Machine Learning Assignment 3

Name: Akhila Boddu

Student ID: 700742171

GitHub link: https://github.com/AkhilaBoddu/ML-Assignment-3.git

Video Link: https://drive.google.com/file/d/15y3B1HLX6TzsCCWGeA_ACCblNOioSXox/view?usp=share_link

Question1

Numpy:

a. Using NumPy create random vector of size 15 having only Integers in the range 1-20.

Source Code:

#1. Numpy:

# Using NumPy create random vector of size 15 having only Integers in the range 1-20.

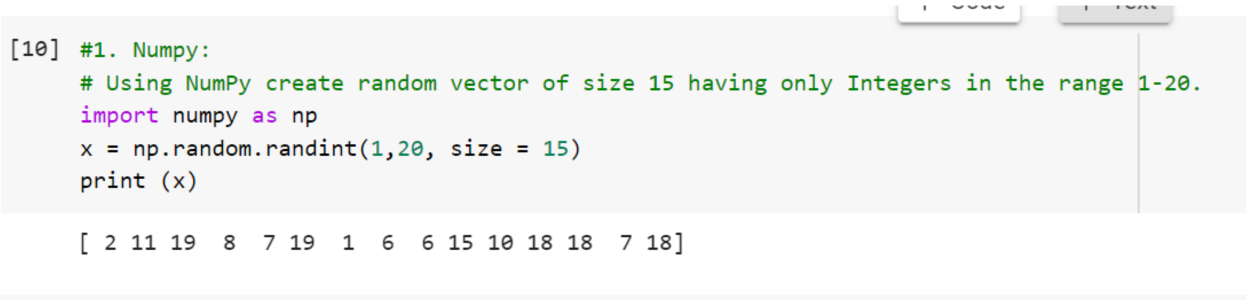
import numpy as np

x = np.random.randint(1,20, size = 15)

print (x)

Description:

In the above code Firstly I'm importing numpy, created a vector of size 15 with range 1-20 and printing it.



```
[10] #1. Numpy:
     # Using NumPy create random vector of size 15 having only Integers in the range 1-20.
     import numpy as np
     x = np.random.randint(1,20, size = 15)
     print (x)

     [ 2 11 19  8  7 19  1  6  6 15 10 18 18  7 18]
```

a.1. Reshape the array to 3 by 5

Source Code:

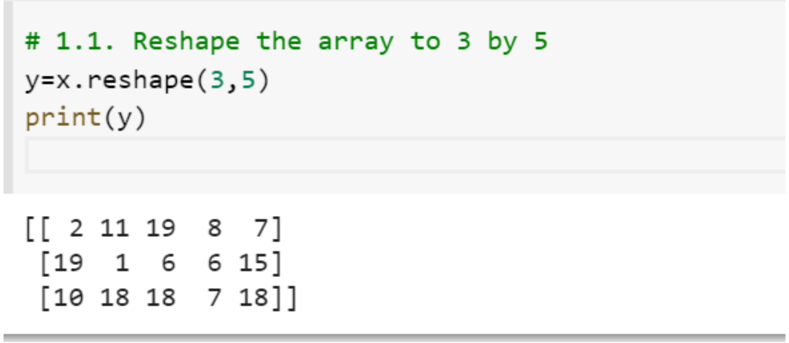
# Reshape the array to 3 by 5

y=x.reshape(3,5)

print(y)

Description:

In the above source code, First I’m just reshaping the array to 3 by 5 matrix, and printing it.



```
# 1.1. Reshape the array to 3 by 5
y=x.reshape(3,5)
print(y)
```

```
[[ 2 11 19  8  7]
 [19  1  6  6 15]
 [10 18 18  7 18]]
```

a.2. Print array shape.

Source Code:

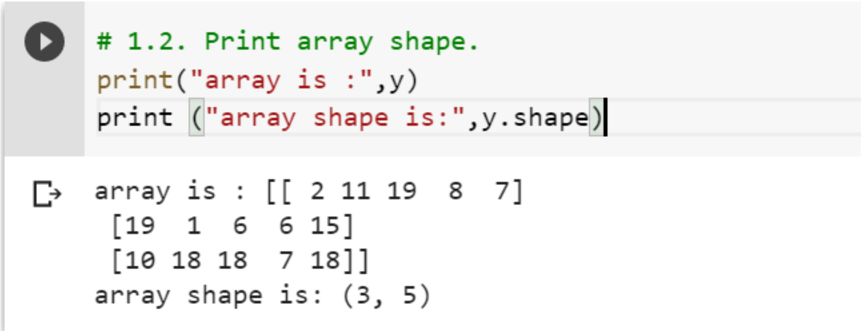
# Print array shape.

