GitHub URL: https://github.com/nitishchappidi/Assignment1

Video URL:
https://drive.google.com/file/d/1GhZ_uitApyd9NxGwKvx6hT6iRas2_hxT/view?usp=sharing

Machine Learning Programming Assignment – 1

1. Numpy:

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

1. Reshape the array to 3 by 5

2. Print array shape.

3. Replace the max in each row by 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.

In [23]:
```
import numpy as np

# create random vector of size 15 with integers in range 1-20
vector = np.random.randint(1, 21, size=15)

# reshape to 3 by 5
array = vector.reshape(3, 5)

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

# replace max in each row by 0
array[np.arange(len(array)), array.argmax(axis=1)] = 0

print(array)
```

```
Array shape: (3, 5)
[[ 0 15 10 15 11]
 [ 5  0 16 12 20]
 [ 3  0  1 16 11]]
```

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

In [25]:
```
import numpy as np

# create square array
arr = np.array([[3, -2], [1, 0]])

# compute eigenvalues and right eigenvectors
eigen_vals, eigen_vecs = np.linalg.eig(arr)

print("Eigen values:", eigen_vals)
print("Right Eigen vectors:\n", eigen_vecs)
```

```
Eigen values: [2. 1.]
Right Eigen vectors:
 [[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

c. Compute the sum of the diagonal element of a given array. [[0 1 2] [3 4 5]]

In [26]:
```
import numpy as np

# create array
arr = np.array([[0, 1, 2], [3, 4, 5]])

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

print("Diagonal elements Sum = ", diagonal_sum)
```

```
Diagonal elements Sum =  4
```

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]]

In [27]:
```python
import numpy as np

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

# reshape to 3 by 2
arr_3_2 = arr.reshape(3, 2)

# reshape to 2 by 3
arr_2_3 = arr.reshape(2, 3)

print("Original array:\n", arr)
print("3 by 2 Array:\n", arr_3_2)
print("2 by 3 Array:\n", arr_2_3)
```

```
Original array:
 [[1 2]
 [3 4]
 [5 6]]
3 by 2 Array:
 [[1 2]
 [3 4]
 [5 6]]
2 by 3 Array:
 [[1 2 3]
 [4 5 6]]
```

2. 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

In [28]:
```python
# Import the matplotlib.pyplot module, which allows us to create plots
import matplotlib.pyplot as plt

# Define the data we want 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"]

# Define how much we want to "explode" each slice of the pie chart
explode = (0.1, 0, 0, 0, 0, 0)

# Use the pie function to create the pie chart
plt.pie(popularity,    # The data to plot (popularity percentages)
        explode=explode,    # How much to "explode" each slice
        labels=languages,   # Labels for each slice (the language names)
        colors=colors,      # Colors for each slice
        autopct='%1.1f%%',  # Format for the percentage labels
        shadow=True,        # Whether to include a shadow effect
        startangle=140      # The angle at which the chart starts
       )

# Set the aspect ratio of the chart to be equal, so it appears circular
plt.axis('equal')

# Display the chart
plt.show()
```

OUTPUT: