Numpy

Sahil Danayak

India in Digits

A Data visualization workshop by Coding Club

Coding Club
IIT Guwahati

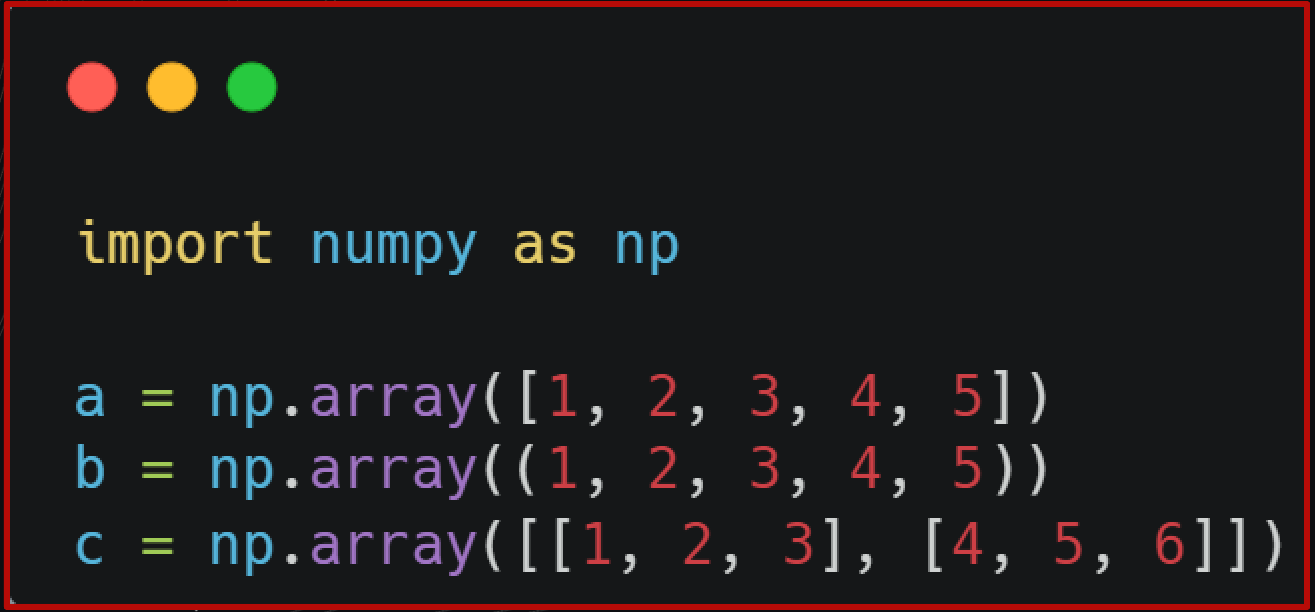
Numpy

- Written in C language
- Numpy arrays provide more efficient data storage and operations compared list
- Allows to manage vectors, matrices and higher dimensional arrays
- Used in Scientific computing, Deep learning and financial analysis

Installation

```
pip install numpy
```

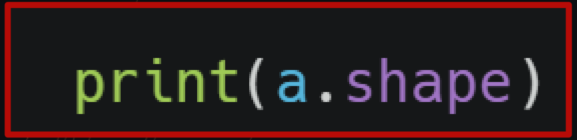
Numpy Arrays



```python
import numpy as np

a = np.array([1, 2, 3, 4, 5])
b = np.array((1, 2, 3, 4, 5))
c = np.array([[1, 2, 3], [4, 5, 6]])
```

Shape of Array

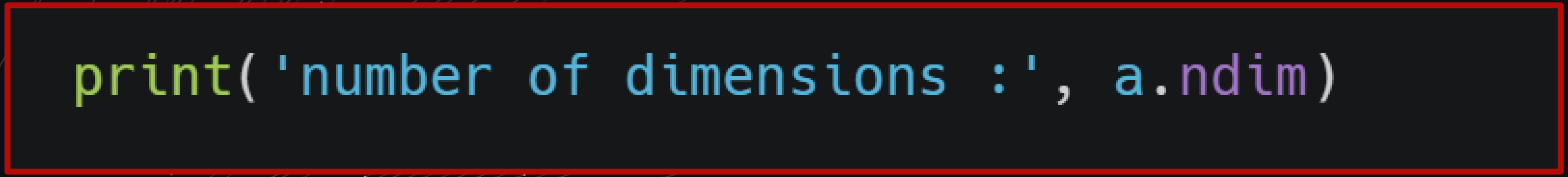
Returns a tuple with each index having the number of corresponding elements



```python
print(a.shape)
```

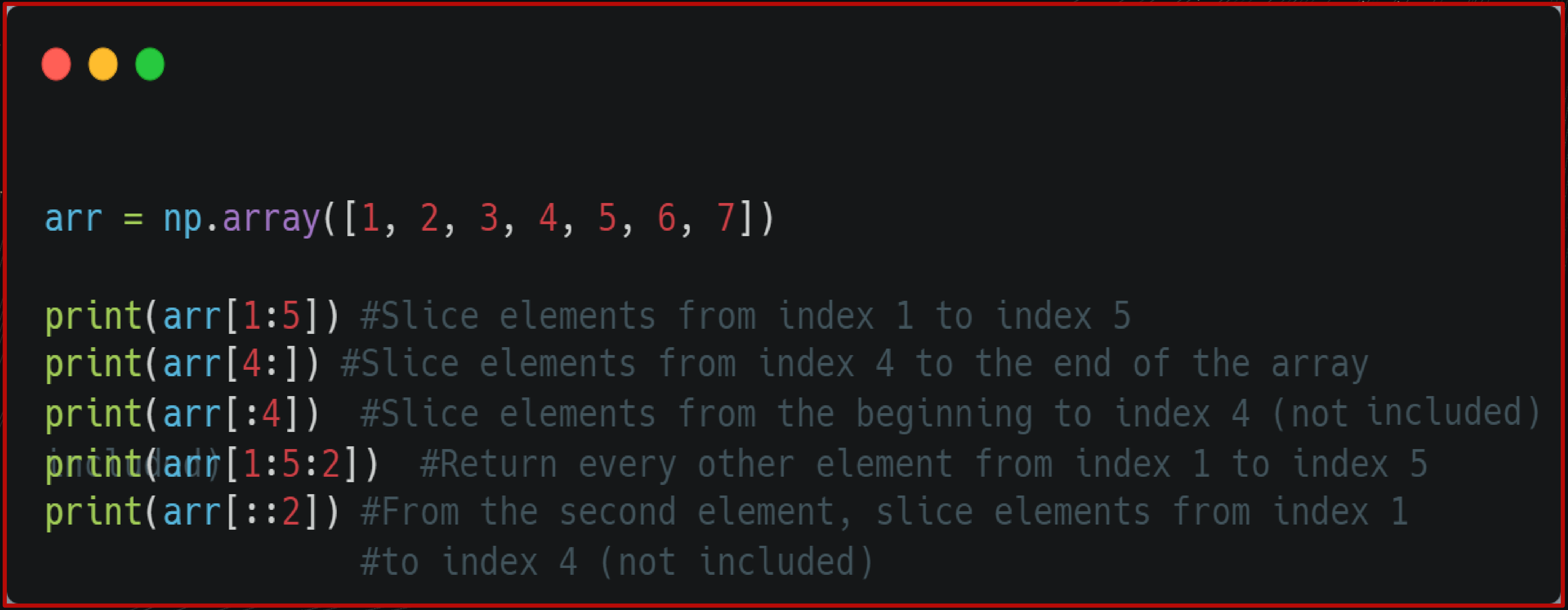
Rank of Array

Returns a integer representing dimension of array



```python
print('number of dimensions :', a.ndim)
```