print("array is :",y)

print ("array shape is:",y.shape)

Description:

In the above source code, First printing the above used array shape.



```
# 1.2. Print array shape.
print("array is :",y)
print ("array shape is:",y.shape)
```

```
array is : [[ 2 11 19  8  7]
 [19  1  6  6 15]
 [10 18 18  7 18]]
array shape is: (3, 5)
```

a.3. Replace the max in each row by 0

Source Code:

# 1.3. Replace the max in each row by 0

new_a = np.where(y == [

  [i]

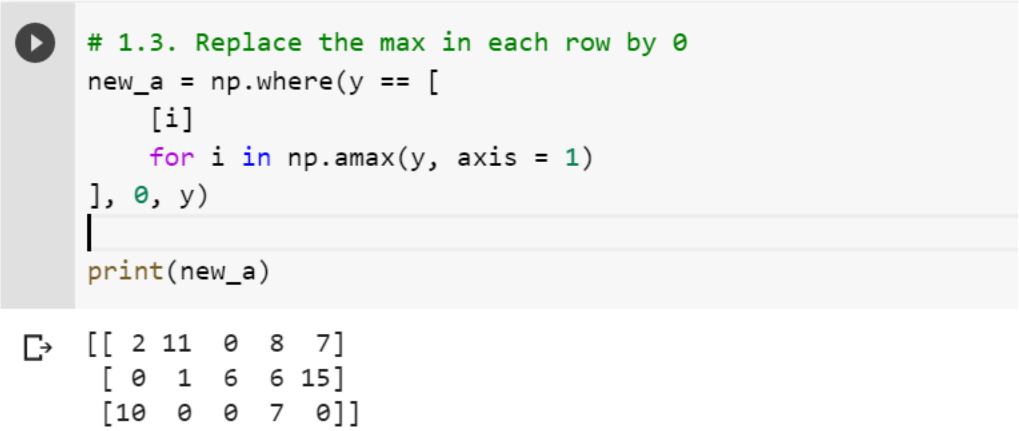
  for i in np.amax(y, axis = 1)

], 0, y)

print(new_a)

Description:

In the above source code, First I'm trying to replace the max in each row by 0, By using the amax in the for loop.



```
# 1.3. Replace the max in each row by 0
new_a = np.where(y == [
    [i]
    for i in np.amax(y, axis = 1)
], 0, y)

print(new_a)
```

```
[[ 2 11  0  8  7]
 [ 0  1  6  6 15]
 [10  0  0  7  0]]
```

Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type of the array.

Source Code:

# Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type of the array.

import numpy as np

# create a 2-dimensional array of size 4x3

arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]], dtype=np.int32)

# print the array shape

print("Array shape:", arr.shape)

# print the array type

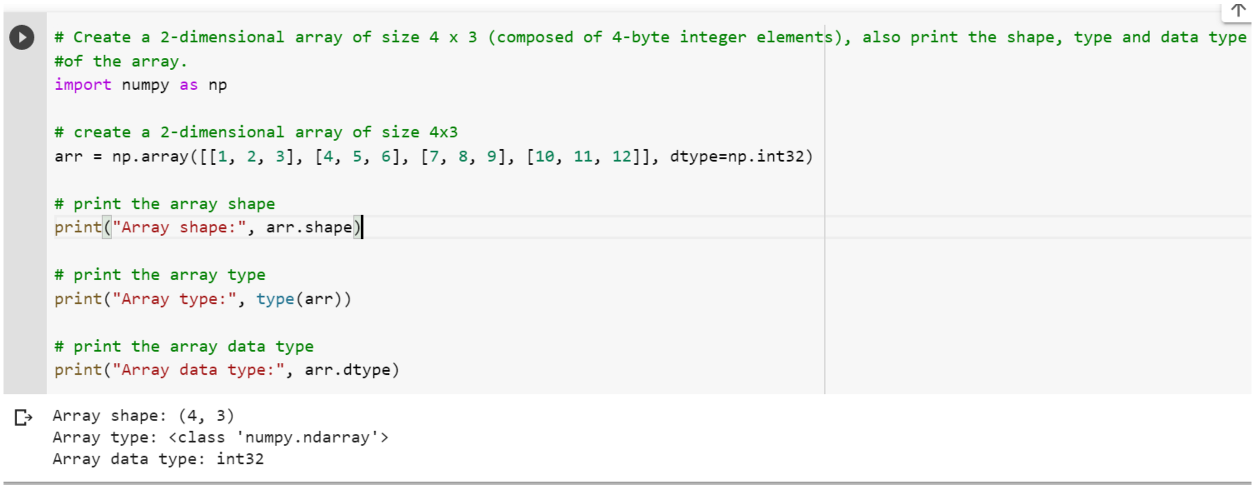
print("Array type:", type(arr))

