 ={'Name':['Priyang','Aadhya','Krisha','Vedant','Parshv',
                'Mittal','Archana'],
                'Marks':[98,np.nan,99,87,90,np.nan,82],
                'Grades':[np.nan,'AB','AA',np.nan,'AC','BA','BB']
               }
df1=pd.DataFrame(dict1)
df1
```



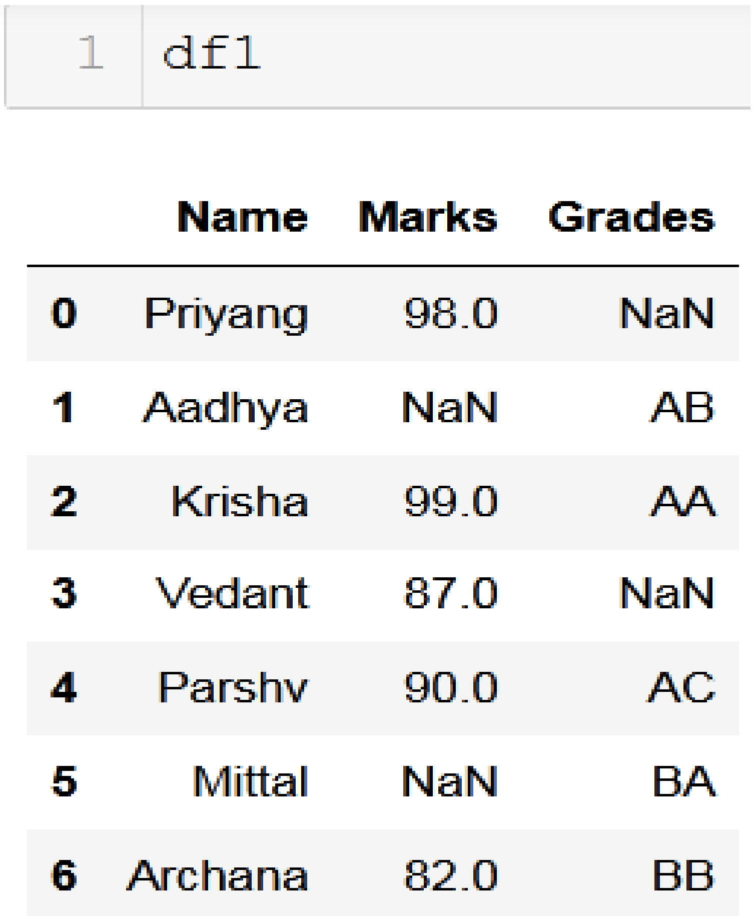
	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB



```python
df1.isnull().sum()
```

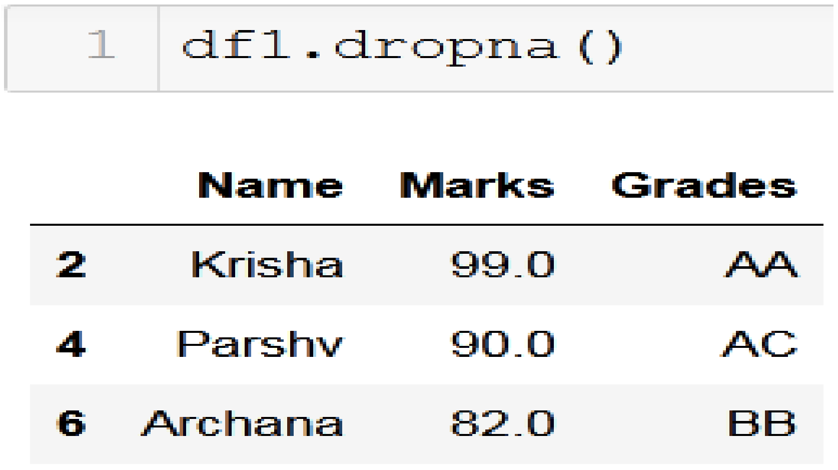
```
Name      0
Marks     2
Grades    2
dtype: int64
```

Handling Missing Values : dropna()



```
df1
```

	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB



```
df1.dropna()
```

	Name	Marks	Grades
2	Krisha	99.0	AA
4	Parshv	90.0	AC
6	Archana	82.0	BB

Handling Missing Values : dropna()

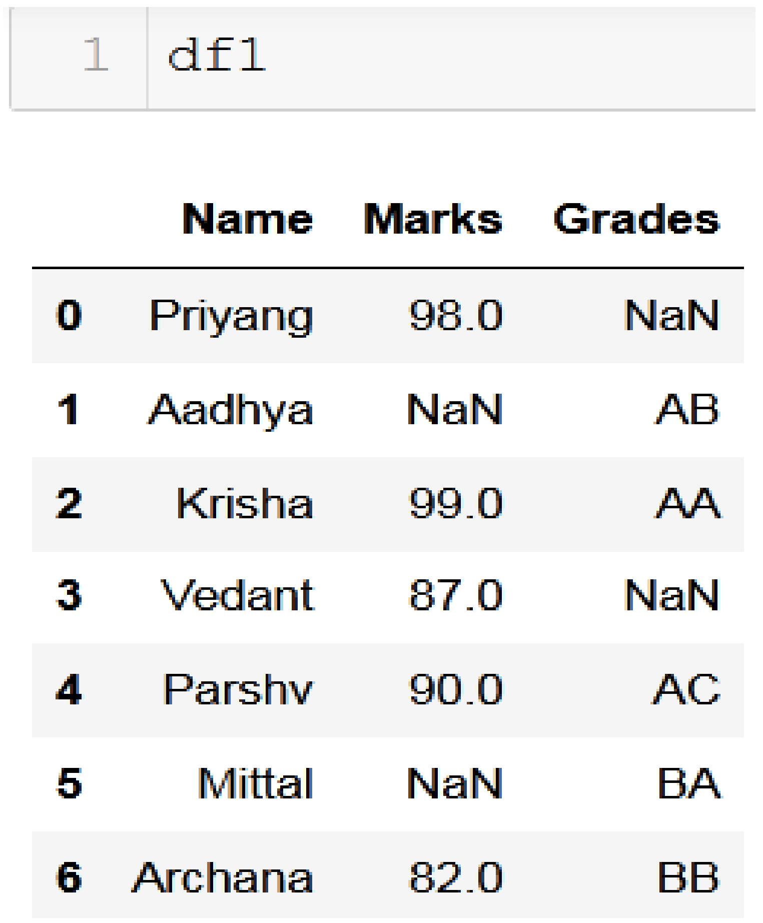
```
df1.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)
```

	Name	Marks	Grades
2	Krisha	99.0	AA
4	Parshv	90.0	AC
6	Archana	82.0	BB