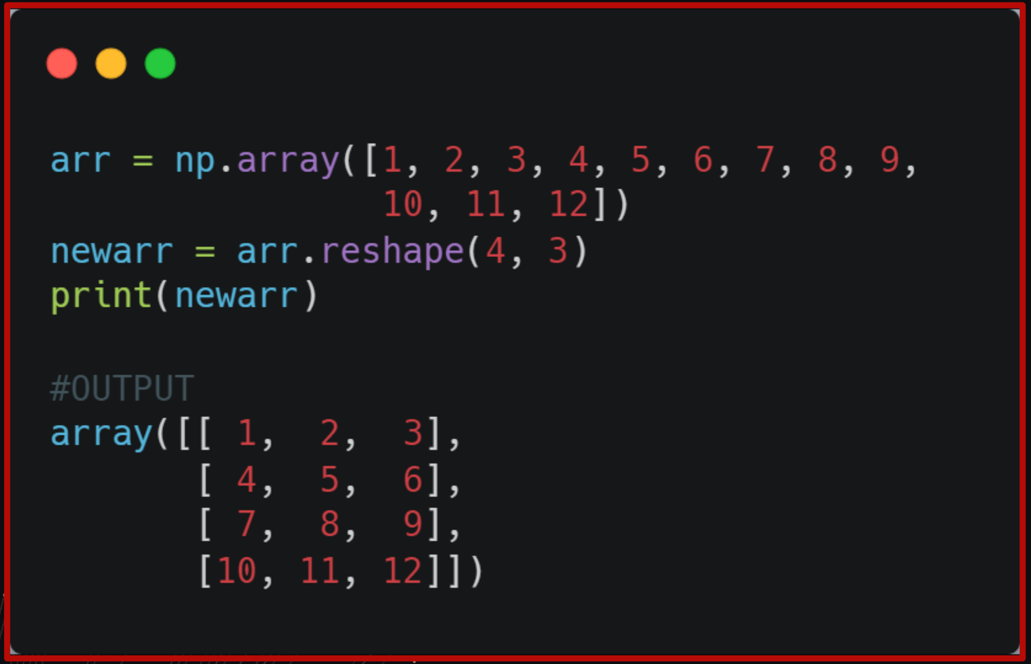
Array slicing



```
arr = np.array([1, 2, 3, 4, 5, 6, 7])

print(arr[1:5]) #Slice elements from index 1 to index 5
print(arr[4:]) #Slice elements from index 4 to the end of the array
print(arr[:4])  #Slice elements from the beginning to index 4 (not included)
print(arr[1:5:2])  #Return every other element from index 1 to index 5
print(arr[::2]) #From the second element, slice elements from index 1
                #to index 4 (not included)
```

Reshaping Array

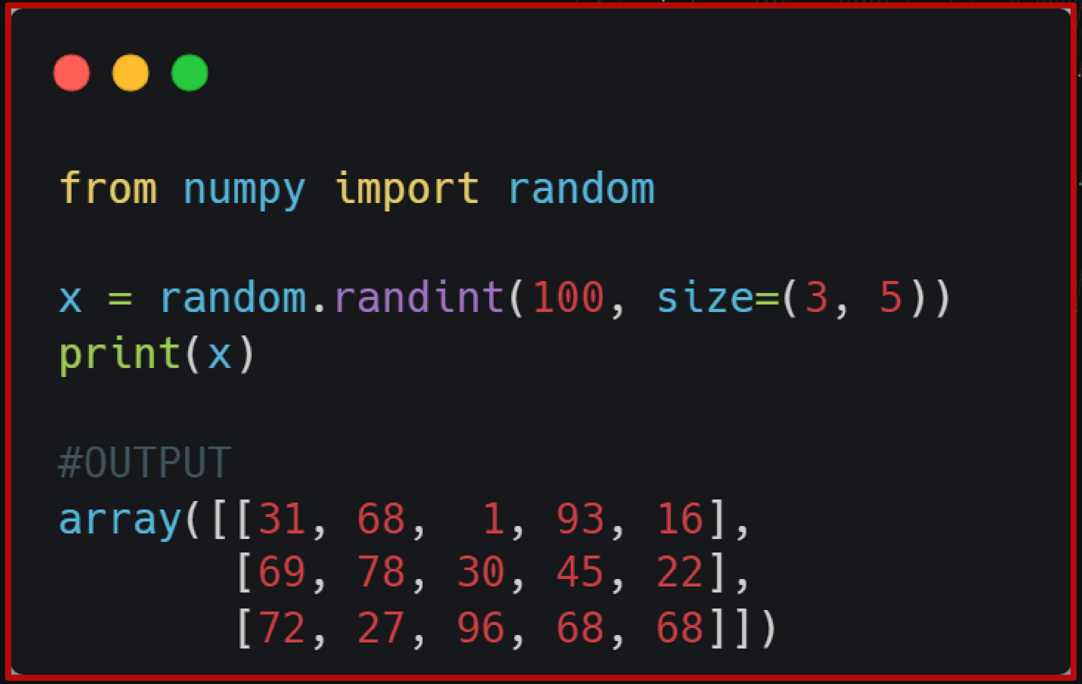
The outermost dimension will have 4 arrays, each with 3 elements.



```
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9,
                10, 11, 12])
newarr = arr.reshape(4, 3)
print(newarr)

#OUTPUT
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])
```

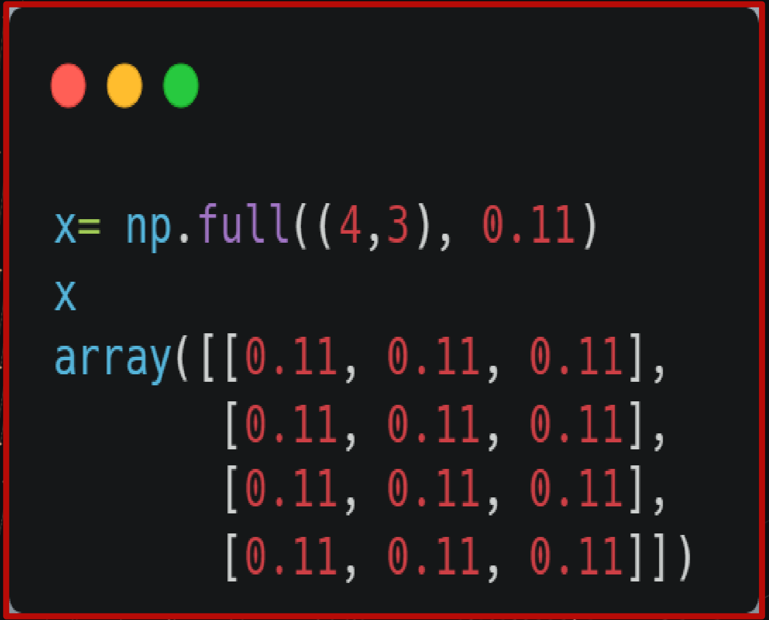
Random. Randint

The array is completely filled with elements between 0 and 100.



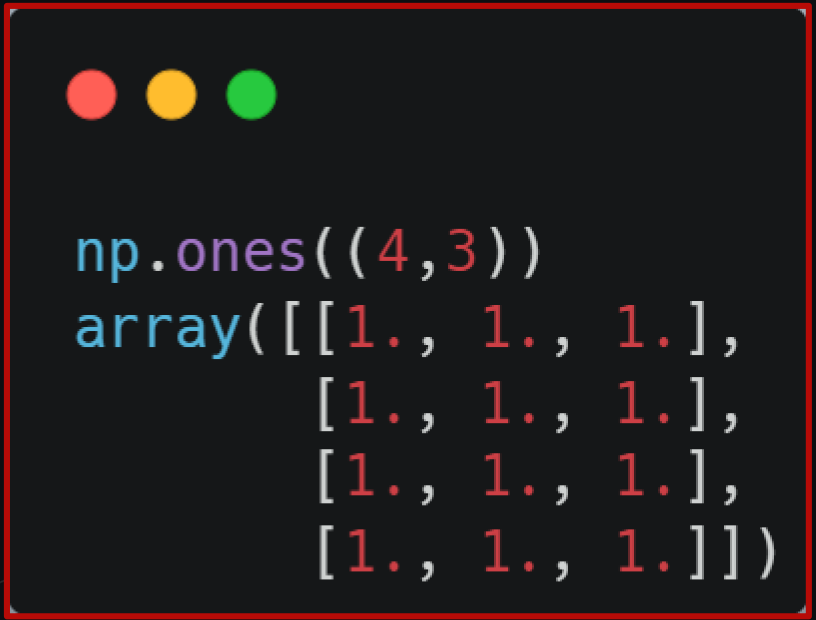
```
from numpy import random

x = random.randint(100, size=(3, 5))
print(x)

#OUTPUT
array([[31, 68,  1, 93, 16],
       [69, 78, 30, 45, 22],
       [72, 27, 96, 68, 68]])
```

