**Python Dictionaries**

[00:00:00.00] [AUDIO LOGO]

[00:00:08.56] RYAN AHMED: Hello, everyone and welcome to this lesson on Python dictionaries. In Python, dictionaries consist of a collection of key value pairs that are unordered. The keys in Python dictionaries are unique. Note that the way we access elements in dictionaries is different from lists. Dictionary elements are accessed using their key, however, in Python lists, elements are accessed by their position in the list via indexing.

[00:00:38.63] Here are the key learning objectives of this lesson. Define Python dictionaries and understand their use cases. Access specific elements in Python dictionaries using their keys. Add new items and delete items from the dictionary. So let's head over to our Jupyter Notebook and get started.

[00:00:59.06] [AUDIO LOGO]

[00:01:05.91] All right, so right now we are in the Jupyter Notebook titled Python Dictionary. OK, so what is a Python dictionary? A Python dictionary consists of a collection of key value pairs that are unordered. Simply, each key value pair maps the key to its corresponding value dictionary elements are accessed using their specific keys, and please note that in Python dictionaries, keys must be unique while no requirements are placed on values.

[00:01:41.33] So let me show you a practical example. Let's assume that we would like to define a Python dictionary that contains portfolio investments for a given investor. A portfolio is a collection of assets that could be in the form of equity, fixed income, derivatives, or alternative investments. If you recall, equity investment simply allows investors to hold a portion of ownership of a company such as stocks, fixed income, securities such as bonds, and the fixed amount of cash flow at given dates. While derivative securities, this is a form of security with a price derived from an underlying asset, such as options.

[00:02:29.55] So for example, let's assume that I would like to list equity, fixed income, and derivatives for a given investor using Python dictionary. That's how you write or define a Python dictionary. First, you give the dictionary a name. So you say investor\_assets equals to. You define the Python dictionary using curly braces, so these are the curly braces that declares or defines a Python dictionary. And then what you do is you specify a key and then you specify the value, and that will be the first item in a Python dictionary.

[00:03:06.77] For example, I'm assuming that my investor has equity of $5,000. So I'm going to specify a key, call it equity, and then I'm going to add colon, and then I'm going to list the value associated with that key. So the investor has $5,000 invested in equity. In item two, I'm assuming that the investor also has fixed income as well, investments. So in form of bonds, as an example.

[00:03:35.94] So I'm going to say fixed income, colon, 1,000. So this is the key and this is the value associated with that key. And then I'm assuming also that my investor has derivatives and maybe has $300 invested in the derivatives. So I'm going to say derivatives, colon, and then put 300 in it. Again, here I have the key and I have value associated with that key. To access specific elements in a Python dictionary, that's the syntax that you specify. You say investor\_assets.

[00:04:07.52] So you specify the name of the Python dictionary, you open square brackets, you add quotation marks, and then you specify the key that you're interested in. For example, here I'm saying I'm interested in the value associated with the fixed income key. So I'm going to say fixed income, and then the output from that line of code. I'm going to get a thousand, and that would be the value associated with the fixed income key.

[00:04:37.94] Remember that accessing elements in a dictionary is fundamentally different compared to accessing elements in Python lists. If you recall in the previous couple of lessons that in lists you can access elements by the position in the list using indexing. For example, I had the first element with an index of zero and then one, two, and three, if you recall. While in dictionaries right now, you can access elements using the keys. So we don't have an index here. You access elements of dictionaries using the key and you get the value associated with that key.

[00:05:14.67] So let's go ahead and get started with our code. All right, so let's go ahead and define our investor\_assets dictionary. So here I'm going to say investor\_assets equals to. I'm going to open curly braces, and then I'm going to define the key and value pairs. So I'm going to say equity, colon, 5,000, comma, and that will be the first element or item that we have in my Python dictionary. And then I'm going to say fixed income, colon, 1,000, and then derivatives, colon, 300.

[00:05:47.73] And then I'm going to print out my Python dictionary, which is investor\_assets. If you press Shift and Enter, here we go. Here you have been able to define your first Python dictionary. Well, what if I wanted to perform a sanity check and check out the data type of my Python dictionary? If I say type, a open parentheses, and then specify the name of my Python dictionary, which is investor\_assets, you press Shift-Enter. Here we go. You will end up with dict, indicating that this is a dictionary data type. OK.

[00:06:23.54] Next, what if I wanted to access specific elements within a Python dictionary? Well, as I mentioned in the introductory part of this lesson, is you can access elements in a Python dictionary using their keys. So here I'm going to say investor\_assets. You open the square brackets. You add quotation marks, and then you specify the key. For example, here I'm saying I wanted to grab the value associated with my key, which is fixed income. If you press Shift-Enter, here we go. Now you end up with 1,000, and that was the value associated with fixed income key. OK?

[00:07:01.65] All right. Let's assume that we would like to maybe add an additional element to a predefined Python dictionary. For example, let's assume that my investor decided to add alternative investments, for example, to his or her portfolio. And alternative investments, these are nontraditional investments such as real estate and commodities. So here, to add a specific element or new element you should say investor\_assets. That's the name of the pre-existing Python dictionary that we defined before.

[00:07:37.74] You open the square brackets. You specify the new key. So this is the new key that I'm going to include, and that is going to be alternative investments. And then you say equals to, and then assign a value to that new key. So if you press Shift-Enter right now, here we go. You will see that we have been able to update my Python dictionary with a new element in here, a new key value pair.

[00:08:03.67] Well, what if you'd like to remove an item from a Python dictionary? All you need to do is to say delete, del, specify the item that you would like to delete. So here I'm going to say investor\_assets of fixed income, and that is going to delete the fixed income for me. You press Shift and Enter, you'll end up with the equity, derivatives, and alternative investments, and now we have been able to delete the fixed income from my Python dictionary. OK? All right.

[00:08:32.05] So that's it. That's all I have for this lesson. I hope you enjoyed it. In the next lesson, I'm going to walk you through our practice opportunities solution, and I want you to please go ahead, give it a shot, and write your solution here in this code sense, and then you can compare your answers to mine at the end of the notebook. Please stay tuned, best of luck, and I'll see you in the next lesson.

[00:08:54.49] [AUDIO LOGO]