**Customization**

Wow, now you've got histograms under your datavis belt, let's figure out how to customize our plots. Creating a plot is one thing. Making the correct plot, that makes the message very clear -- that's the real challenge.

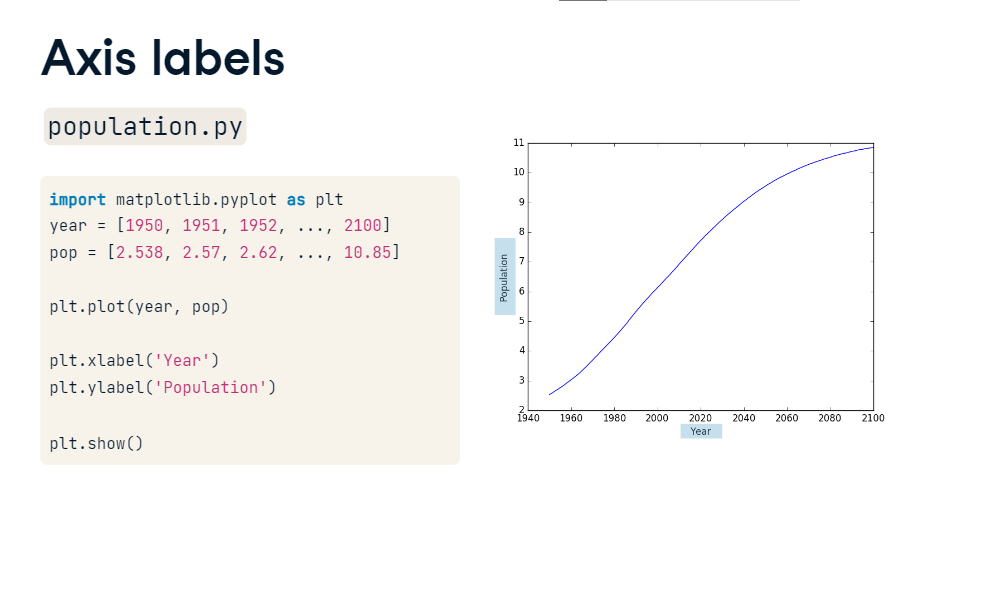
**Data visualization**

For each visualization, you have many options. First of all, there are the different plot types. And for each plot, you can do an infinite number of customizations. You can change colors, shapes, labels, axes, and so on. The choice depends on: one, the data, and two, the story you want to tell with this data. Since there are so many possible customizations, the best way to learn this is by example.

**Basic plot**

Let's start with the code in this script to build a simple line plot. It's similar to the line plot we've created in the first video, but this time the year and pop lists contain more data, including projections until the year 2100, forecasted by the United Nations. If we run this script, we already get a pretty nice plot: it shows that the population explosion that's going on will have slowed down by the end of the century. But some things can be improved. First, it should be clearer which data we are displaying, especially to people who are seeing the graph for the first time. And second, the plot really needs to draw the attention to the population explosion.

**Axis labels**

The first thing you always need to do is label your axes. Let's do this by adding the xlabel and ylabel functions. As inputs, we pass strings that should be placed alongside the axes. Make sure to call these functions before calling the show function, otherwise your customizations will not be displayed. If we run the script again, this time the axes are annotated. Fantastic! 

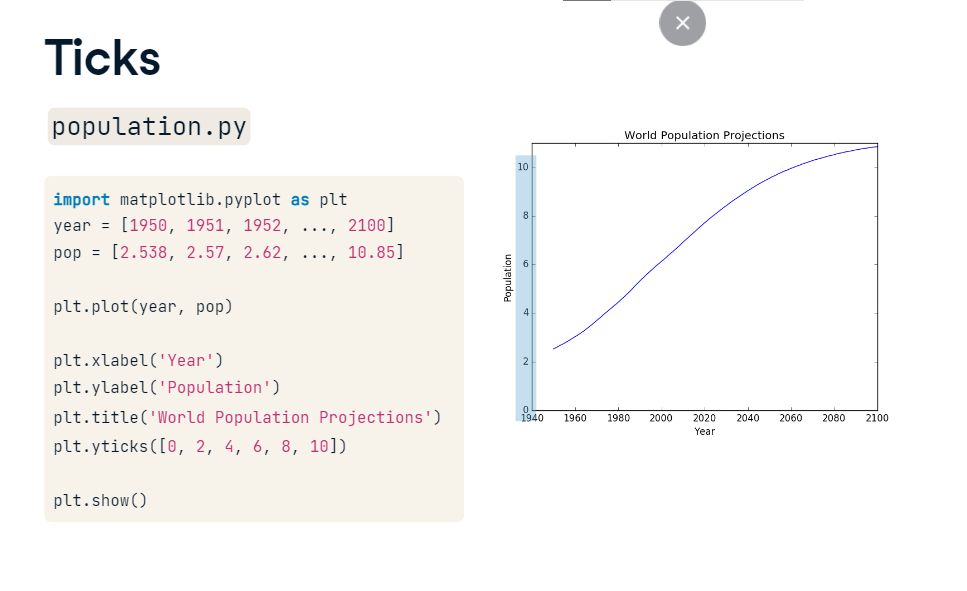
**Title**

We're also going to add a title to our plot, with the title function. We pass the actual title, 'World Population Projections', as an argument.

