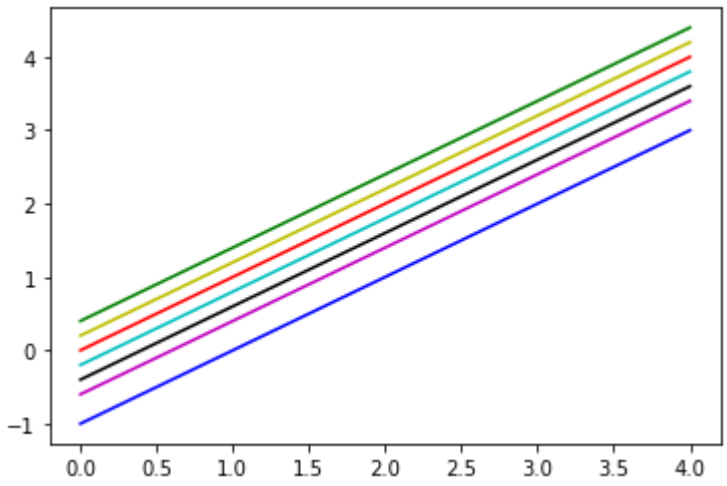
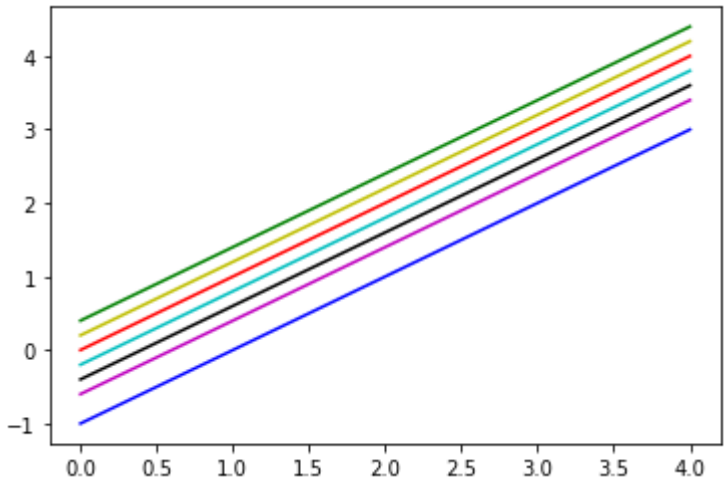
In [1]:

```
# Colors, Styles, and Markers
# Let's start with colors. The following code lists all the
# primary colors supported by Matplotlib.

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(5)
y = x
plt.plot(x, y+0.4, 'g')
plt.plot(x, y+0.2, 'y')
plt.plot(x, y, 'r')
plt.plot(x, y-0.2, 'c')
plt.plot(x, y-0.4, 'k')
plt.plot(x, y-0.6, 'm')
plt.plot(x, y-0.8, 'w')
plt.plot(x, y-1, 'b')
plt.show()
```



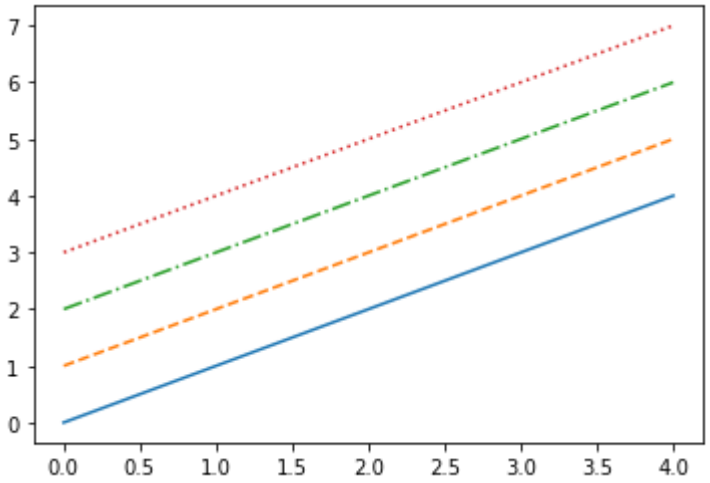
In [2]:

```
# You can also write the previous code as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(5)
y = x
plt.plot(x, y+0.4, 'g', x, y+0.2, 'y', x, y, 'r',
         x, y-0.2, 'c', x, y-0.4,'k', x, y-0.6, 'm',
         x, y-0.8, 'w', x, y-1, 'b')
plt.show()
```



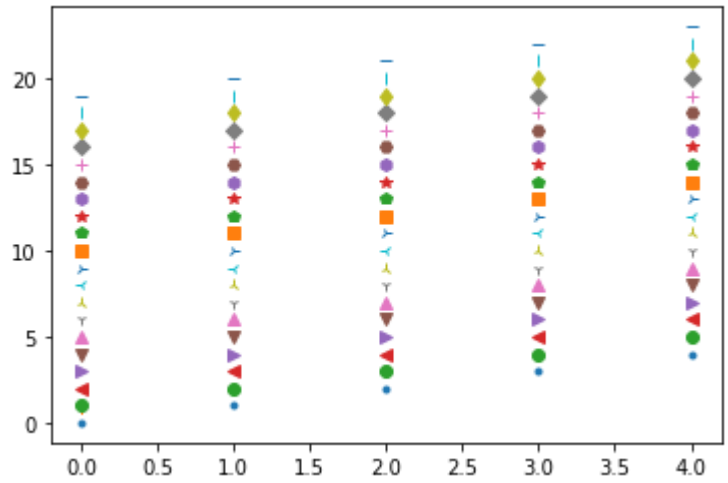
In [3]:

```
# You can customize the line style as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(5)
y = x
plt.plot(x, y, '-', x, y+1, '--', x, y+2, '-.', x, y+3, ':')
plt.show()
```



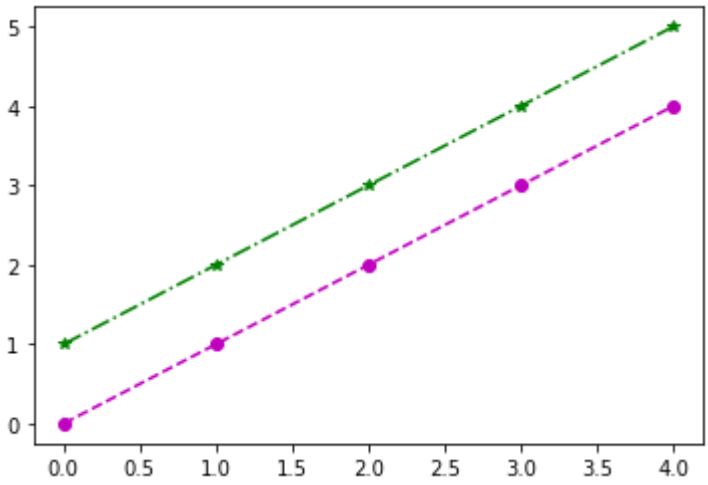
In [4]:

```
# You can even change the markers as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(5)
y = x
plt.plot(x, y, '.')
plt.plot(x, y+0.5, ',')
plt.plot(x, y+1, 'o')
plt.plot(x, y+2, '<')
plt.plot(x, y+3, '>')
plt.plot(x, y+4, 'v')
plt.plot(x, y+5, '^')
plt.plot(x, y+6, '1')
plt.plot(x, y+7, '2')
plt.plot(x, y+8, '3')
plt.plot(x, y+9, '4')
plt.plot(x, y+10, 's')
plt.plot(x, y+11, 'p')
plt.plot(x, y+12, '*')
plt.plot(x, y+13, 'h')
plt.plot(x, y+14, 'H')
plt.plot(x, y+15, '+')
plt.plot(x, y+16, 'D')
plt.plot(x, y+17, 'd')
plt.plot(x, y+18, '|')
plt.plot(x, y+19, '_')
plt.show()
```



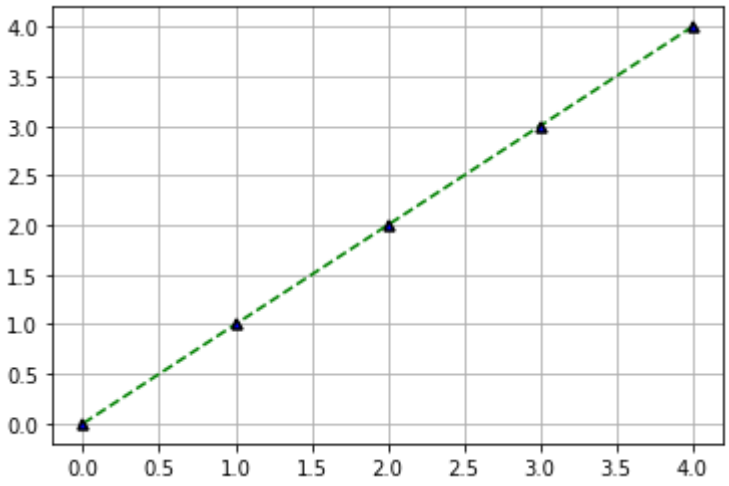
In [5]:

```
# You can combine all three techniques
# (for colors, markers, and line styles) to
# customize the visualization as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(5)
y = x
plt.plot(x, y, 'mo--')
plt.plot(x, y+1 , 'g*-.')
plt.show()
```



In [6]:

```
# These are the basic customizations you can do in Matplotlib. You can customize
# everything in great detail. Here is a code example:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(5)
y = x
plt.plot(x, y, color='g', linestyle='--', linewidth=1.5,
marker='^', markerfacecolor='b', markeredgecolor='k',
markeredgewidth=1.5, markersize=5)
plt.grid(True)
plt.show()
```



In [7]:

```
# You can even customize the values on the x- and y-axes as follows:

%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
x = y = np.arange(10)
plt.plot(x, y, 'o--')
plt.xticks(range(len(x)), ['a', 'b', 'c', 'd', 'e', 'f',
                           'g', 'h', 'i', 'j'])
plt.yticks(range(0, 10, 1))
plt.show()
```