```
axis : {0 or 'index', 1 or 'columns'}, default 0
---------------------------------------
0, or 'index' : Drop rows which contain missing values.
1, or 'columns' : Drop columns which contain missing value.
```

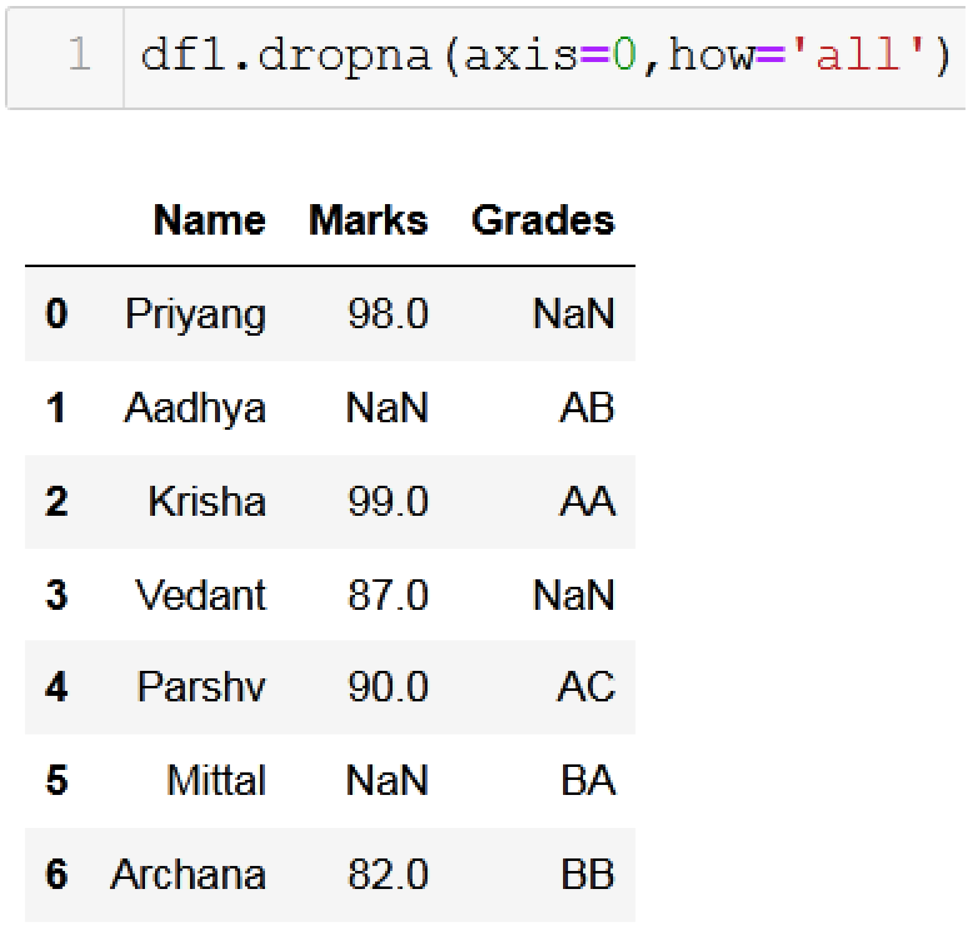
```
how : {'any', 'all'}, default 'any'
----------------------------------------
    Determine if row or column is removed from DataFrame, when we have
    at least one NA or all NA.

    * 'any' : If any NA values are present, drop that row or column.
    * 'all' : If all values are NA, drop that row or column.
```

Handling Missing Values : dropna()



```
df1
```

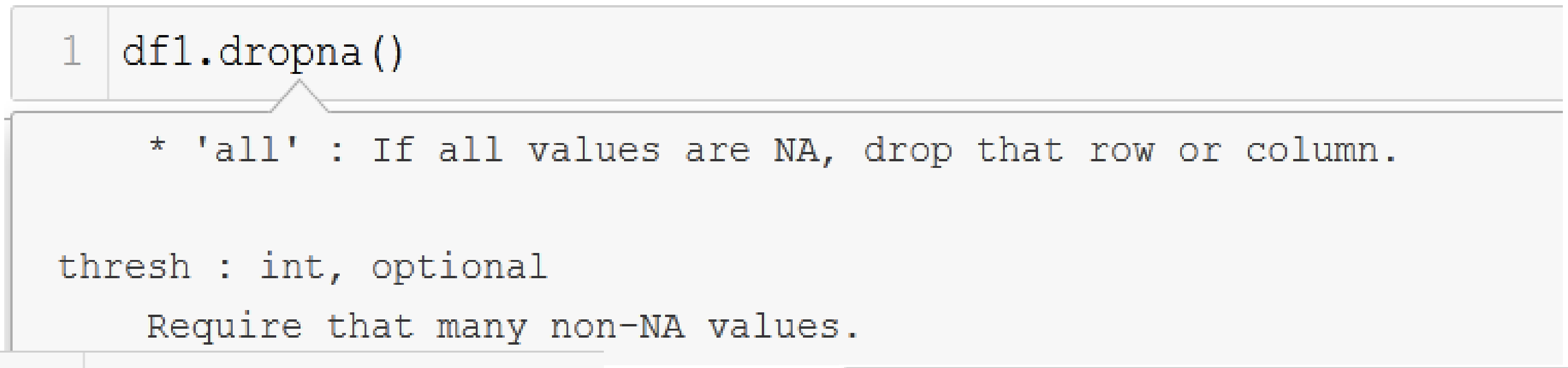
	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB



```
df1.dropna(axis=0,how='all')
```

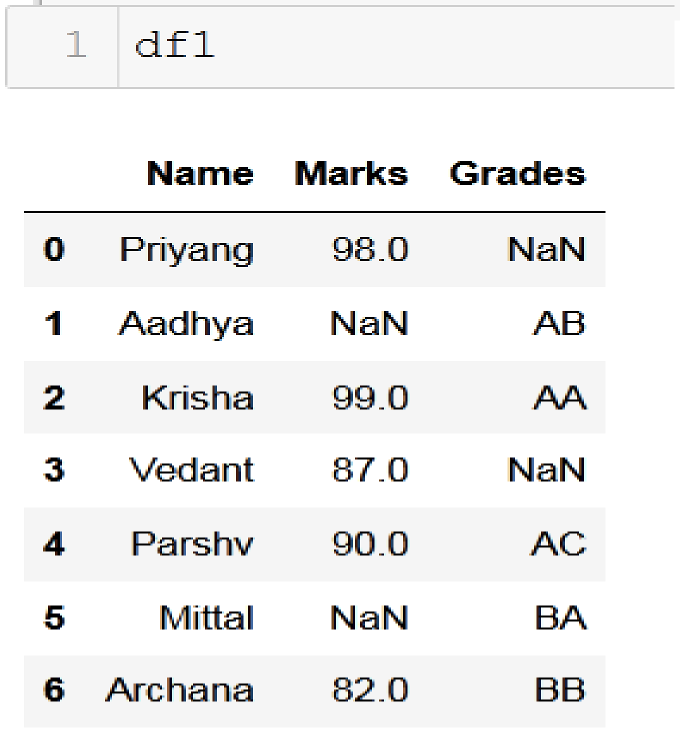
	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB

Handling Missing Values : dropna()



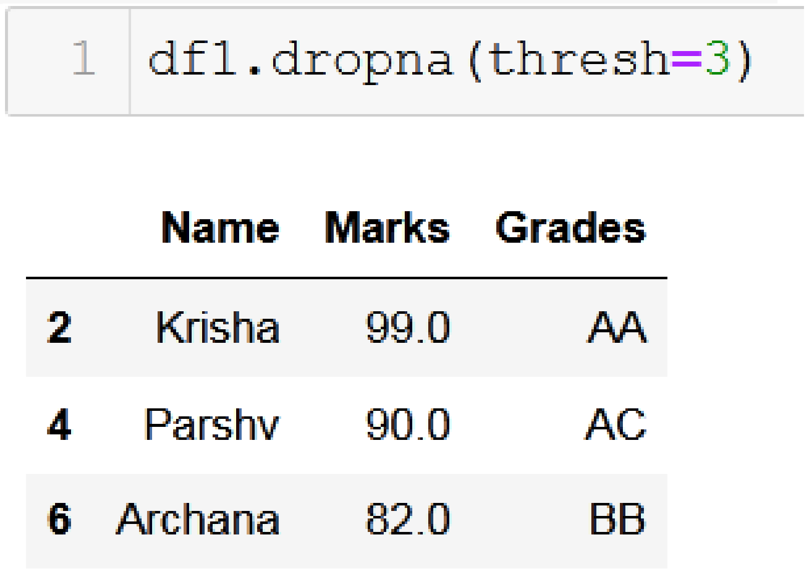
```
df1.dropna()
```

```
    * 'all' : If all values are NA, drop that row or column.

thresh : int, optional
    Require that many non-NA values.
```



```
df1
```

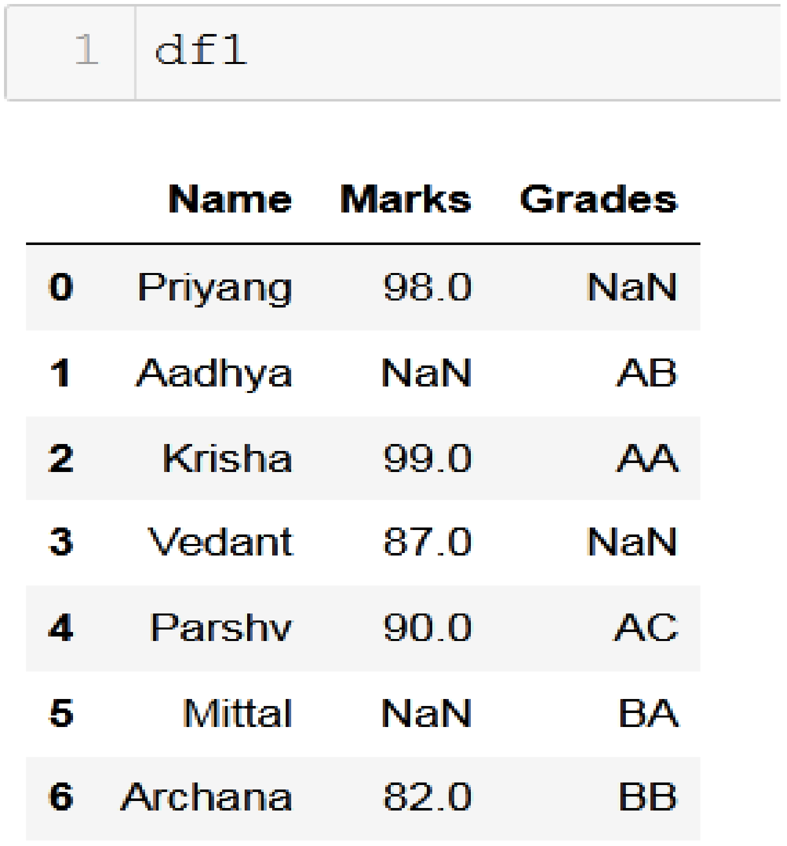
	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB



```
df1.dropna(thresh=3)
```

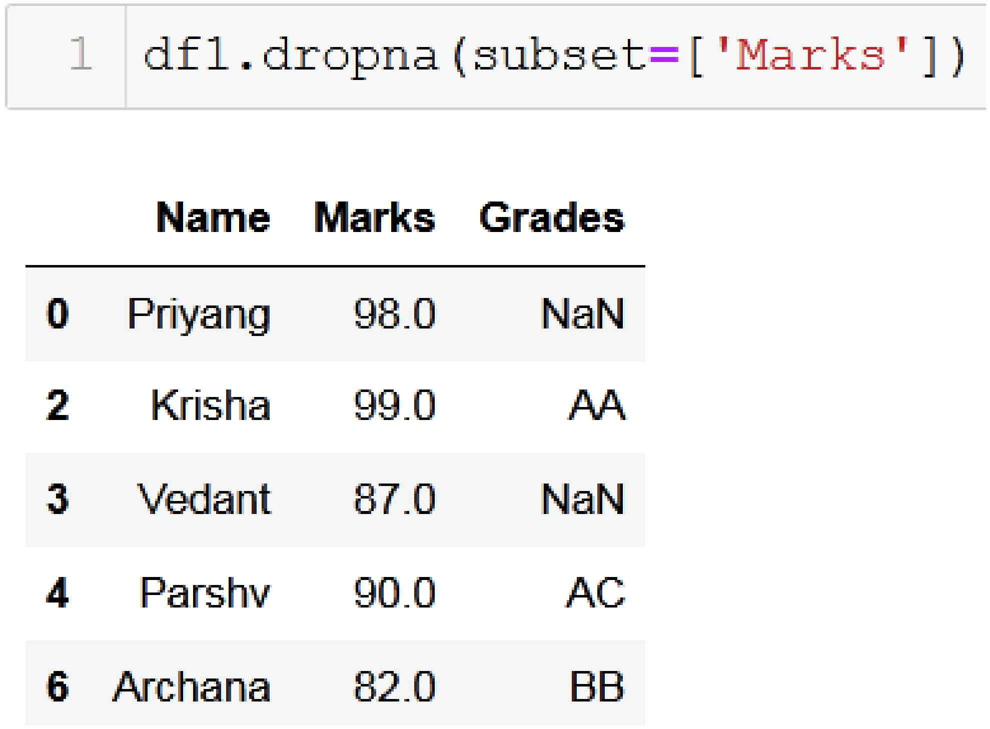
	Name	Marks	Grades
2	Krisha	99.0	AA
4	Parshv	90.0	AC
6	Archana	82.0	BB

Handling Missing Values : dropna()



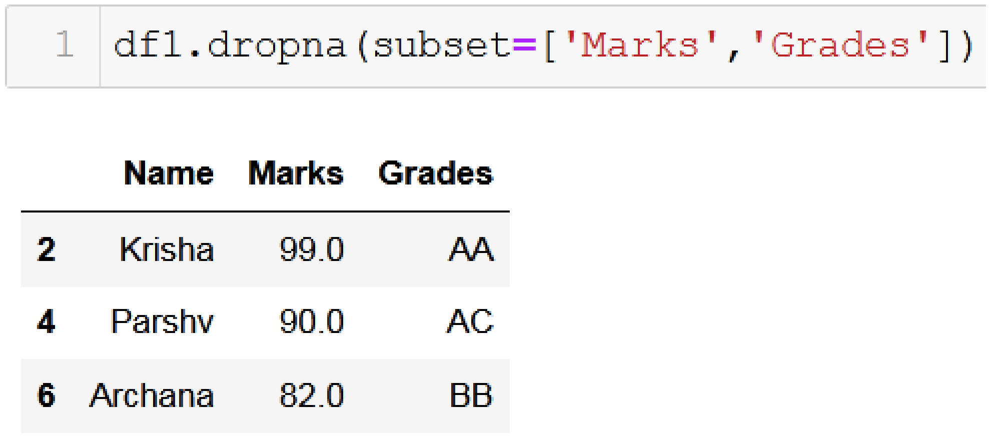
```python
df1
```

	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB



```python
df1.dropna(subset=['Marks'])
```

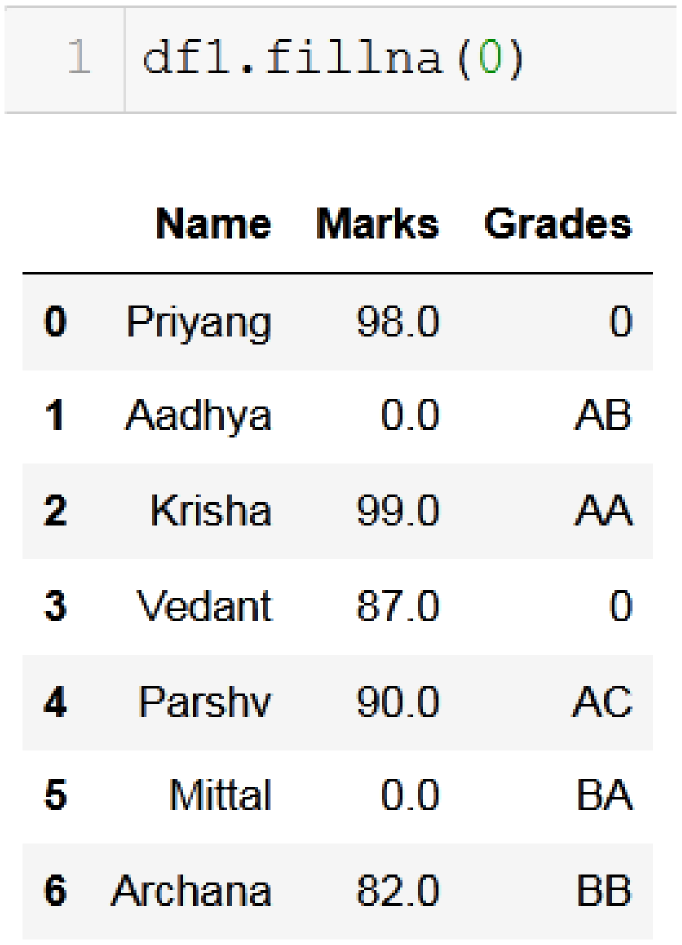
	Name	Marks	Grades
0	Priyang	98.0	NaN
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
6	Archana	82.0	BB



```python
df1.dropna(subset=['Marks','Grades'])
```

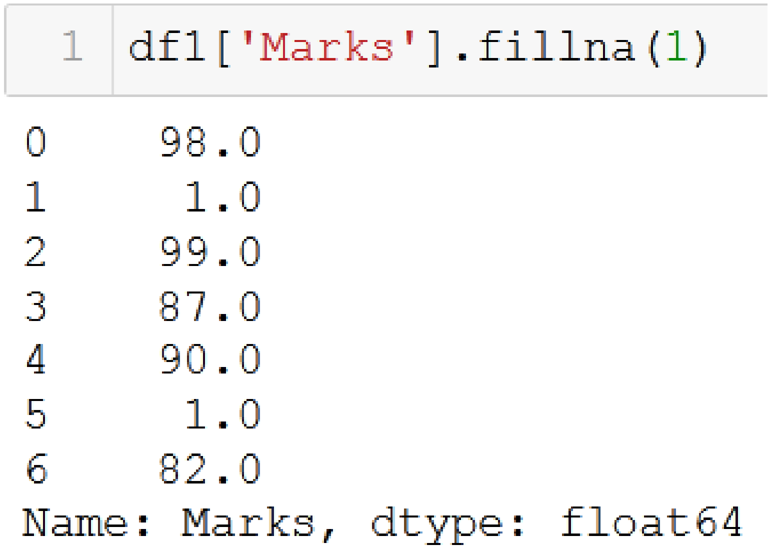
	Name	Marks	Grades
2	Krisha	99.0	AA
4	Parshv	90.0	AC
6	Archana	82.0	BB

Handling Missing Values : fillna()



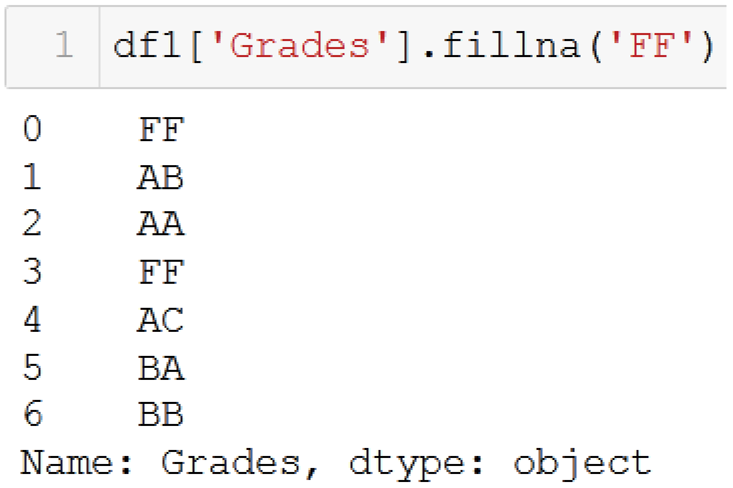
```
df1.fillna(0)
```

	Name	Marks	Grades
0	Priyang	98.0	0
1	Aadhya	0.0	AB
2	Krisha	99.0	AA
3	Vedant	87.0	0
4	Parshv	90.0	AC
5	Mittal	0.0	BA
6	Archana	82.0	BB



```
df1['Marks'].fillna(1)
```

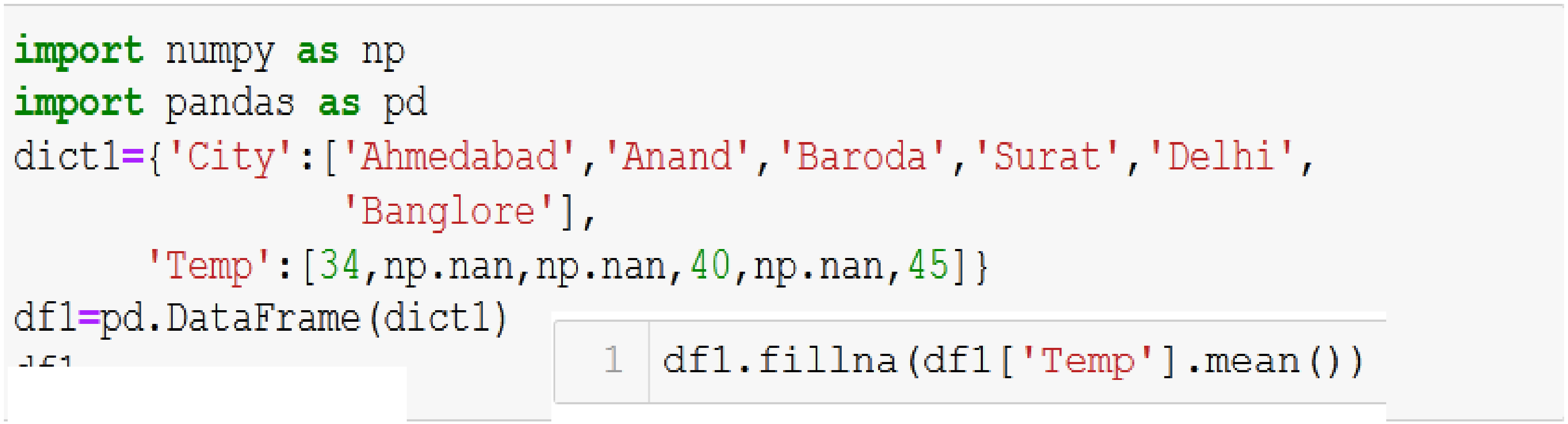
```
0    98.0
1     1.0
2    99.0
3    87.0
4    90.0
5     1.0
6    82.0
Name: Marks, dtype: float64
```



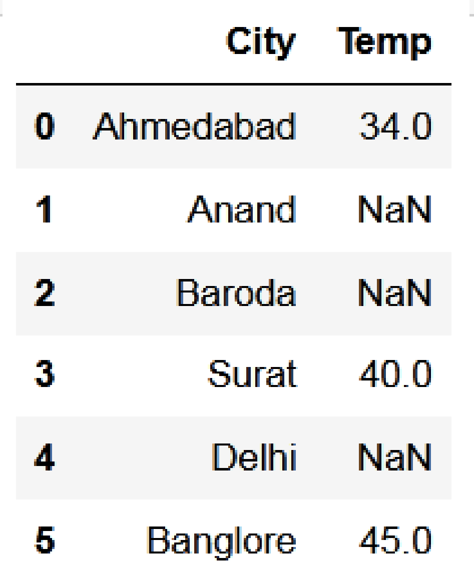
```
df1['Grades'].fillna('FF')
```

```
0    FF
1    AB
2    AA
3    FF
4    AC
5    BA
6    BB
Name: Grades, dtype: object
```

Handling Missing Values : fillna()

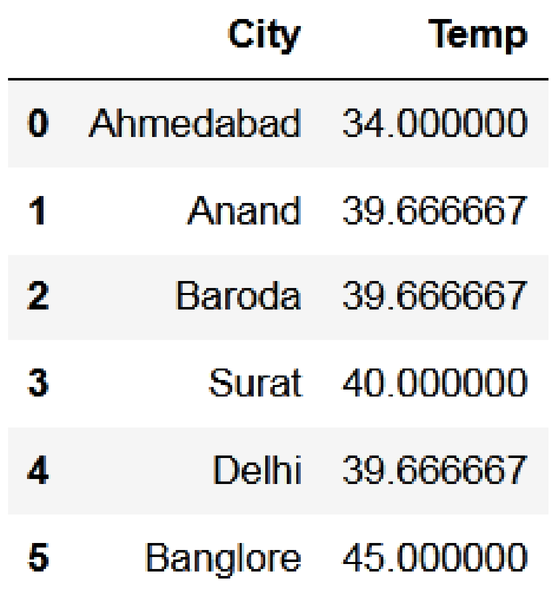