Other Numpy Operations



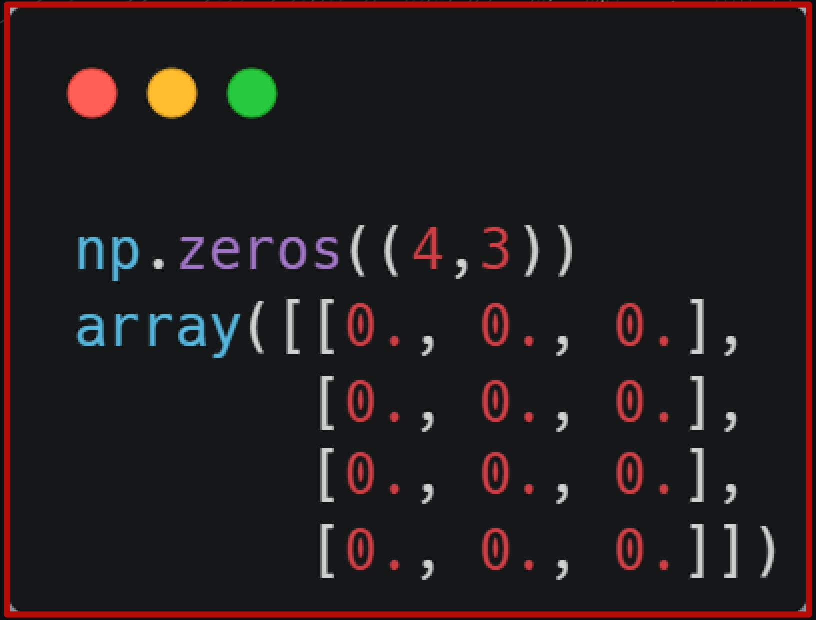
```
x= np.full((4,3), 0.11)
x
array([[0.11, 0.11, 0.11],
       [0.11, 0.11, 0.11],
       [0.11, 0.11, 0.11],
       [0.11, 0.11, 0.11]])
```

Creates array with all elements as 0.11



```
np.ones((4,3))
array([[1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.],
       [1., 1., 1.]])
```

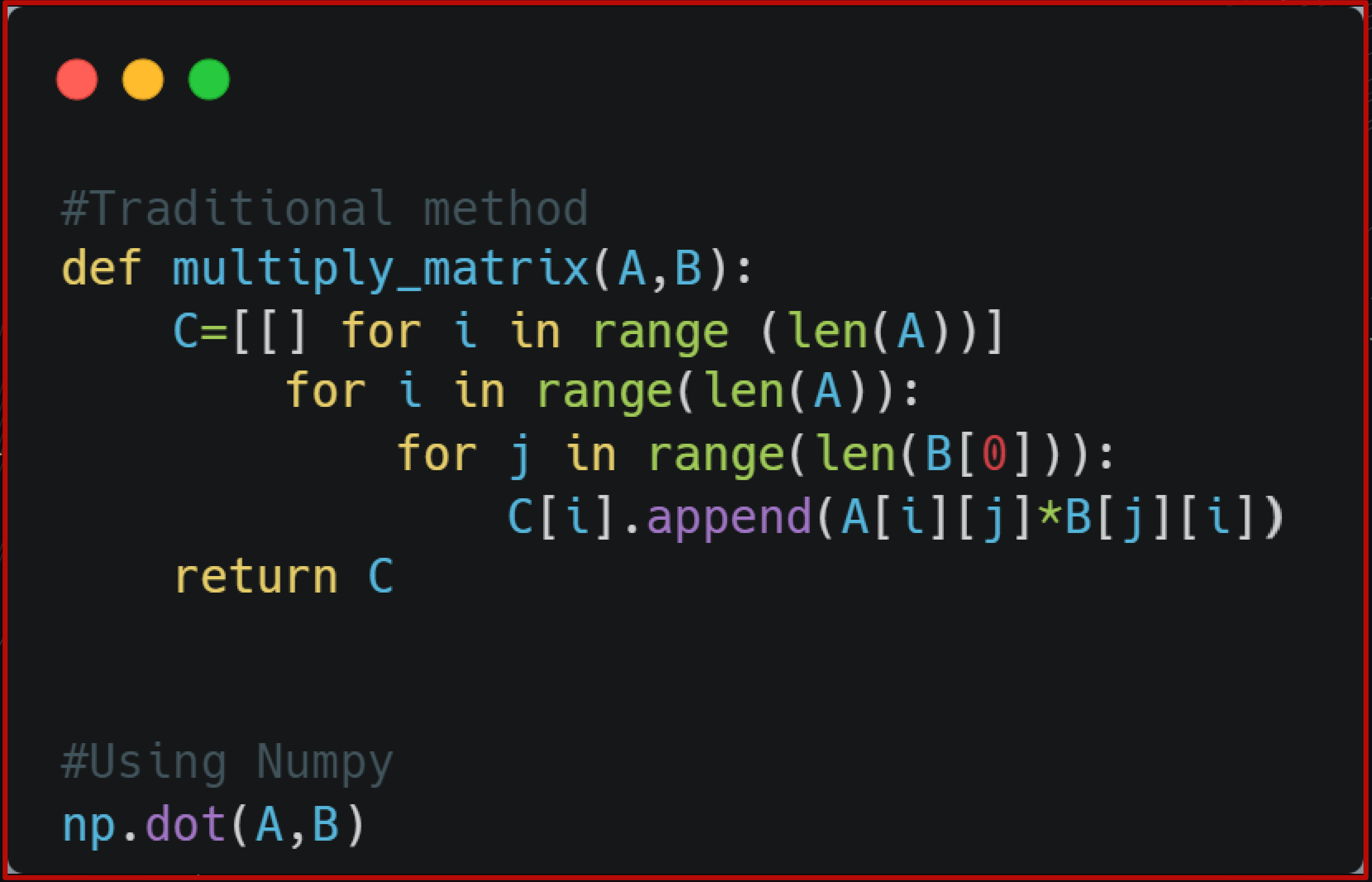
Creates array with all elements as 1



```
np.zeros((4,3))
array([[0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.],
       [0., 0., 0.]])
```

Creates array with all elements as 0

Dot Product



```
#Traditional method
def multiply_matrix(A,B):
    C=[[] for i in range (len(A))]
        for i in range(len(A)):
            for j in range(len(B[0])):
                C[i].append(A[i][j]*B[j][i])
    return C


#Using Numpy
np.dot(A,B)
```

India in Digits

A Data visualization
workshop by Coding Club

Pandas

Sahil Danayak

Coding Club
IIT Guwahati

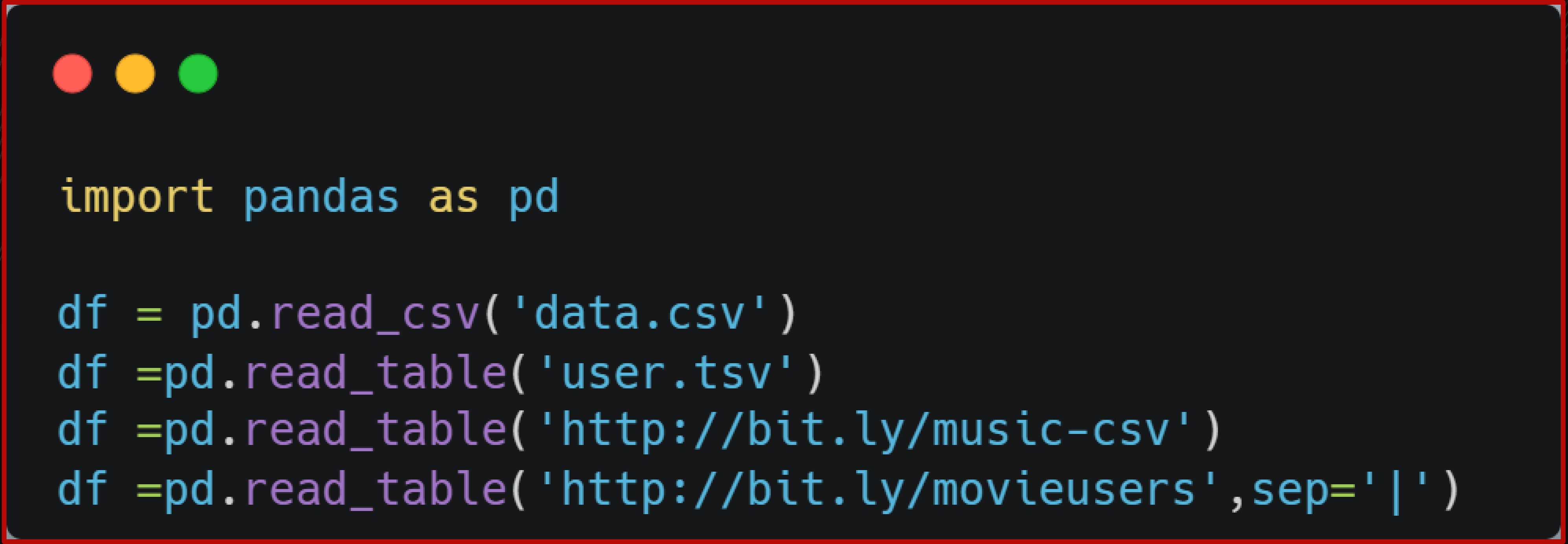
Pandas

- Pandas is a Python library used for working with data sets.
- It has functions for analyzing, cleaning, exploring, and manipulating data.
- The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Installation

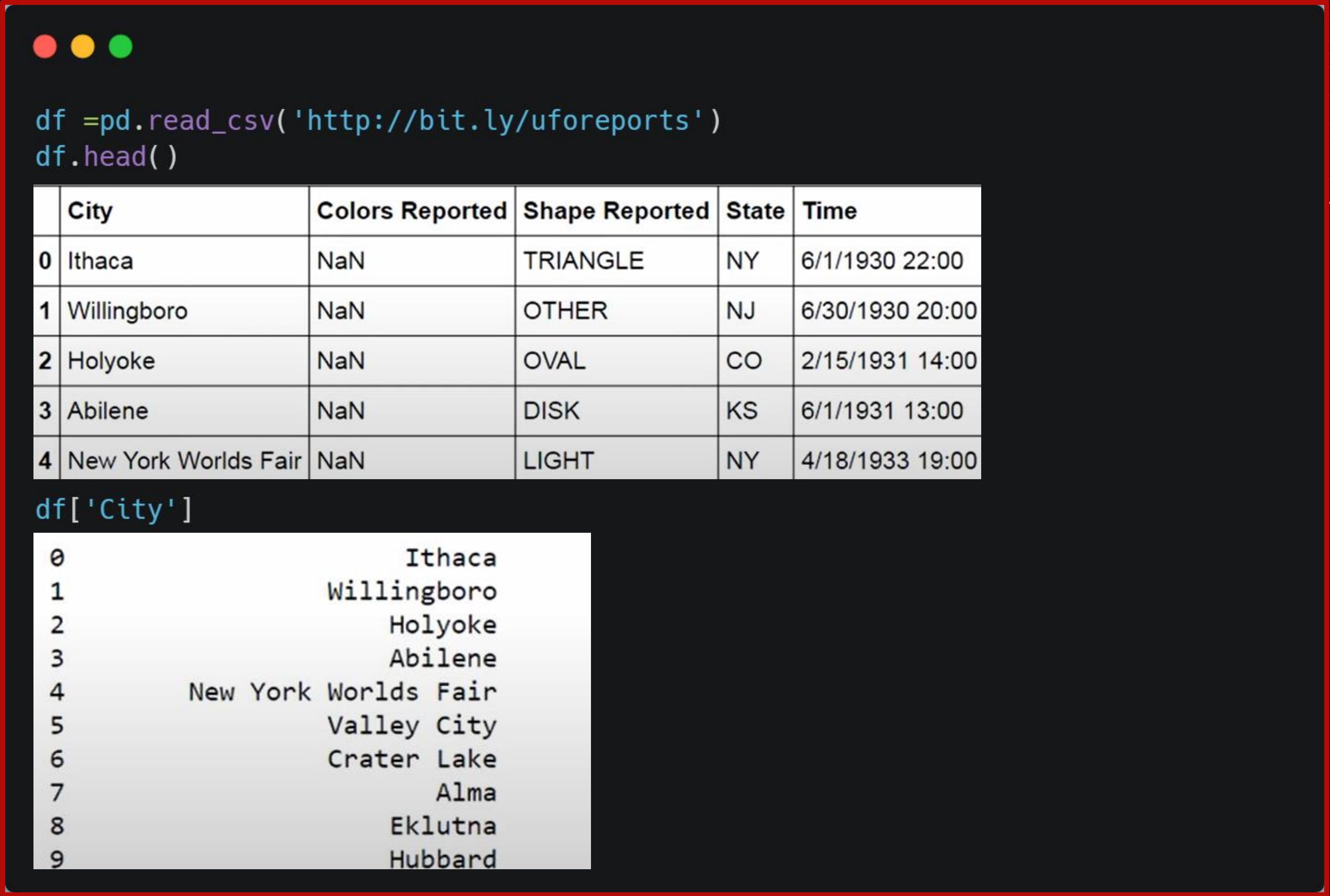
```
pip install pandas
```

Database Creation



```
import pandas as pd

df = pd.read_csv('data.csv')
df =pd.read_table('user.tsv')
df =pd.read_table('http://bit.ly/music-csv')
df =pd.read_table('http://bit.ly/movieusers',sep='|')
```

Printing Values from Dataframe



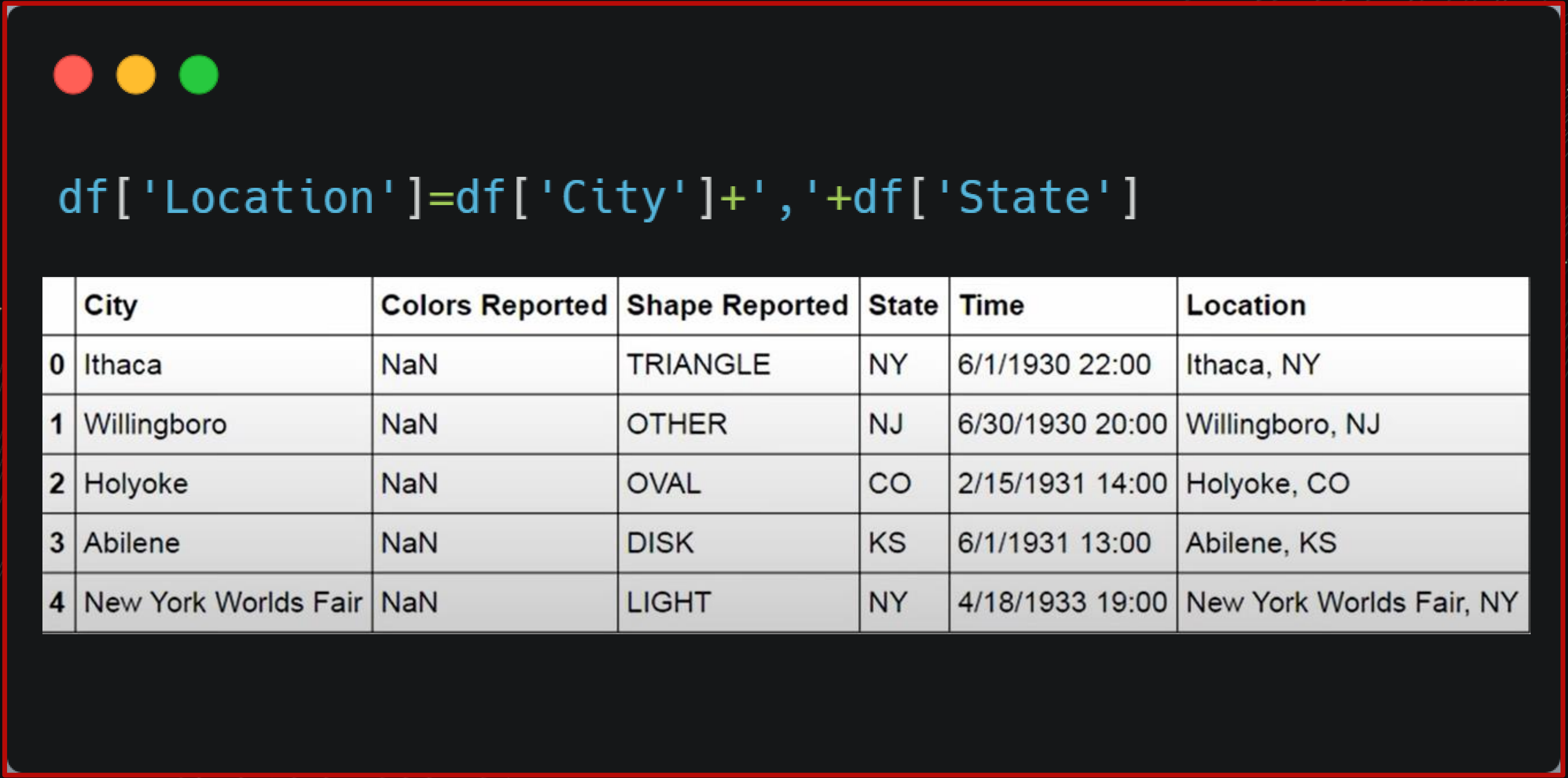
```
df =pd.read_csv('http://bit.ly/uforeports')
df.head()
```

