**Dictionaries, Part 1**

Weeeeeelcome back. Now do I have a treat for you. In this video, I'll be talking about a new Python type: the dictionary. Dictionaries are sooooo useful in the data world and to see this, imagine the following: you work for the World Bank and want to keep track of the population in each country.

**List**

You can put the populations in a list. You start with Afghanistan, 30.55 million, Albania, 2.77 million, Algeria, around 40 million, and so on. To keep track about which population belongs to which country, you can create a second list, with the names of the countries in the same order as the populations. Now suppose that want to get the population of Albania. First, you have to figure out where in the list Albania is, so that you can use this position to get the correct population. Remember about the method index(), a list method you learned about in the Intro to Python course? Let's use it to get the index of Albania in countries, like this. Now, we can use this index to subset the pop list, to get the population corresponding to Albania. As expected, we get 2.77, the population of Albania in millions. So we built two lists, and used the index to connect corresponding elements in both lists. It worked, but it's a pretty terrible approach: it's not convenient and not intuitive. Wouldn't it be easier if we had a way to connect each country directly to its population, without using an index? This is where the dictionary comes into play.



**Dictionary**

Let's convert this population data to a dictionary. I included the lists to start from on the top here, so you can see what's going on. To create the dictionary, you need curly brackets. Next, inside the curly brackets, you have a bunch of what are called key:value pairs. In our case, the keys are the country names, and the values are the corresponding populations.

**Dictionary**

The first key is Afghanistan, and its corresponding value is 30.55. Notice the colon that separates the key and value here. Let's do the same thing for the two other key-value pairs, and store the dictionary under the name world.

If you know want to find the population for Albania, you simply type world, and then the string Albania inside square brackets. In other words, you pass the key in square brackets, and you get the corresponding value. The key opens the door to the value: pretty poetic, isn't it? This approach is not only intuitive, it's also very efficient, because Python can make the lookup of these keys very fast, even for huge dictionaries.