```python
import numpy as np
import pandas as pd
dict1={'City':['Ahmedabad','Anand','Baroda','Surat','Delhi',
               'Banglore'],
       'Temp':[34,np.nan,np.nan,40,np.nan,45]}
df1=pd.DataFrame(dict1)
df1
```



	City	Temp
0	Ahmedabad	34.0
1	Anand	NaN
2	Baroda	NaN
3	Surat	40.0
4	Delhi	NaN
5	Banglore	45.0

```python
df1.fillna(df1['Temp'].mean())
```

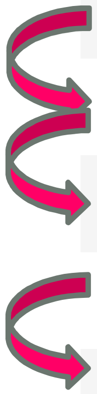


	City	Temp
0	Ahmedabad	34.000000
1	Anand	39.666667
2	Baroda	39.666667
3	Surat	40.000000
4	Delhi	39.666667
5	Banglore	45.000000

Handling Missing Values : fillna()

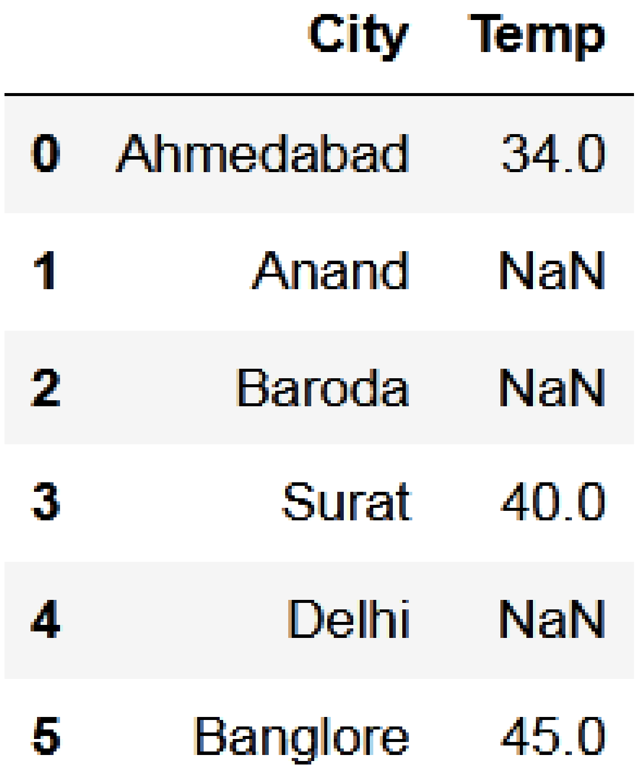
```
df1.fillna(method='ffill')
```

	City	Temp
0	Ahmedabad	34.0
1	Anand	NaN
2	Baroda	NaN
3	Surat	40.0
4	Delhi	NaN
5	Banglore	45.0

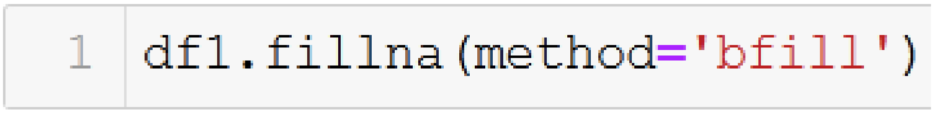


	City	Temp
0	Ahmedabad	34.0
1	Anand	34.0
2	Baroda	34.0
3	Surat	40.0
4	Delhi	40.0
5	Banglore	45.0

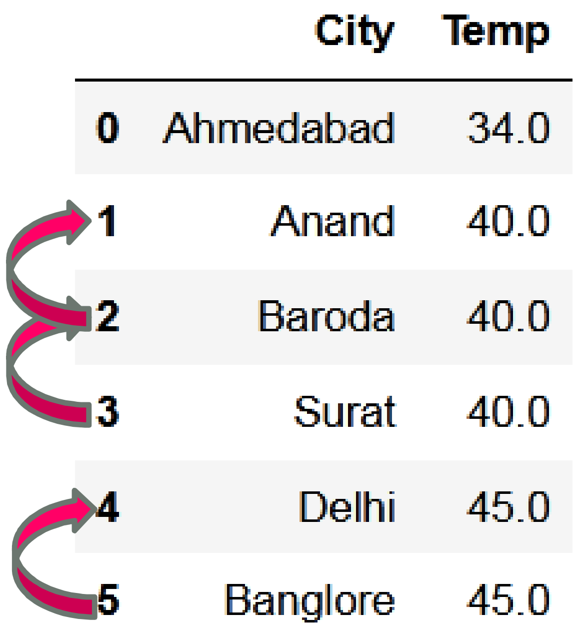
Handling Missing Values : fillna()



	City	Temp
0	Ahmedabad	34.0
1	Anand	NaN
2	Baroda	NaN
3	Surat	40.0
4	Delhi	NaN
5	Banglore	45.0