	City	Colors Reported	Shape Reported	State	Time
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00
3	Abilene	NaN	DISK	KS	6/1/1931 13:00
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00

```
df['City']
```

```
0                   Ithaca
1              Willingboro
2                  Holyoke
3                  Abilene
4     New York Worlds Fair
5              Valley City
6              Crater Lake
7                     Alma
8                  Eklutna
9                  Hubbard
```

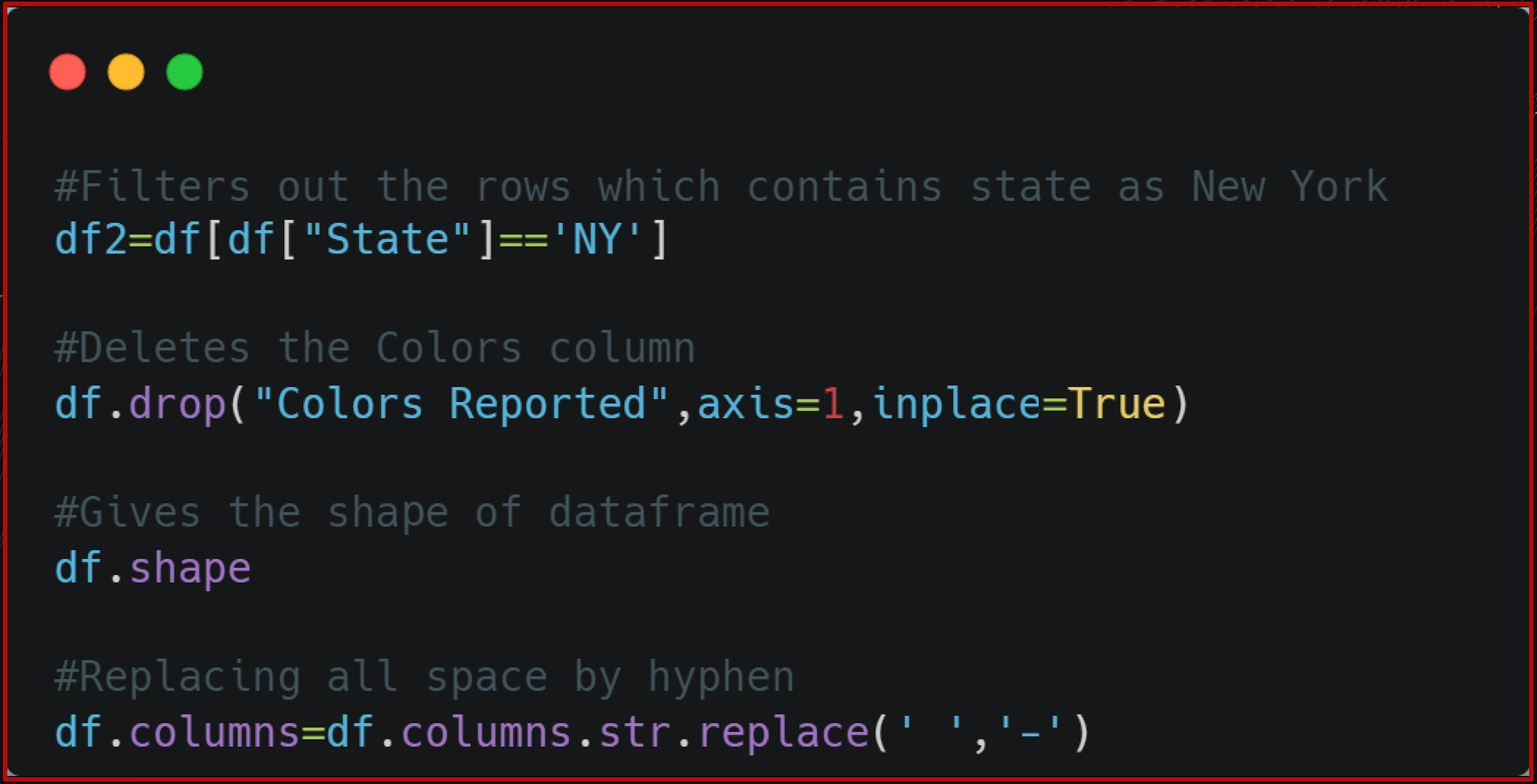
Creating New Fields



```
df['Location']=df['City']+','+df['State']
```

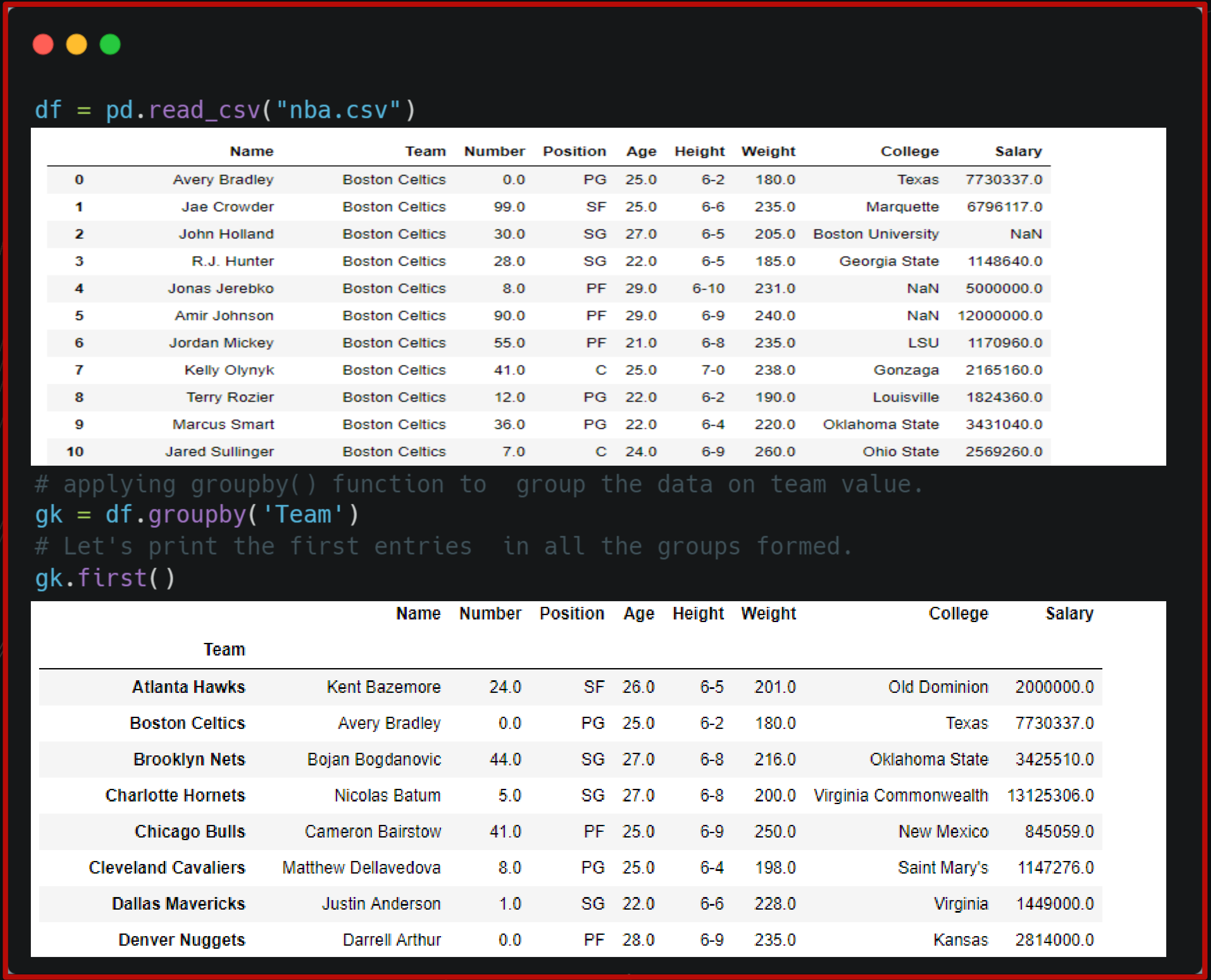
	City	Colors Reported	Shape Reported	State	Time	Location
0	Ithaca	NaN	TRIANGLE	NY	6/1/1930 22:00	Ithaca, NY
1	Willingboro	NaN	OTHER	NJ	6/30/1930 20:00	Willingboro, NJ
2	Holyoke	NaN	OVAL	CO	2/15/1931 14:00	Holyoke, CO
3	Abilene	NaN	DISK	KS	6/1/1931 13:00	Abilene, KS
4	New York Worlds Fair	NaN	LIGHT	NY	4/18/1933 19:00	New York Worlds Fair, NY