# print the array data type

print("Array data type:", arr.dtype)

Description:

In the above source code, First I’m creating a 2-dimensional array of size 4x3 by using np.array. printing the array shape by using arr.shape , array type by using type(arr) and also array data type by using arr.dtype.



```
# Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type
#of the array.
import numpy as np

# create a 2-dimensional array of size 4x3
arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]], dtype=np.int32)

# print the array shape
print("Array shape:", arr.shape)

# print the array type
print("Array type:", type(arr))

# print the array data type
print("Array data type:", arr.dtype)
```

```
Array shape: (4, 3)
Array type: <class 'numpy.ndarray'>
Array data type: int32
```

Question1.b

Write a program to compute the eigenvalues and right eigenvectors of a given square array given below: [[3 -2] [1 0]]

Source Code:

#1(b) Write a program to compute the eigenvalues and right eigenvectors

import numpy as np

# define the square array

A = np.array([[3, -2], [1, 0]])

# compute the eigenvalues and right eigenvectors

eigenvalues, eigenvectors = np.linalg.eig(A)

# print the eigenvalues and right eigenvectors

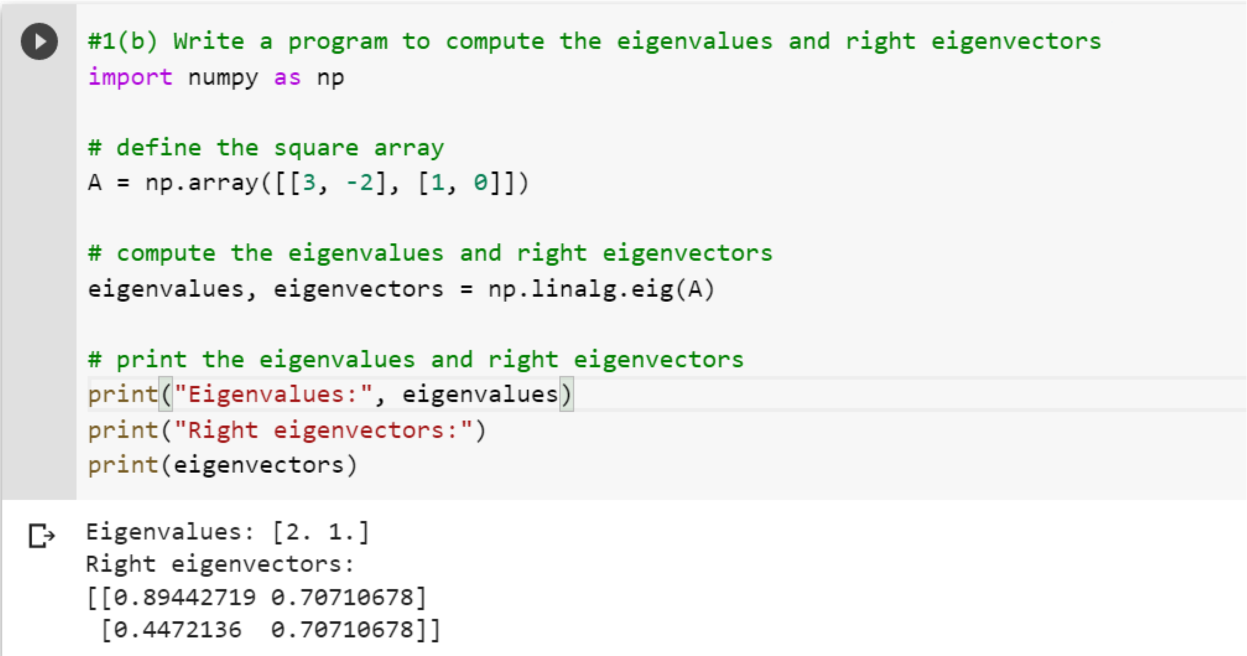
print("Eigenvalues:", eigenvalues)

print("Right eigenvectors:")

print(eigenvectors)

Description:

In the above source code, First I’m trying to compute the eigenvalues and right eigenvectors by using np.linalg.eig(A) and printing their values.



```
#1(b) Write a program to compute the eigenvalues and right eigenvectors
import numpy as np

# define the square array
A = np.array([[3, -2], [1, 0]])

# compute the eigenvalues and right eigenvectors
eigenvalues, eigenvectors = np.linalg.eig(A)

# print the eigenvalues and right eigenvectors
print("Eigenvalues:", eigenvalues)
print("Right eigenvectors:")
print(eigenvectors)
```

```
Eigenvalues: [2. 1.]
Right eigenvectors:
[[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

Question1.c

Compute the sum of the diagonal element of a given array.

[[0 1 2] [3 4 5]]

Source Code:

#1(c)Compute the sum of the diagonal element of a given array.

import numpy as np

# define the array

A = np.array([[0, 1, 2], [3, 4, 5]])

# compute the sum of the diagonal elements

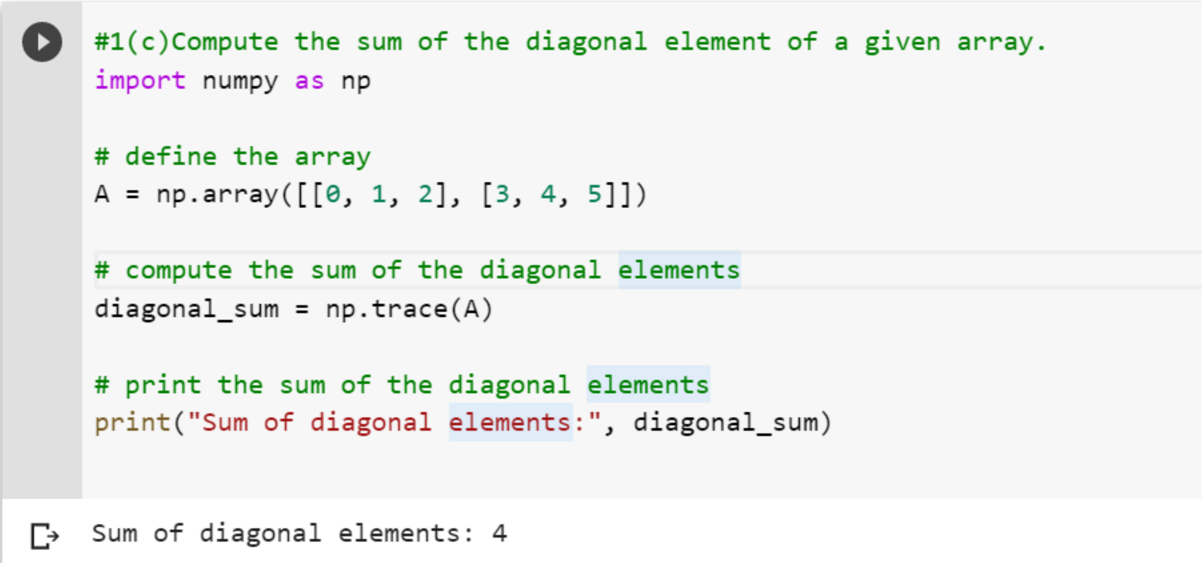
diagonal_sum = np.trace(A)

# print the sum of the diagonal elements

print("Sum of diagonal elements:", diagonal_sum)

Description:

In the above source code, First I’m trying to compute the sum of the diagonal element of a given array. Here I’m defining the array and finding the diagonal sum by using np.trace() and printing it.



```
#1(c)Compute the sum of the diagonal element of a given array.
import numpy as np

# define the array
A = np.array([[0, 1, 2], [3, 4, 5]])

# compute the sum of the diagonal elements
diagonal_sum = np.trace(A)

# print the sum of the diagonal elements
print("Sum of diagonal elements:", diagonal_sum)
```

```
Sum of diagonal elements: 4
```

Question1.d

Write a NumPy program to create a new shape to an array without changing its data.

Reshape 3x2:

[[1 2]

[3 4]

[5 6]]

Reshape 2x3:

[[1 2 3]

[4 5 6]]

Source Code:

#1(d)Write a NumPy program to create a new shape to an array without changing its data.

import numpy as np

# define the original array

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

# reshape to 3x2

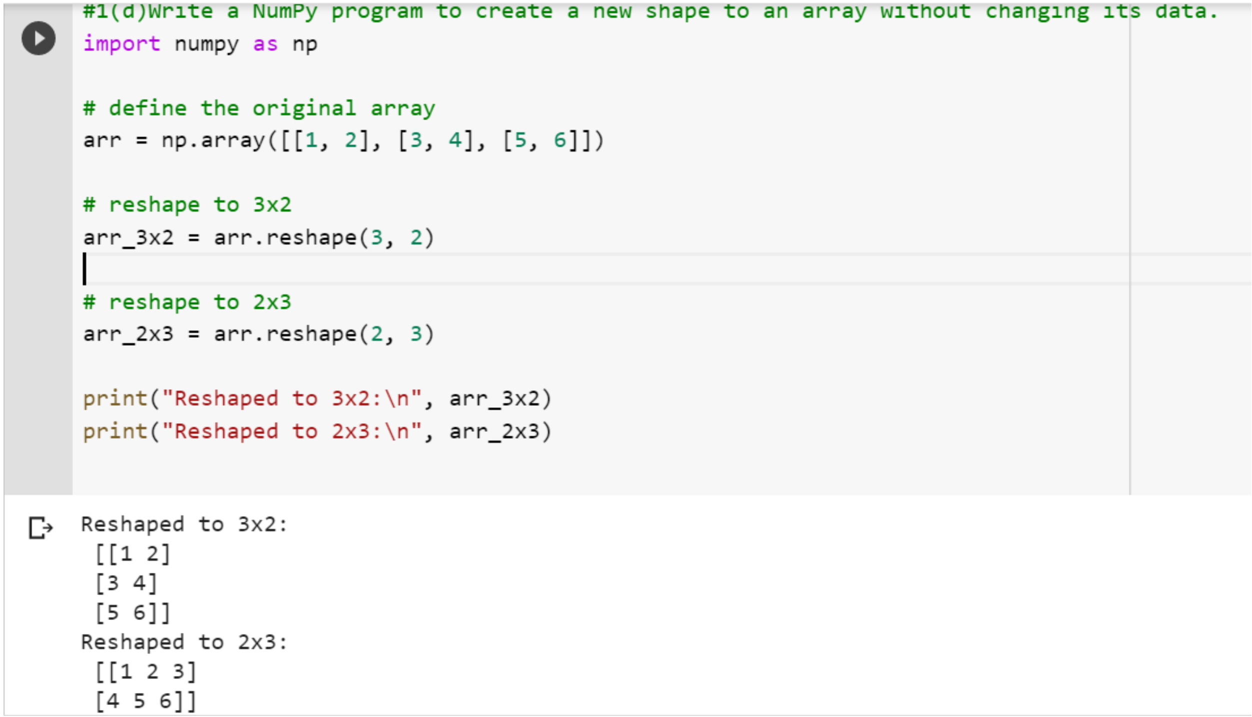
arr_3x2 = arr.reshape(3, 2)

# reshape to 2x3

```
arr_2x3 = arr.reshape(2, 3)

print("Reshaped to 3x2:\n", arr_3x2)

print("Reshaped to 2x3:\n", arr_2x3)
```

Description:

In the above source code, First I’m trying to create a new shape to an array without changing its data. Here I’m defining 3x2 matrix then using arr.reshape(3x2) and again reshaping it to (2,3) without changing its data.



```
#1(d)Write a NumPy program to create a new shape to an array without changing its data.
import numpy as np

