**Adding some style**

Previously, we learned how to create line plots and add labels, legends, and text. Now, we're going to make those plots beautiful.

**And miles to go**

After his success in graphing productivity, Officer Deshaun wants to plot the miles driven by each office over the course of a day. He's including more officers in this plot, so it's becoming more difficult to read. Officer Deshaun will improve is plot by customizing the the style of each line.

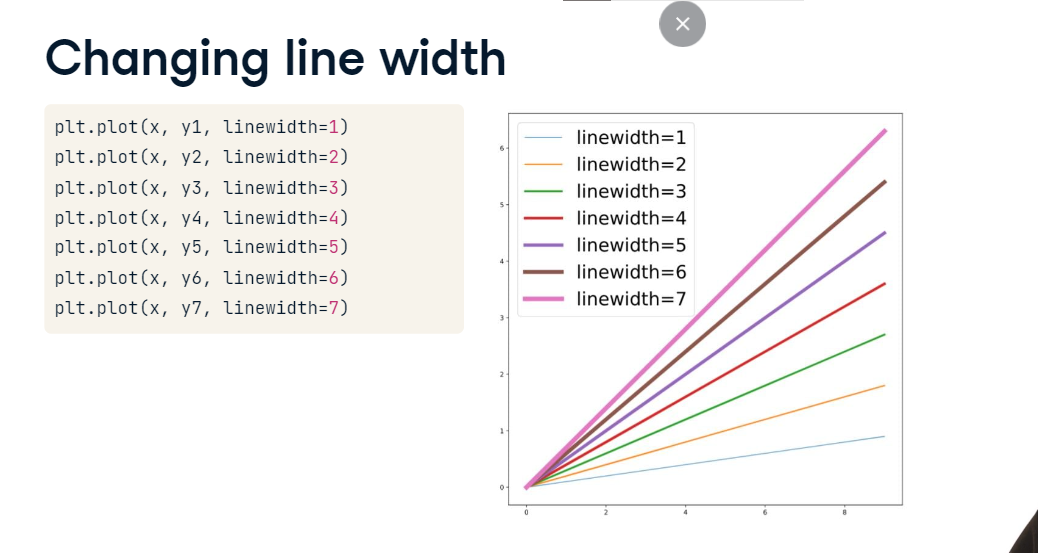
**Changing line color**

The first change Officer Deshaun might want to make to a line is to change its color. He can do this by adding the keyword argument "color" to the plt-dot-plot command. In this case, we plot six different lines in six different colors. The color keyword will accept a string corresponding to a "web color". For a list of allowed color names, visit Wikipedia and look up "web colors". In this example, we plot six different lines, each with a different color. You can't see the code, but we've added a legend so that you can see what each color appears on our plot.



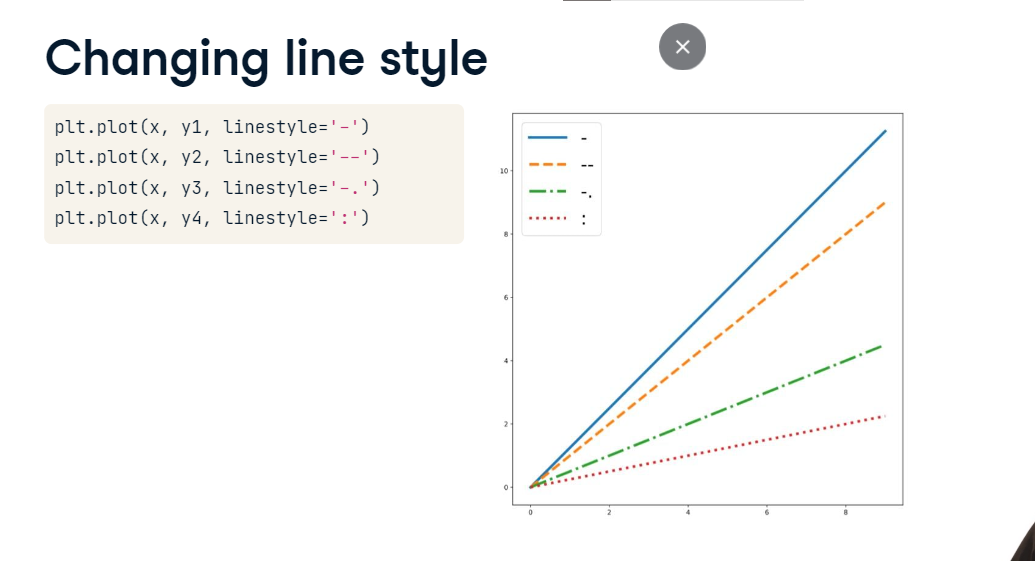
**Changing line width**

Officer Deshaun can also modify the width of a line. The default line width is 1, but you can increase it using the keyword argument linewidth. In this example, Officer Deshaun plotted seven dataset, each with a slightly thicker line than the previous one. Again, we've hidden the extra code, but are showing a legend so that you can see what each linewidth looks like. Increasing line width can make graphs easier to use or can emphasize one line in a multi-line plot.



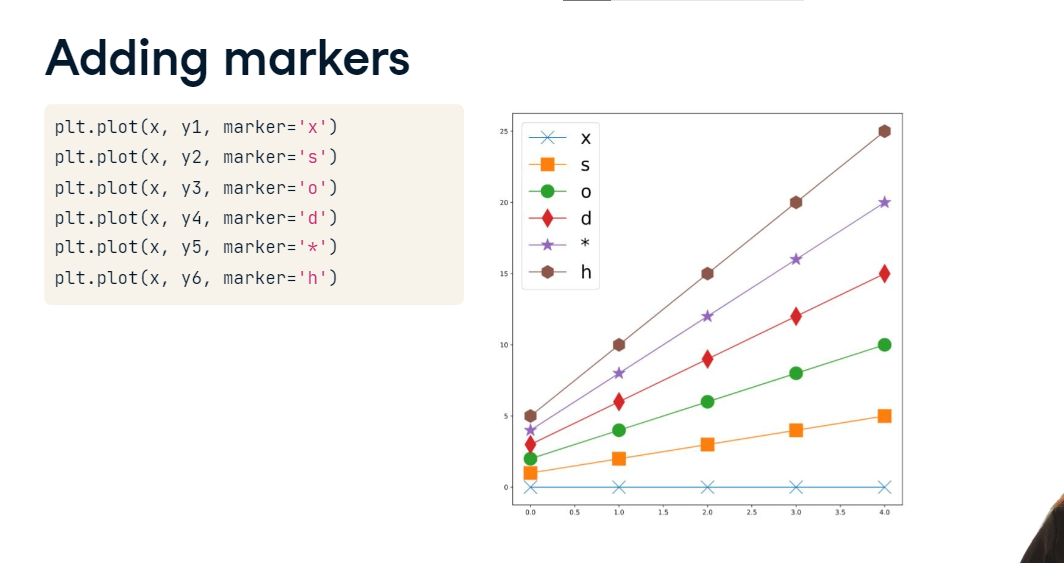
**Changing line style**

Officer Deshaun can further modify a line by changing its style. He can create different types of dashed lines using the keyword argument linestyle. Linestyle accepts several strings which correspond to different types of dashing. In this example, we plot four datasets with four different values of linestyle: a single hyphen for a normal line, two hyphens for a dashed line, a hyphen followed by a dot for a dot/dash line, and colon for a dotted line.



**Adding markers**

Adding markers is a great way to distinguish between different lines or to emphasize the location of data points. Officer Deshaun can add a marker using the keyword marker, which accepts several different strings. For example, if he sets marker equal to the letter s, the plot will have square markers. If he sets marker equal to the letter d, the plot will have diamond-shaped markers. In this example, we plot six different datasets, each with a different marker. Again, we hid the legend code but you can use it to see which symbols make each type of marker.



**Setting a style**

Sometimes, we want to make more drastic change to our graph. We can change the background, colors, and fonts for our entire graph by setting a style. The function plt-dot-style-dot-use accepts several different strings which correspond to different plotting styles. Here we show the same graph plotted in four different styles: fivethirtyeight (from the famous news site), ggplot (from another plotting library), seaborn (from yet another plotting library), and default (which is the normal style in matplotlib). Which do you think is best?