Code



```
#Filters out the rows which contains state as New York
df2=df[df["State"]=='NY']

#Deletes the Colors column
df.drop("Colors Reported",axis=1,inplace=True)

#Gives the shape of dataframe
df.shape

#Replacing all space by hyphen
df.columns=df.columns.str.replace(' ','-')
```

Code



```python
df = pd.read_csv("nba.csv")
```

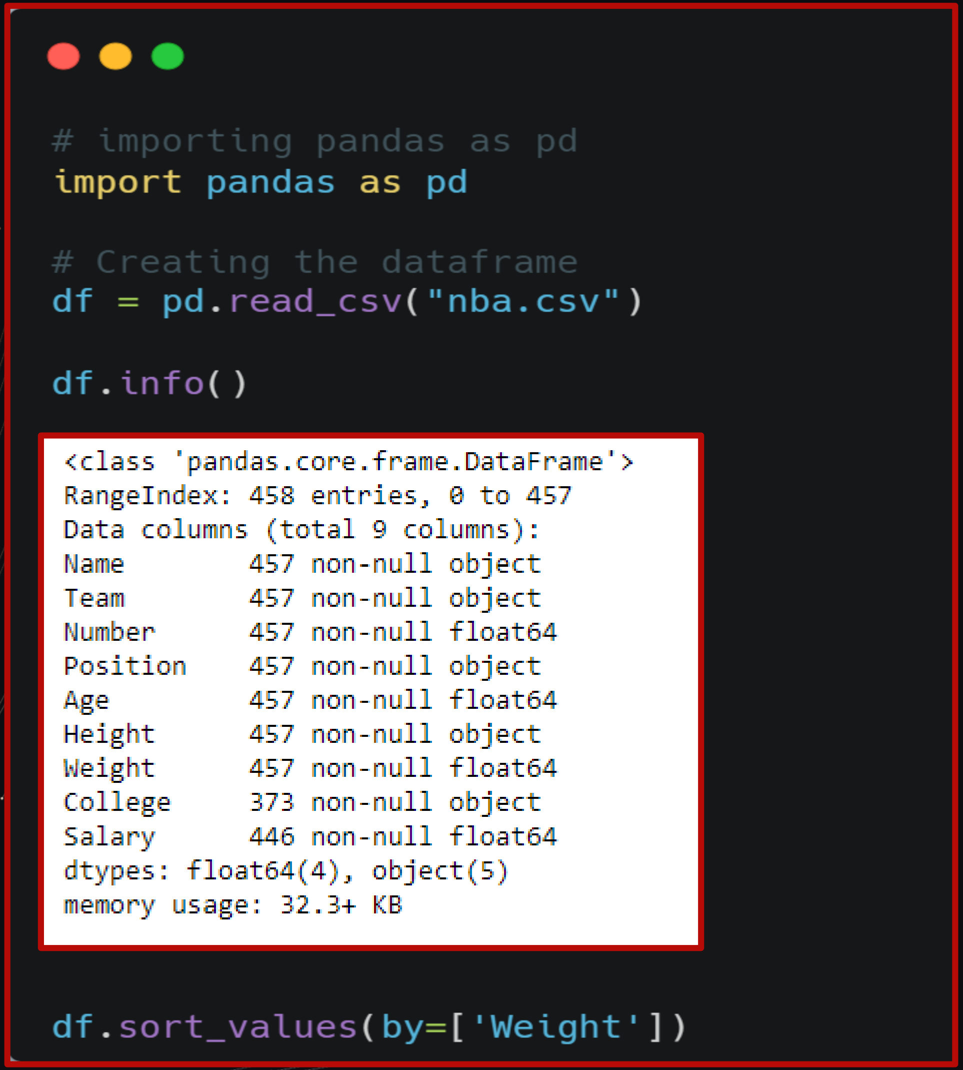
	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99.0	SF	25.0	6-6	235.0	Marquette	6796117.0
2	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28.0	SG	22.0	6-5	185.0	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
5	Amir Johnson	Boston Celtics	90.0	PF	29.0	6-9	240.0	NaN	12000000.0
6	Jordan Mickey	Boston Celtics	55.0	PF	21.0	6-8	235.0	LSU	1170960.0
7	Kelly Olynyk	Boston Celtics	41.0	C	25.0	7-0	238.0	Gonzaga	2165160.0
8	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
9	Marcus Smart	Boston Celtics	36.0	PG	22.0	6-4	220.0	Oklahoma State	3431040.0
10	Jared Sullinger	Boston Celtics	7.0	C	24.0	6-9	260.0	Ohio State	2569260.0

```python
# applying groupby() function to  group the data on team value.
gk = df.groupby('Team')
# Let's print the first entries  in all the groups formed.
gk.first()
```

	Name	Number	Position	Age	Height	Weight	College	Salary
Team								
Atlanta Hawks	Kent Bazemore	24.0	SF	26.0	6-5	201.0	Old Dominion	2000000.0
Boston Celtics	Avery Bradley	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
Brooklyn Nets	Bojan Bogdanovic	44.0	SG	27.0	6-8	216.0	Oklahoma State	3425510.0
Charlotte Hornets	Nicolas Batum	5.0	SG	27.0	6-8	200.0	Virginia Commonwealth	13125306.0
Chicago Bulls	Cameron Bairstow	41.0	PF	25.0	6-9	250.0	New Mexico	845059.0
Cleveland Cavaliers	Matthew Dellavedova	8.0	PG	25.0	6-4	198.0	Saint Mary's	1147276.0
Dallas Mavericks	Justin Anderson	1.0	SG	22.0	6-6	228.0	Virginia	1449000.0
Denver Nuggets	Darrell Arthur	0.0	PF	28.0	6-9	235.0	Kansas	2814000.0

Scan this QR Code to download nba.csv

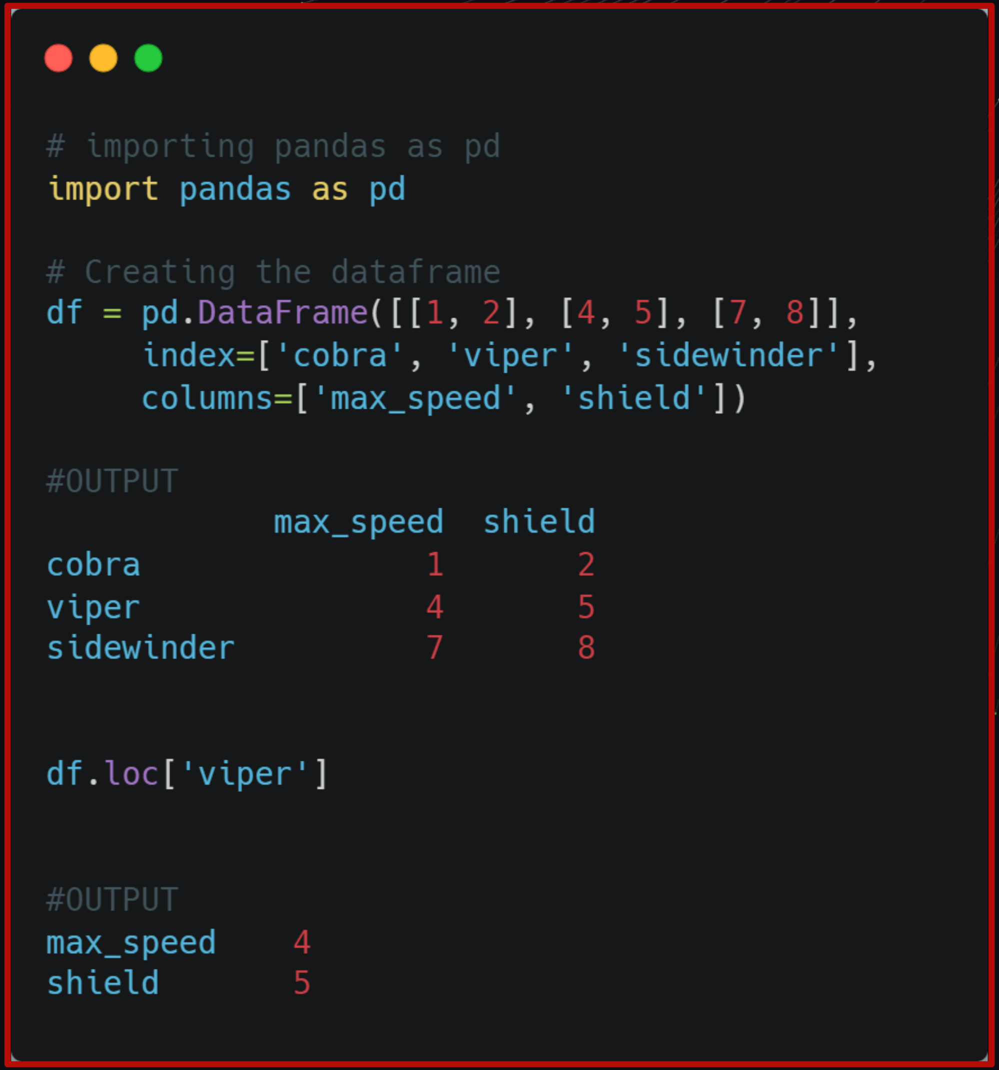
Code



```
# importing pandas as pd
import pandas as pd

# Creating the dataframe
df = pd.read_csv("nba.csv")

df.info()
```