**Title**

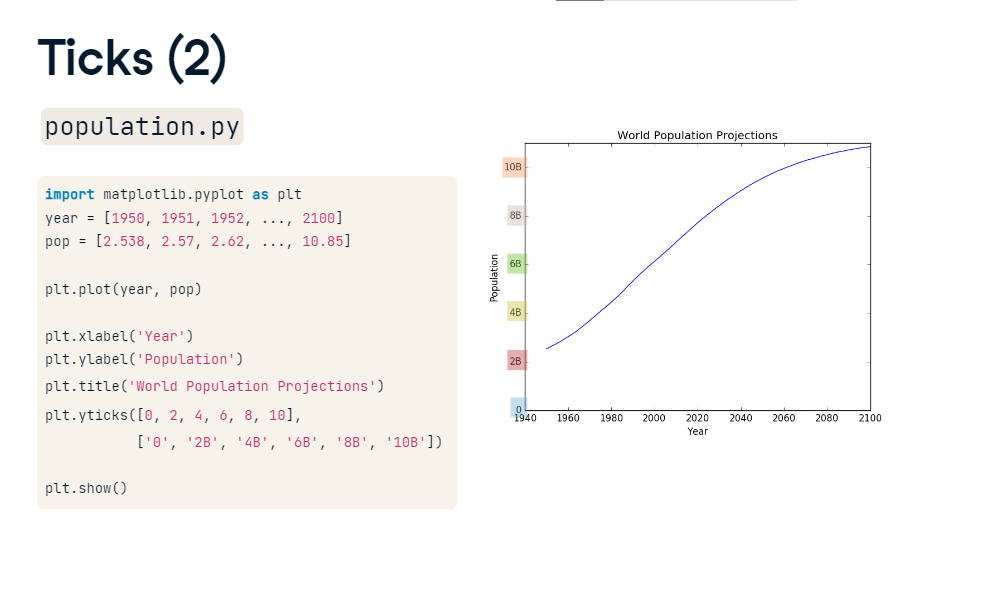
And there's the title! So, using xlabel, ylabel and title, we can give the reader more information about the data on the plot: now they can at least tell what the plot is about. To put the population growth in perspective, I want to have the y-axis start from zero.

You can do this with the yticks function. The first input is a list, in this example with the numbers zero up to ten, with intervals of 2.

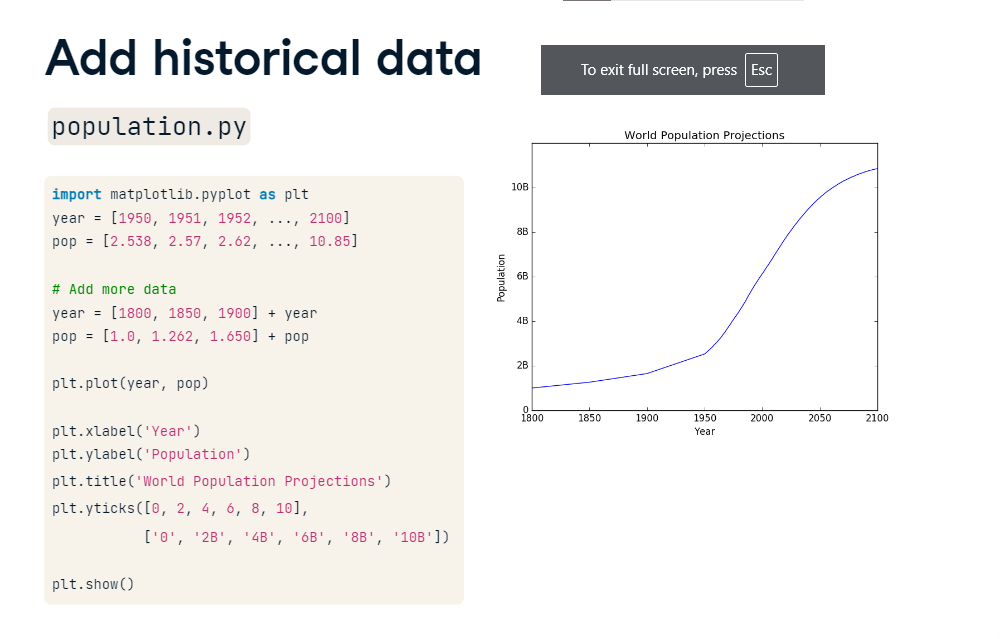
If we run this, the plot will change: the curve shifts up. Now it's clear that already in 1950, there were already about 2.5 billion people on this planet. 

**Ticks (2)**

Next, to make it clear we're talking about billions, we can add a second argument to the yticks function, which is a list with the display names of the ticks. This list should have the same length as the first list. The tick 0 gets the name 0, the tick 2 gets the name 2B, the tick 4 gets the name 4B and so on. By the way, B stands for Billions here. If we run

the labels will change accordingly. Awesome! 

**Add historical data**

Finally, let's add some more historical data to accentuate the population explosion in the last 60 years. On wikipedia, I found the world population data for the years 1800, 1850 and 1900. I can write them in list form and append them to the pop and year lists with the plus sign. If I now run the script once more,three data points are added to the graph, giving a more complete picture. 

**Before vs. after**

Now that's how you turn an average line plot into a visual that has a clear story to tell! Over to you now.