```
df1.fillna(method='bfill')
```



	City	Temp
0	Ahmedabad	34.0
1	Anand	40.0
2	Baroda	40.0
3	Surat	40.0
4	Delhi	45.0
5	Banglore	45.0

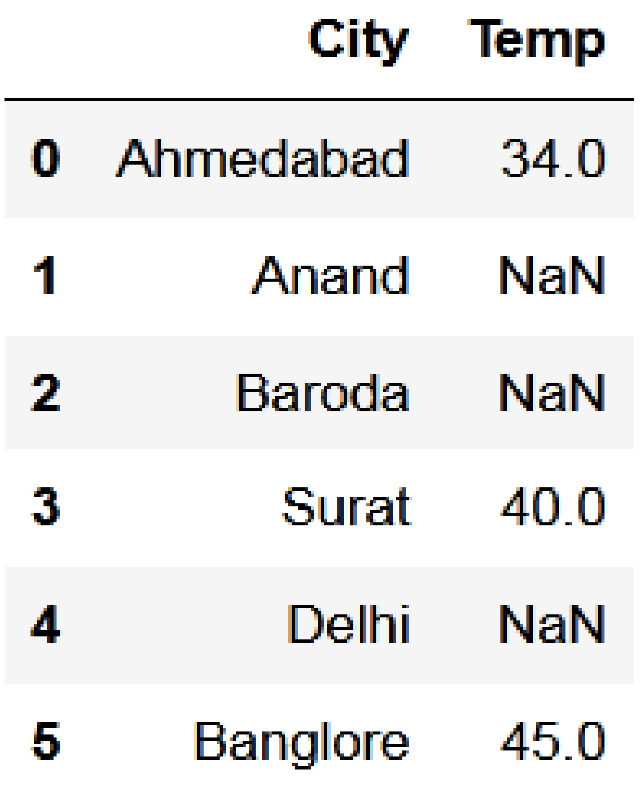
Handling Missing Values : fillna()

	City	Temp
0	Ahmedabad	34.0
1	Anand	NaN
2	Baroda	NaN
3	Surat	40.0
4	Delhi	NaN
5	Banglore	45.0

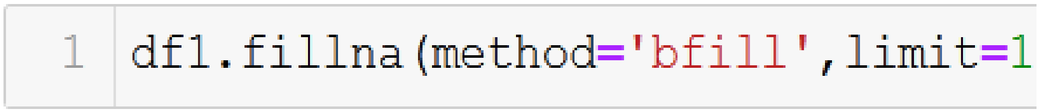
```
dfl.fillna(method='ffill',limit=1)
```

	City	Temp
0	Ahmedabad	34.0
1	Anand	34.0
2	Baroda	NaN
3	Surat	40.0
4	Delhi	40.0
5	Banglore	45.0

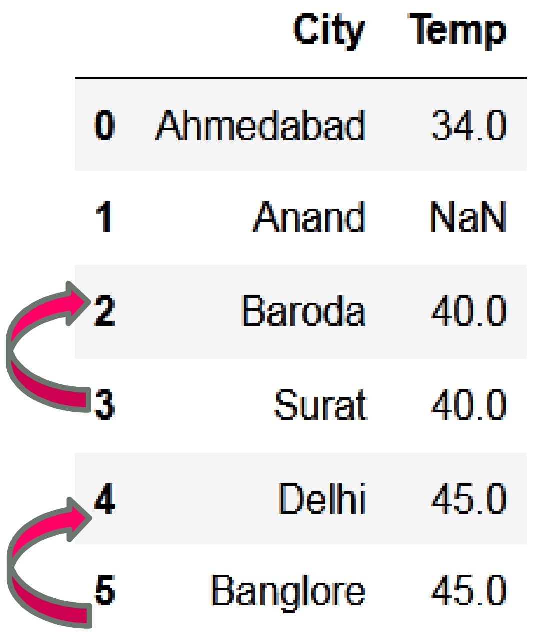
Handling Missing Values : fillna()



	City	Temp
0	Ahmedabad	34.0
1	Anand	NaN
2	Baroda	NaN
3	Surat	40.0
4	Delhi	NaN
5	Banglore	45.0

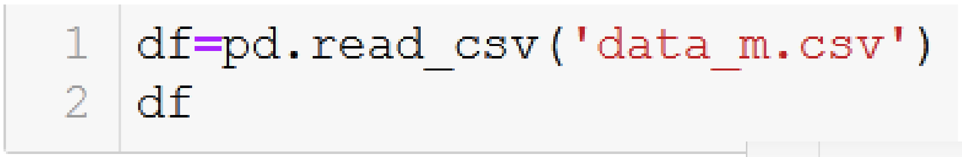


```
df1.fillna(method='bfill',limit=1
```

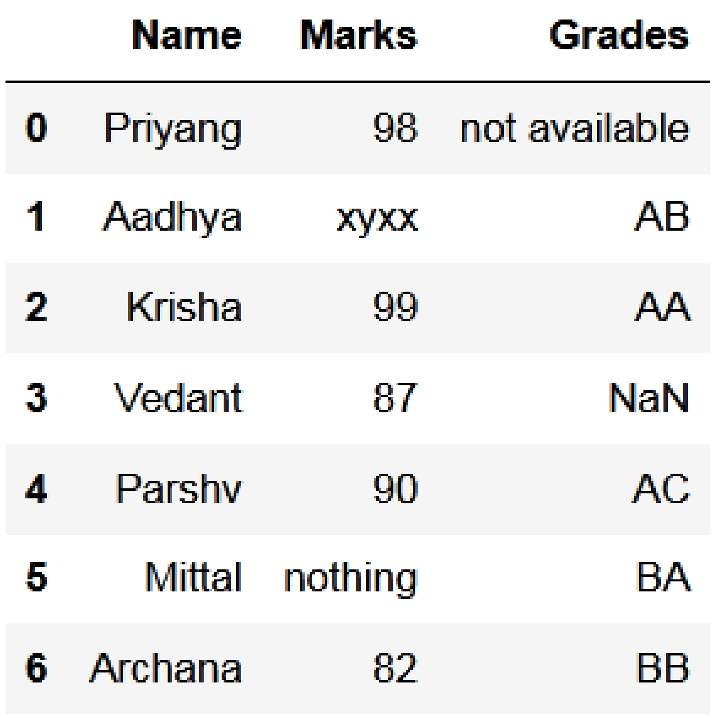


	City	Temp
0	Ahmedabad	34.0
1	Anand	NaN
2	Baroda	40.0
3	Surat	40.0
4	Delhi	45.0
5	Banglore	45.0

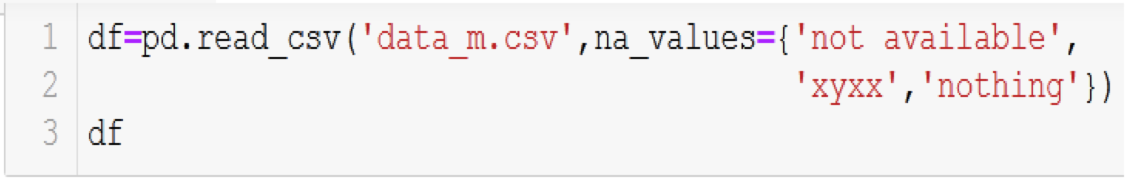
Handling Missing Values



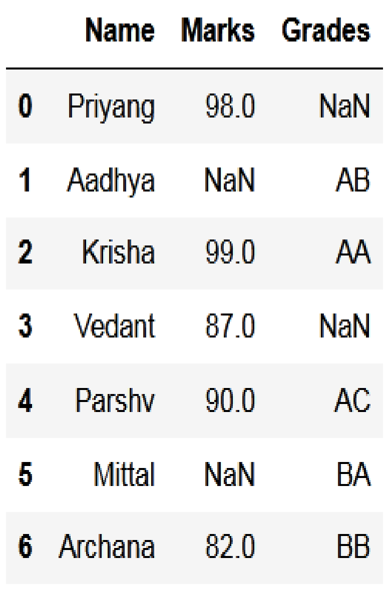
```
df=pd.read_csv('data_m.csv')
df
```



	Name	Marks	Grades
0	Priyang	98	not available
1	Aadhya	xyxx	AB
2	Krisha	99	AA
3	Vedant	87	NaN
4	Parshv	90	AC
5	Mittal	nothing	BA
6	Archana	82	BB

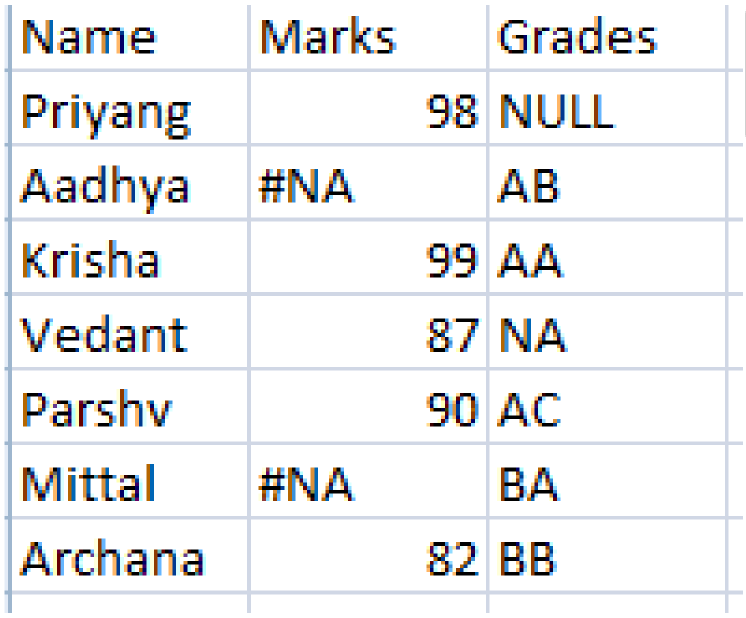


```
df=pd.read_csv('data_m.csv',na_values={'not available',
                                       'xyxx','nothing'})

df
```

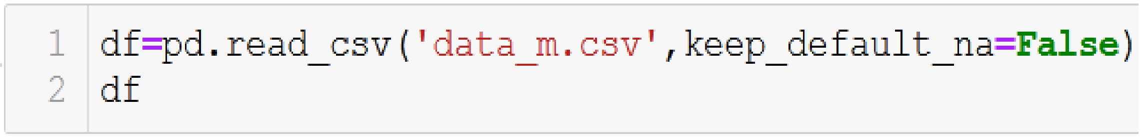


	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB

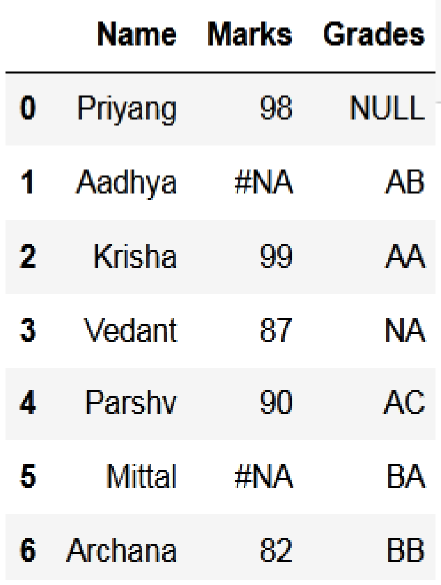
Handling Missing Values



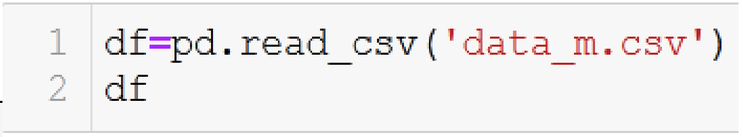
Name	Marks	Grades
Priyang	98	NULL
Aadhya	#NA	AB
Krisha	99	AA
Vedant	87	NA
Parshv	90	AC
Mittal	#NA	BA
Archana	82	BB



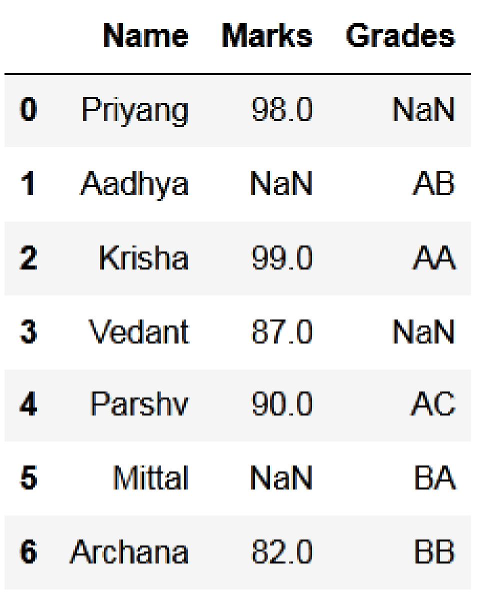
```python
df=pd.read_csv('data_m.csv',keep_default_na=False)
df
```



	Name	Marks	Grades
0	Priyang	98	NULL
1	Aadhya	#NA	AB
2	Krisha	99	AA
3	Vedant	87	NA
4	Parshv	90	AC
5	Mittal	#NA	BA
6	Archana	82	BB



```python
df=pd.read_csv('data_m.csv')
df
```



	Name	Marks	Grades
0	Priyang	98.0	NaN
1	Aadhya	NaN	AB
2	Krisha	99.0	AA
3	Vedant	87.0	NaN
4	Parshv	90.0	AC
5	Mittal	NaN	BA
6	Archana	82.0	BB

Inplace

if inplace is True :

It modifies an existing dataframe, Returns nothing.

else:

It returns a copy of dataframe object with the performed operation(s), Without modifying the existing dataframe.

Inplace

```
if inplace is True :
        df. SomeOperation(inplace=True)
else:
        df1=df.SomeOperation(inplace=False)
```