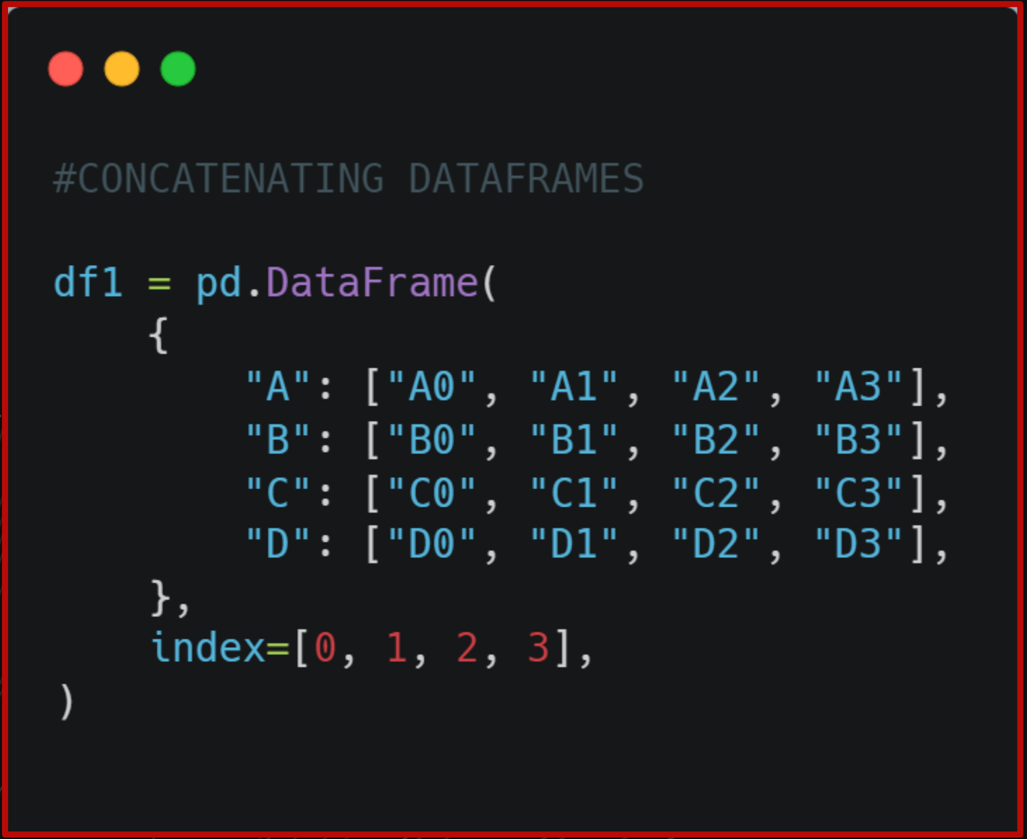
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 458 entries, 0 to 457
Data columns (total 9 columns):
Name        457 non-null object
Team        457 non-null object
Number      457 non-null float64
Position    457 non-null object
Age         457 non-null float64
Height      457 non-null object
Weight      457 non-null float64
College     373 non-null object
Salary      446 non-null float64
dtypes: float64(4), object(5)
memory usage: 32.3+ KB
```

```
df.sort_values(by=['Weight'])
```

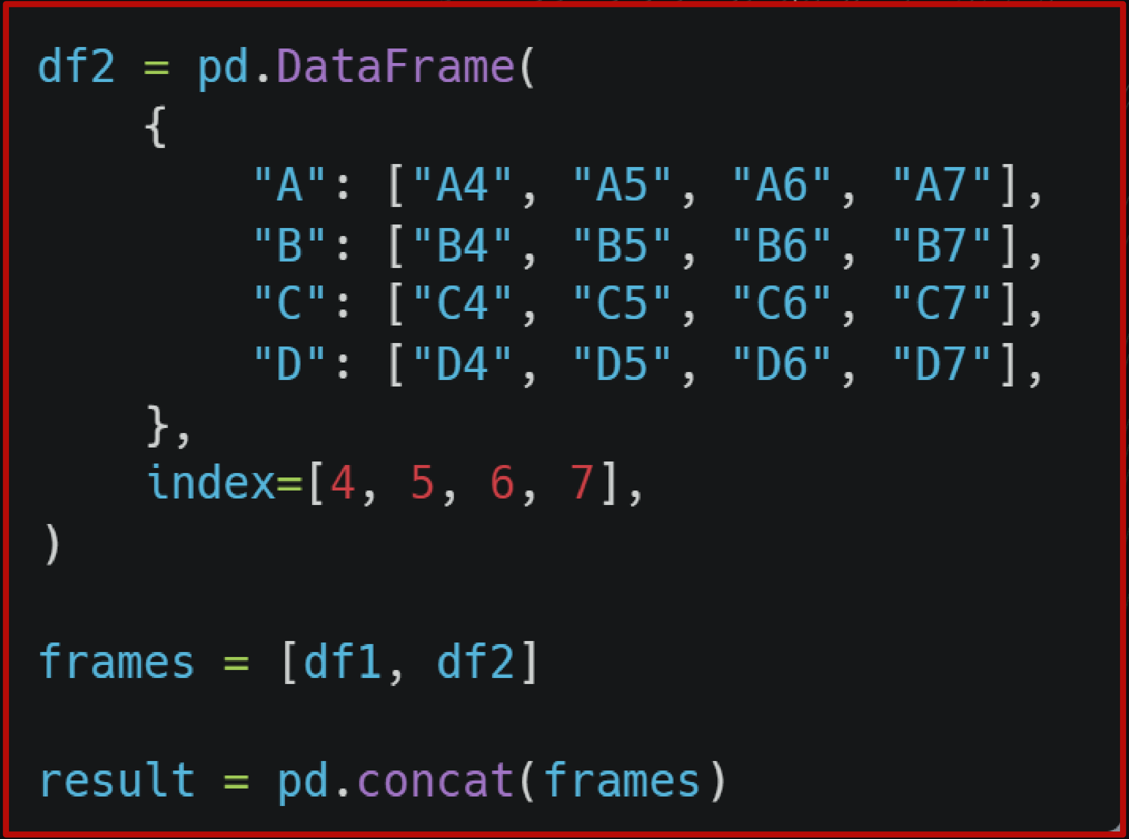


```
# importing pandas as pd
import pandas as pd

# Creating the dataframe
df = pd.DataFrame([[1, 2], [4, 5], [7, 8]],
     index=['cobra', 'viper', 'sidewinder'],
     columns=['max_speed', 'shield'])

#OUTPUT
            max_speed  shield
cobra               1       2
viper               4       5
sidewinder          7       8


df.loc['viper']


