Name : Ankur Damke **Date:** 23/01/2024

Reg. ID : 201090082 **Group No :** 11

DATA SCIENCE AND ANALYSIS (DSA)

Lab

Experiment No. 1

Fundamentals of Python Programming

Aim: To study and implement basic python functions such as NumPy, pandas, matplotlib and seaborn.

Software Used : Python 3 with Jupyter Notebook.

Theory :

What is Python?

Python is a high-level, general-purpose, and very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in Software Industry. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like DJango (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks

- Multimedia
- Scientific computing
- Text processing and many more..

1) Python NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

2) Pandas

Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

3) Django

Django is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because Django provides built-in features for everything including Django Admin Interface, default database – SQLlite3, etc. When you’re building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use and that too for rapid development.

4) Matplotlib

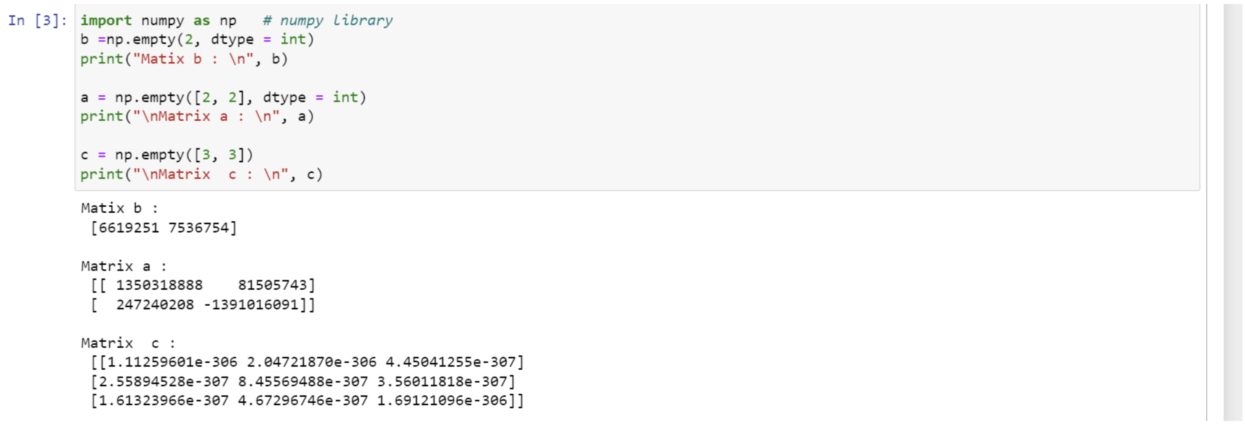
Matplotlib is an amazing visualization library in **Python** for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on **NumPy** arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

5) Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top matplotlib library and is also closely integrated with the data structures from **pandas.**

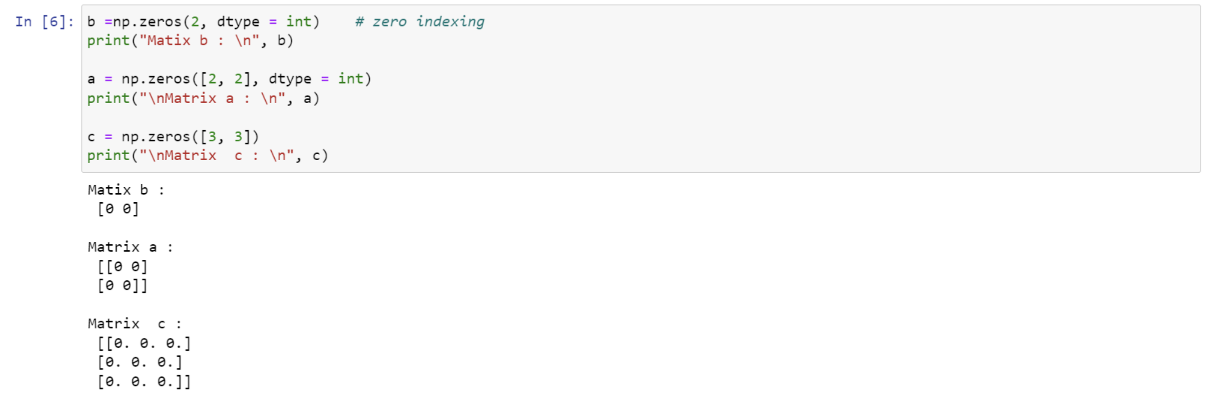
Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

Code and Output :



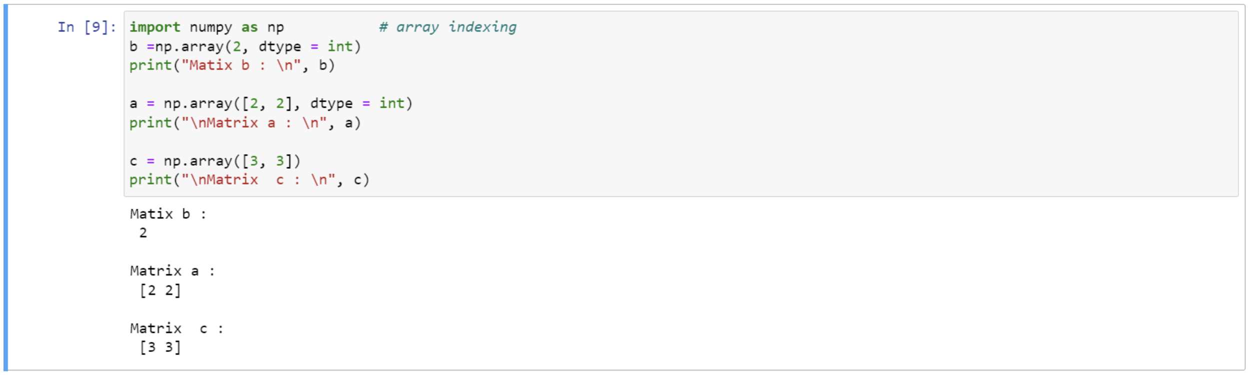
```
In [3]: import numpy as np   # numpy library
        b =np.empty(2, dtype = int)
        print("Matix b : \n", b)

        a = np.empty([2, 2], dtype = int)
        print("\nMatrix a : \n", a)

        c = np.empty([3, 3])
        print("\nMatrix  c : \n", c)
```

```
Matix b :
 [6619251 7536754]

Matrix a :
 [[ 1350318888    81505743]
 [  247240208 -1391016091]]

Matrix  c :
 [[1.11259601e-306 2.04721870e-306 4.45041255e-307]
 [2.55894528e-307 8.45569488e-307 3.56011818e-307]
 [1.61323966e-307 4.67296746e-307 1.69121096e-306]]
```



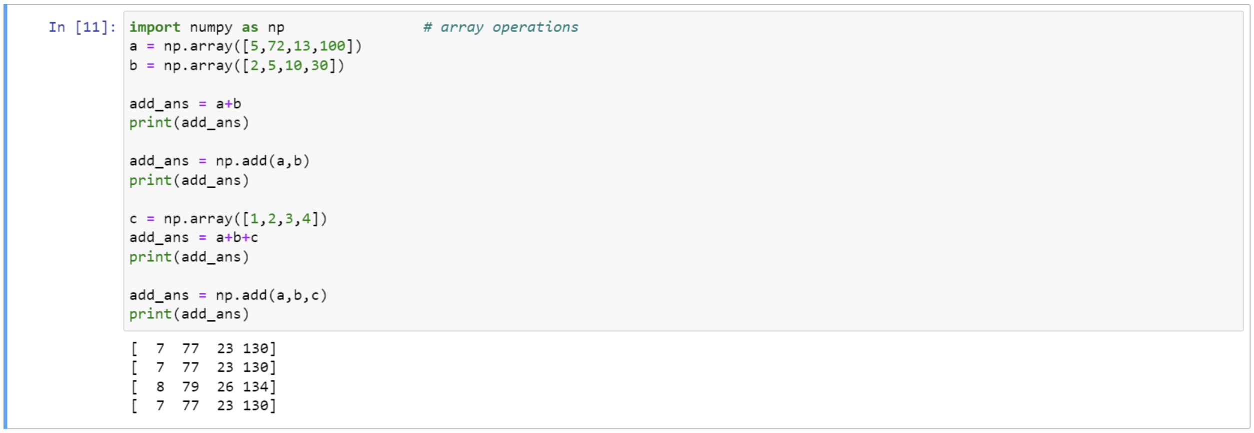
```
In [6]: b =np.zeros(2, dtype = int)    # zero indexing
        print("Matix b : \n", b)

        a = np.zeros([2, 2], dtype = int)
        print("\nMatrix a : \n", a)

        c = np.zeros([3, 3])
        print("\nMatrix  c : \n", c)
```

```
Matix b :
 [0 0]

Matrix a :
 [[0 0]
 [0 0]]

Matrix  c :
 [[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```



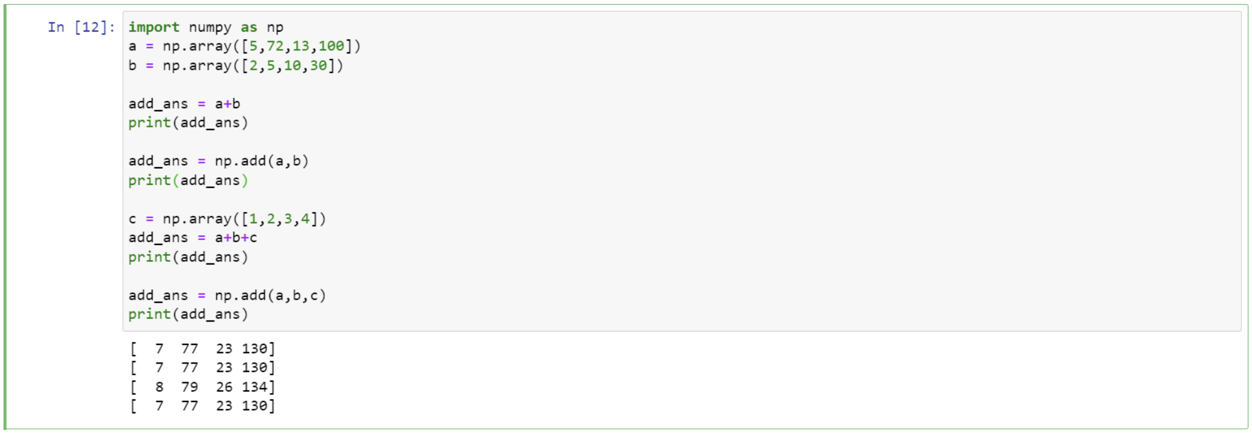
```
In [9]: import numpy as np           # array indexing
        b =np.array(2, dtype = int)
        print("Matix b : \n", b)

