2) Getting Started With Matplotlib.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

%matplotlib is a magic command. This performs the necessary behind-the-scenes setup for IPython to work correctly hand in hand wi matplotlib; it does not, however, actually execute any Python import commands, that is, no names are added to the namespace.

If the %matplotlib magic is called without an argument, the output of a plotting command is displayed using the default matplotlib back in a separate window. Alternatively, the backend can be explicitly requested.

A particularly interesting backend, provided by IPython, is the inline backend. This is available only for the Jupyter Notebook and the Jupyter QtConsole. It can be invoked as follows:

In [1]:
```
1 %matplotlib inline
```

With this backend, the output of plotting commands is displayed inline within frontends like the Jupyter notebook, directly below the cc cell that produced it. The resulting plots will then also be stored in the notebook document.

Some Useful Functions:

1) **plot().**

Plots the points.

2) **xlabel().**

Names the x axis.

3) **ylabel().**

Names the y axis.

4) **title().**

Gives a title to a graph.

5) **show().**

Shows the plot.

6) **axis().**

This function is used to set some axis properties to the graph.

Note: For every x, y pair of arguments, there is an optional third argument which is the format string that indicates the color and line ty of the plot. The letters and symbols of the format string are from MATLAB, and you concatenate a color string with a line style string. T default format string is 'b-', which is a solid blue line.

In [4]:

```
import numpy as np
import matplotlib.pyplot as plt

# Create numerical ranges.
x = np.arange(5)
print(x)
print('\n------------------\n')

y = 2*x
print(y)
print('\n------------------\n')

z = 0 / (x + 0.1)
print(z)
print('\n------------------\n')
```

```
[0 1 2 3 4]

------------------

[0 2 4 6 8]

------------------

[0. 0. 0. 0. 0.]

------------------
```

In [5]:

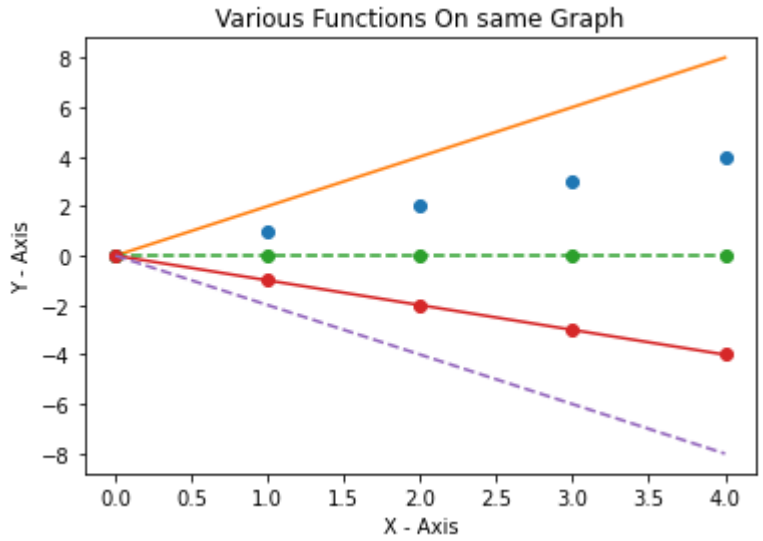
```
# Plotting the points
plt.plot(x, 'o')
plt.plot(y, '-')
plt.plot(z, 'o--')
plt.plot(-x, 'o-')
plt.plot(-y, '--')

# Naming the x axis
plt.xlabel('X - Axis')

# Naming the y axis
plt.ylabel('Y - Axis')

# Giving a title to the graph.
plt.title('Various Functions On same Graph')

# Showing the plot.
plt.show()
```



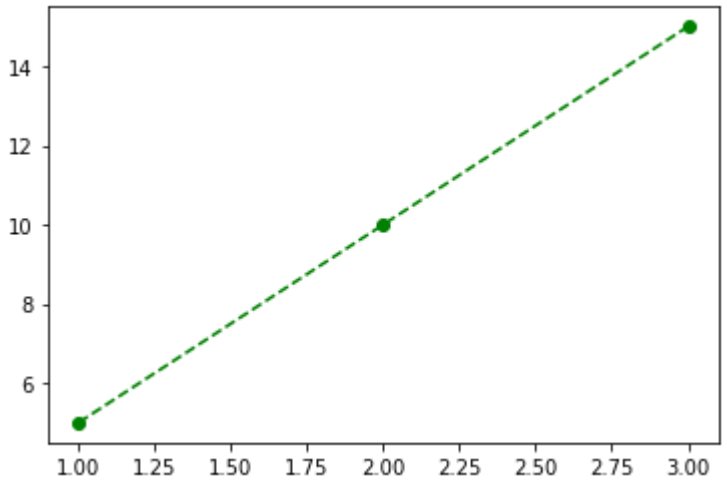
In [8]:

```
# x axis values
x = [1,2,3]

# Corresponding y axis values
y = [5,10,15]

# Plotting the points
plt.plot(x, y, 'go--')

# Showing the plot.
plt.show()
```



In [9]:

```
# Plotting the points
plt.plot(x,'ro-')

# Making the axis disappears.
plt.axis('off')

# Showing the plot.
plt.show()
```