#OUTPUT
max_speed    4
shield       5
```

Code

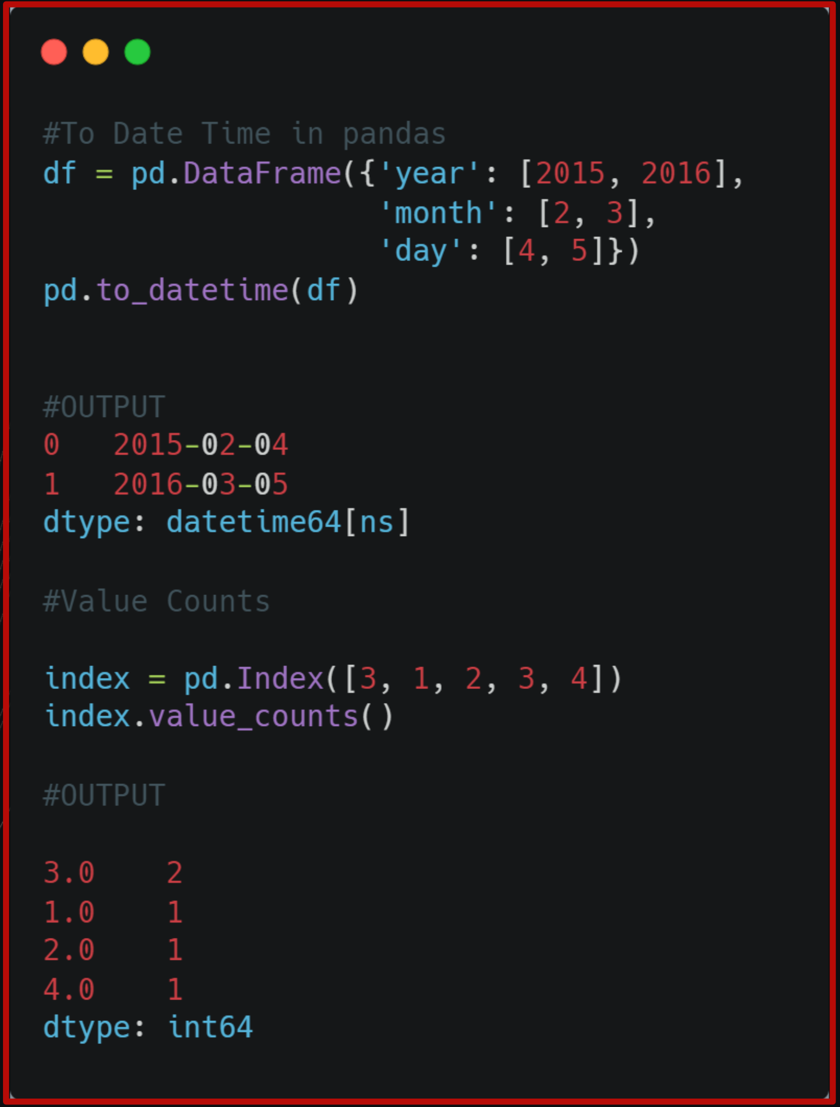


```
#CONCATENATING DATAFRAMES

df1 = pd.DataFrame(
    {
        "A": ["A0", "A1", "A2", "A3"],
        "B": ["B0", "B1", "B2", "B3"],
        "C": ["C0", "C1", "C2", "C3"],
        "D": ["D0", "D1", "D2", "D3"],
    },
    index=[0, 1, 2, 3],
)
```

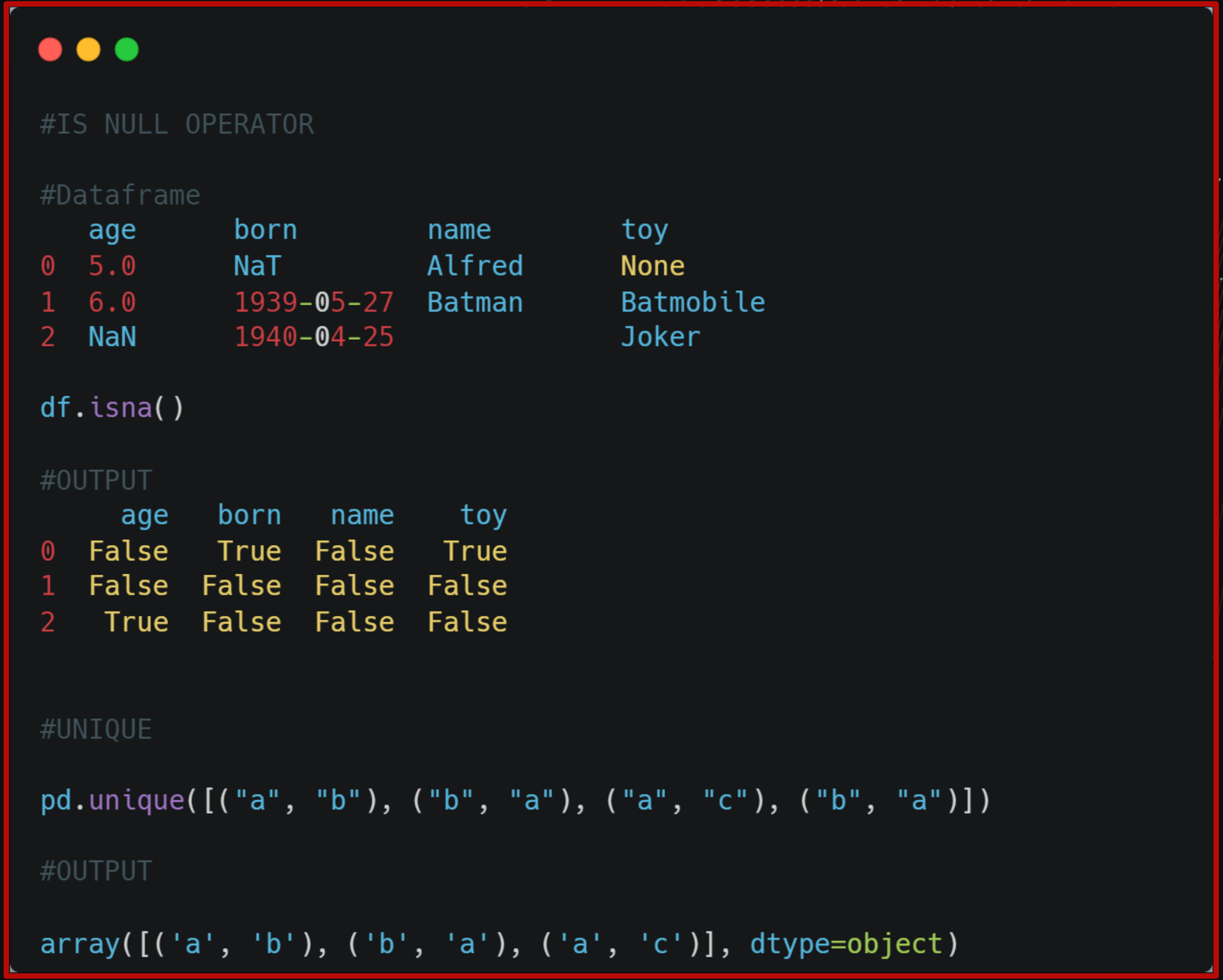


```
df2 = pd.DataFrame(
    {
        "A": ["A4", "A5", "A6", "A7"],
        "B": ["B4", "B5", "B6", "B7"],
        "C": ["C4", "C5", "C6", "C7"],
        "D": ["D4", "D5", "D6", "D7"],
    },
    index=[4, 5, 6, 7],
)

frames = [df1, df2]

result = pd.concat(frames)
```

Code



```
#To Date Time in pandas
df = pd.DataFrame({'year': [2015, 2016],
                   'month': [2, 3],
                   'day': [4, 5]})
pd.to_datetime(df)


#OUTPUT
0   2015-02-04
1   2016-03-05
dtype: datetime64[ns]

#Value Counts

index = pd.Index([3, 1, 2, 3, 4])
index.value_counts()

#OUTPUT

3.0    2
1.0    1
2.0    1
4.0    1
dtype: int64
```



```
#IS NULL OPERATOR

#Dataframe
   age      born        name        toy
0  5.0      NaT         Alfred      None
1  6.0      1939-05-27  Batman      Batmobile
2  NaN      1940-04-25              Joker

df.isna()

#OUTPUT
     age   born   name    toy
0  False   True  False   True
1  False  False  False  False
2   True  False  False  False


#UNIQUE

pd.unique([("a", "b"), ("b", "a"), ("a", "c"), ("b", "a")])

#OUTPUT

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

Congrats on learning
“Pandas & Numpy”.

Coming up next
“Matplotlib & Seaborn”

The End

Coding Club
IIT Guwahati