        a = np.array([2, 2], dtype = int)
        print("\nMatrix a : \n", a)

        c = np.array([3, 3])
        print("\nMatrix  c : \n", c)
```

```
Matix b :
 2

Matrix a :
 [2 2]

Matrix  c :
 [3 3]
```



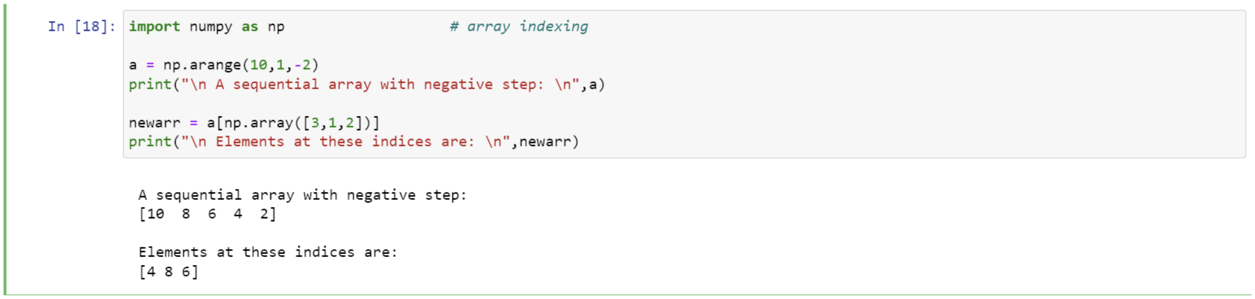
```
In [11]: import numpy as np                # array operations
         a = np.array([5,72,13,100])
         b = np.array([2,5,10,30])

         add_ans = a+b
         print(add_ans)

         add_ans = np.add(a,b)
         print(add_ans)

         c = np.array([1,2,3,4])
         add_ans = a+b+c
         print(add_ans)

         add_ans = np.add(a,b,c)
         print(add_ans)
```

```
[  7  77  23 130]
[  7  77  23 130]
[  8  79  26 134]
[  7  77  23 130]
```



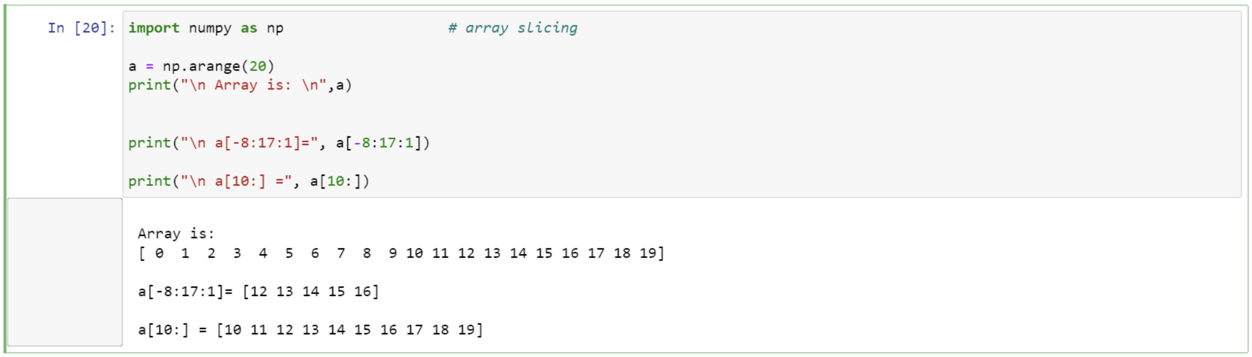
```
In [12]: import numpy as np
         a = np.array([5,72,13,100])
         b = np.array([2,5,10,30])