# define the original array
arr = np.array([[1, 2], [3, 4], [5, 6]])

# reshape to 3x2
arr_3x2 = arr.reshape(3, 2)

# reshape to 2x3
arr_2x3 = arr.reshape(2, 3)

print("Reshaped to 3x2:\n", arr_3x2)
print("Reshaped to 2x3:\n", arr_2x3)
```

```
Reshaped to 3x2:
 [[1 2]
 [3 4]
 [5 6]]
Reshaped to 2x3:
 [[1 2 3]
 [4 5 6]]
```

Question2

Matplotlib

1. Write a Python programming to create a below chart of the popularity of programming Languages.

2. Sample data: Programming languages: Java, Python, PHP, JavaScript, C#, C++

Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

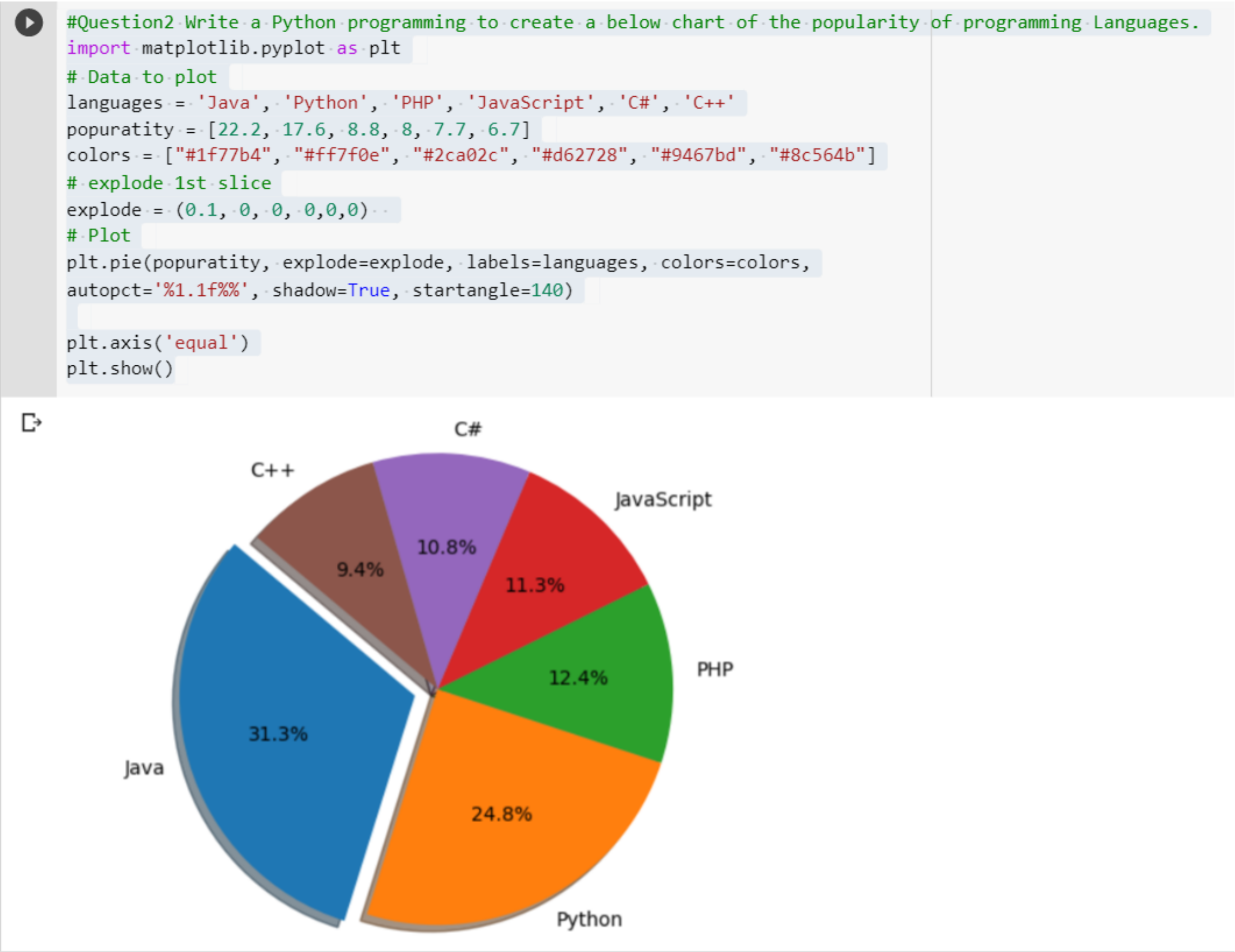
Source Code:

```
# Write a Python programming to create a below chart of the popularity of programming Languages.

import matplotlib.pyplot as plt

# Data to plot

languages = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
```

```
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]

# explode 1st slice

explode = (0.1, 0, 0, 0,0,0)

# Plot

plt.pie(popularity, explode=explode, labels=languages, colors=colors,

autopct='%1.1f%%', shadow=True, startangle=140)

plt.axis('equal')

plt.show()
```

Description:

In the above source code, First I’m trying to create chart of the popularity of Programming languages: Java, Python, PHP, JavaScript, C#, C++. By using plt.pie with popularity, explode, languages and colors. And also plt.axis to diving the popularity equally as defined above. Plt.show() is used to display the pie chart.



```
#Question2 Write a Python programming to create a below chart of the popularity of programming Languages.
import matplotlib.pyplot as plt
# Data to plot
languages = 'Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++'
popuratity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd", "#8c564b"]
# explode 1st slice
explode = (0.1, 0, 0, 0,0,0)
# Plot
plt.pie(popuratity, explode=explode, labels=languages, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=140)

plt.axis('equal')
plt.show()
```