         add_ans = a+b
         print(add_ans)

         add_ans = np.add(a,b)
         print(add_ans)

         c = np.array([1,2,3,4])
         add_ans = a+b+c
         print(add_ans)

         add_ans = np.add(a,b,c)
         print(add_ans)
```

```
[  7  77  23 130]
[  7  77  23 130]
[  8  79  26 134]
[  7  77  23 130]
```



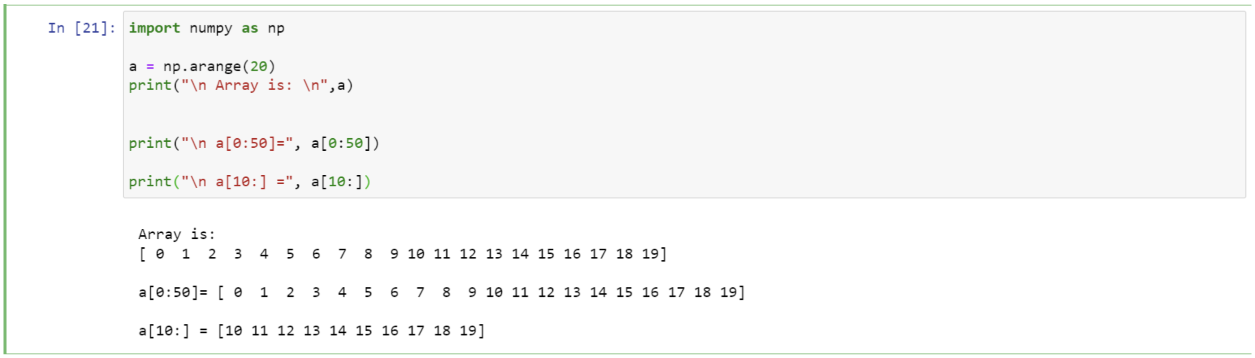
In [18]:
```python
import numpy as np                    # array indexing

a = np.arange(10,1,-2)
print("\n A sequential array with negative step: \n",a)

newarr = a[np.array([3,1,2])]
print("\n Elements at these indices are: \n",newarr)
```

```
 A sequential array with negative step: 
 [10  8  6  4  2]

 Elements at these indices are: 
 [4 8 6]
```



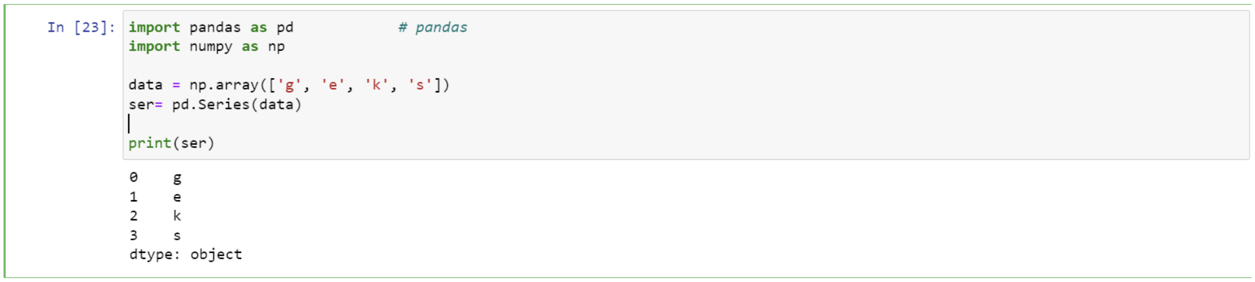
In [20]:
```python
import numpy as np                    # array slicing

a = np.arange(20)
print("\n Array is: \n",a)


print("\n a[-8:17:1]=", a[-8:17:1])

print("\n a[10:] =", a[10:])
```

```
 Array is: 
 [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]

 a[-8:17:1]= [12 13 14 15 16]

 a[10:] = [10 11 12 13 14 15 16 17 18 19]
```



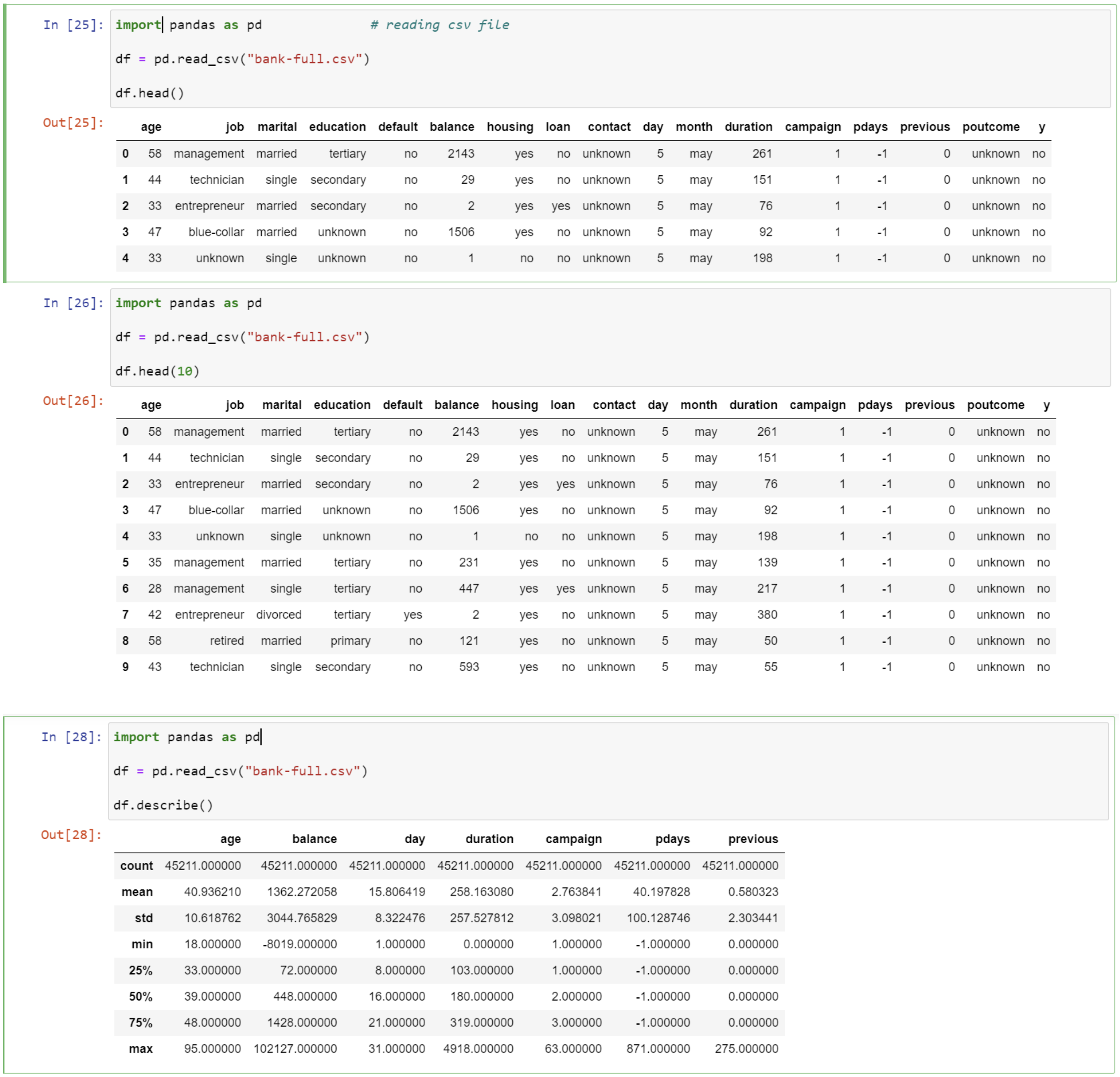
In [21]:
```python
import numpy as np

a = np.arange(20)
print("\n Array is: \n",a)


print("\n a[0:50]=", a[0:50])

print("\n a[10:] =", a[10:])
```

```
 Array is: 
 [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]

 a[0:50]= [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]

 a[10:] = [10 11 12 13 14 15 16 17 18 19]
```



In [23]:
```python
import pandas as pd            # pandas
import numpy as np

data = np.array(['g', 'e', 'k', 's'])
ser= pd.Series(data)

print(ser)
```

```
0    g
1    e
2    k
3    s
dtype: object
```



In [25]:

```python
import pandas as pd                # reading csv file

df = pd.read_csv("bank-full.csv")

df.head()
```

Out[25]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no

In [26]:

```python
import pandas as pd

df = pd.read_csv("bank-full.csv")

df.head(10)
```

Out[26]:

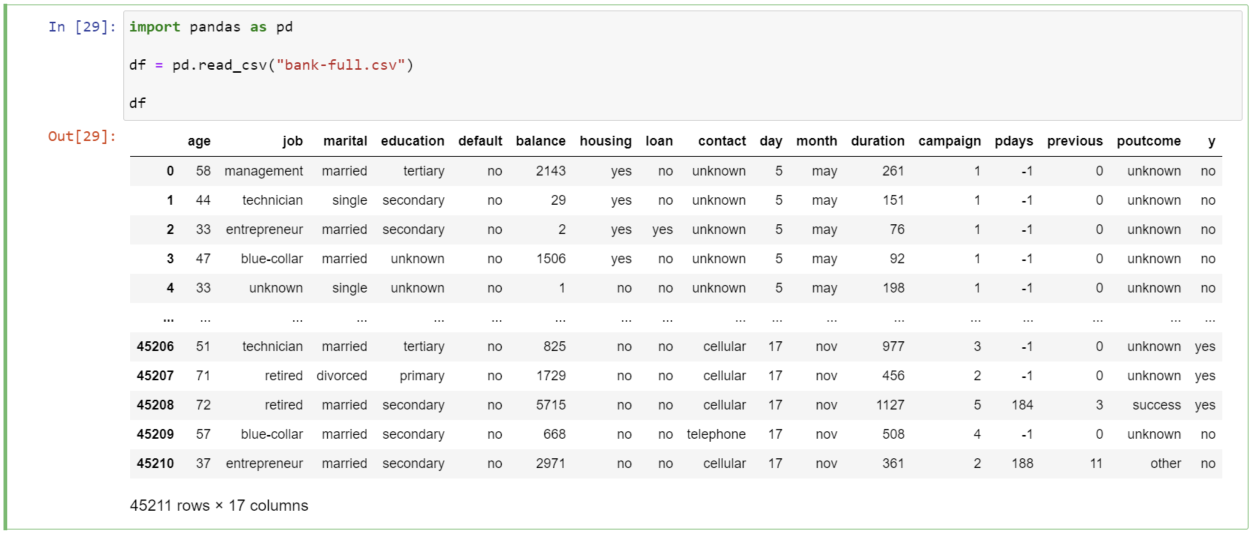
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
5	35	management	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no
6	28	management	single	tertiary	no	447	yes	yes	unknown	5	may	217	1	-1	0	unknown	no
7	42	entrepreneur	divorced	tertiary	yes	2	yes	no	unknown	5	may	380	1	-1	0	unknown	no
8	58	retired	married	primary	no	121	yes	no	unknown	5	may	50	1	-1	0	unknown	no
9	43	technician	single	secondary	no	593	yes	no	unknown	5	may	55	1	-1	0	unknown	no

In [28]:

```python
import pandas as pd

df = pd.read_csv("bank-full.csv")

df.describe()
```

Out[28]:

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

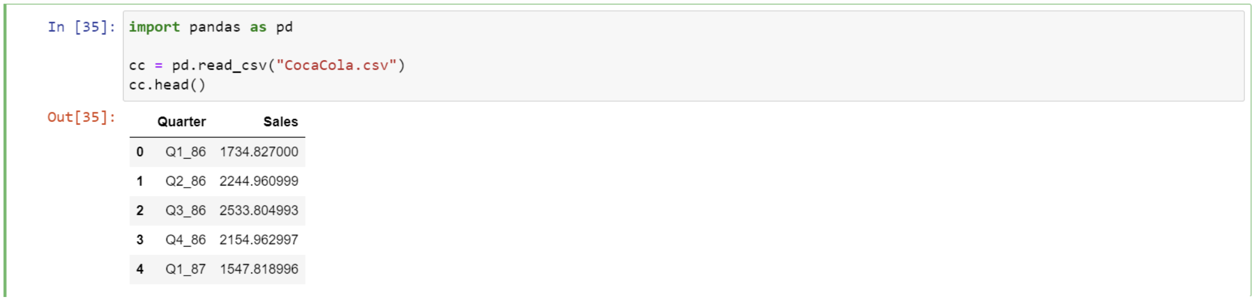


In [29]:
```python
import pandas as pd

df = pd.read_csv("bank-full.csv")

df
```

Out[29]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

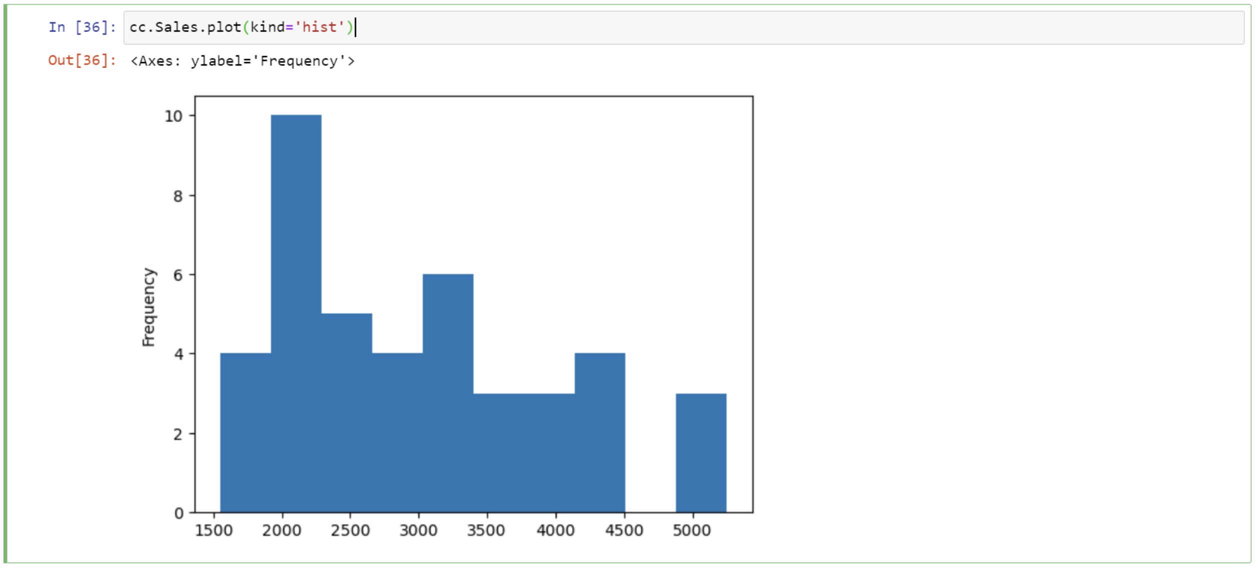
45211 rows × 17 columns



In [35]:
```python
import pandas as pd

cc = pd.read_csv("CocaCola.csv")
cc.head()
```

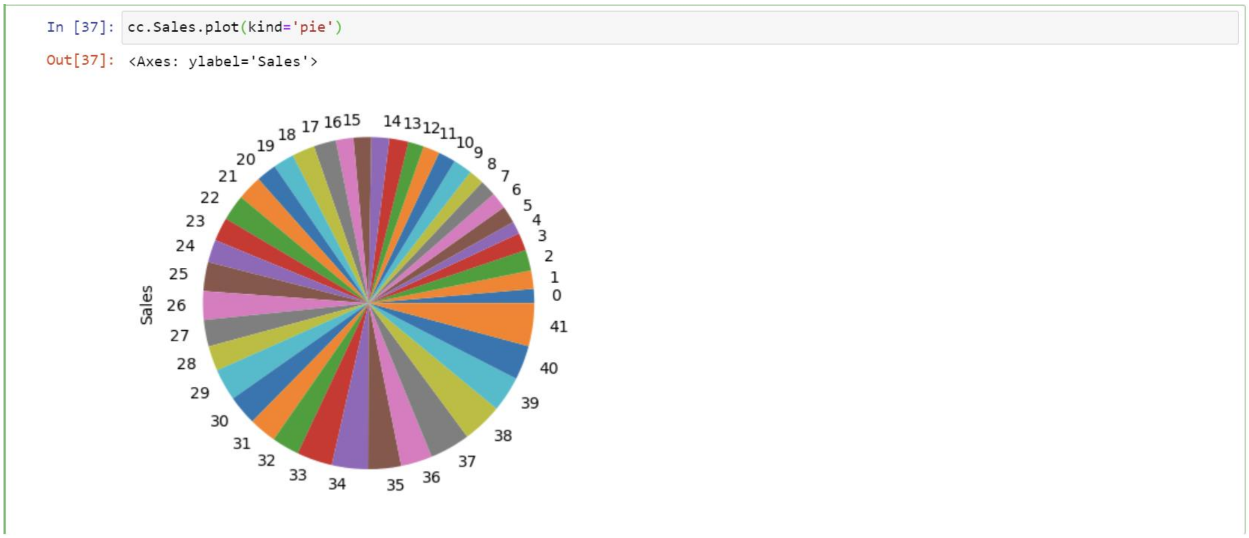
Out[35]:

	Quarter	Sales
0	Q1_86	1734.827000
1	Q2_86	2244.960999
2	Q3_86	2533.804993
3	Q4_86	2154.962997
4	Q1_87	1547.818996

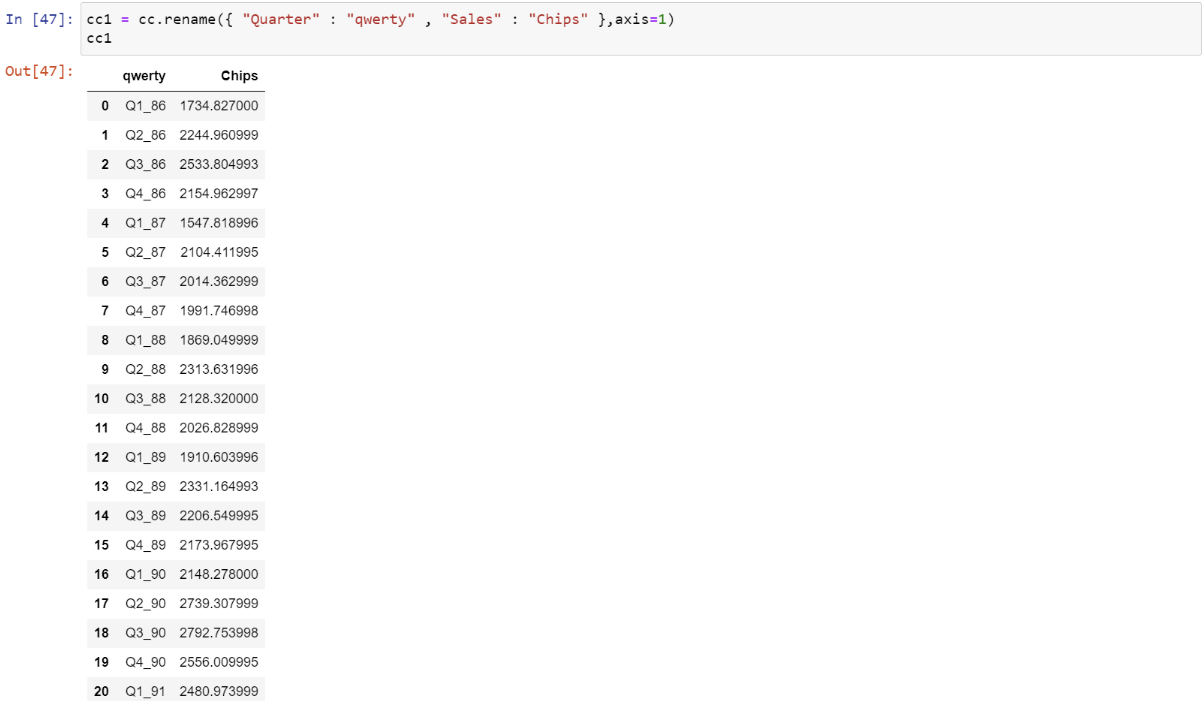


In [36]:
```python
cc.Sales.plot(kind='hist')
```

Out[36]:
```
<Axes: ylabel='Frequency'>
```

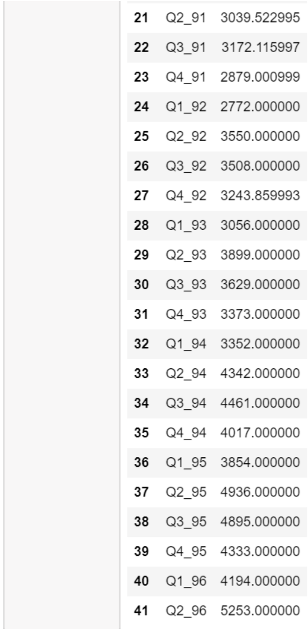


```
In [37]: cc.Sales.plot(kind='pie')

Out[37]: <Axes: ylabel='Sales'>
```



```
In [47]: cc1 = cc.rename({ "Quarter" : "qwerty" , "Sales" : "Chips" },axis=1)
         cc1

Out[47]:
```

	qwerty	Chips
0	Q1_86	1734.827000
1	Q2_86	2244.960999
2	Q3_86	2533.804993
3	Q4_86	2154.962997
4	Q1_87	1547.818996
5	Q2_87	2104.411995
6	Q3_87	2014.362999
7	Q4_87	1991.746998
8	Q1_88	1869.049999
9	Q2_88	2313.631996
10	Q3_88	2128.320000
11	Q4_88	2026.828999
12	Q1_89	1910.603996
13	Q2_89	2331.164993
14	Q3_89	2206.549995
15	Q4_89	2173.967995
16	Q1_90	2148.278000
17	Q2_90	2739.307999
18	Q3_90	2792.753998
19	Q4_90	2556.009995
20	Q1_91	2480.973999



21	Q2_91	3039.522995
22	Q3_91	3172.115997
23	Q4_91	2879.000999
24	Q1_92	2772.000000
25	Q2_92	3550.000000
26	Q3_92	3508.000000
27	Q4_92	3243.859993
28	Q1_93	3056.000000
29	Q2_93	3899.000000
30	Q3_93	3629.000000
31	Q4_93	3373.000000
32	Q1_94	3352.000000
33	Q2_94	4342.000000
34	Q3_94	4461.000000
35	Q4_94	4017.000000
36	Q1_95	3854.000000
37	Q2_95	4936.000000
38	Q3_95	4895.000000
39	Q4_95	4333.000000
40	Q1_96	4194.000000
41	Q2_96	5253.000000

Conclusion :

We have successfully implemented all the operations of NumPy, pandas, matplotlib and seaborn using Jupyter Notebook.