**For Loops**

[00:00:00.00] [MUSIC PLAYING]

[00:00:07.90] RYAN AHMED: Hello, everyone, and welcome to this lesson on for loops. In Python, loops are powerful, because they allow coders to shorten what could have been thousands of lines of code to just a few lines instead. Loops allow us to write a block of code once and then repeat it as many times as needed. In this course, we will learn how to write for loops and while loops and understand their use case. We will use them later in the final capstone project, so let's get started with for loops.

[00:00:39.75] For loops are used to iterate over a sequence, such as a list or a dictionary. Here are the key learning objectives of this lesson, understand the syntax of for loops in Python, write for loops that iterates over Python lists, use for loops to iterate over characters of a Python string. So let's head over to our Jupyter notebook and get started.

[00:01:09.47] All right, so right now, we are in the Jupyter notebook titled For Loops. So for loops in Python are used to repeat a block of code for a fixed number of times. They are simply used to iterate over a sequence, which could be a list or a dictionary. Let me show you an example.

[00:01:30.42] Let's assume that I have a list titled Company Names. This list contains four different companies, company A, B, C, and D. And let's assume that I would like to go through every element or item in my list, and I would like to print it out on the screen.

[00:01:51.92] For example, I would like to go and print out company A, and then company B, and then C, and then D. Simply, I would like to print that output that you see here on the right hand side. So one way of doing that is to just simply say print company A, and then I need to repeat it, again, and then say print company B, company C, and company D. And you have to repeat that multiple times, which could require you to write many, many lines of code to print every item that you have in your list.

[00:02:26.40] Alternatively, what you could do is you can simply leverage loops. So for loops comes into play to handle this and simply to repeat a block of code for a fixed number of times, and it works really well, specifically with lists and dictionaries as well. So let me show you the syntax of for loops, and then we're going to jump into the code and show you in details how to write for loops in there.

[00:02:51.20] So this is simply how you write a for loop. You say for. This is the keyword, and then you specify a temporary variable. Here, I called it i. You call it x, y, or any variable you want. And then you see in. Then you specify the Python list, or a dictionary, or what we call the iterable that you would like to repeat on.

[00:03:13.25] For example, here, I selected company names, and please, note that company names here is the Python list that I would like to iterate on. Afterwards, I had a colon here at the end. That indicates the end of for loop, and then, afterwards, I need to include the body of the four.

[00:03:32.81] Essentially, what is the section that I would like to integrate multiple times? So here, please, note that there is an indentation or a space here, white space, and that will indicate body of for loop. And here, all I'm doing is I'm saying, OK, please, go ahead and print i, and that will essentially generate the exact same output that you see here on the right hand side, which could require four lines of code, for example, if you'd like to do it the traditional way and using print company A, print company B, C, and D.

[00:04:06.68] So let me show you in details what's actually happening behind the scenes. So in the first iteration, i is going to go into company names lists and is going to be assigned the first value. So in the first iteration, i is going to have company A. So when I go here and say print i, so I'm going to go and print company A, and that's basically the first output that I'm going to see.

[00:04:32.01] And then what's going to happen is it's going to go up, again, so I'm going to repeat simply for loop one more time. I'm going to grab the second element in my list, and that would be company B. And then I'm going to print i, so I'm going to print company B.

[00:04:47.45] Next, i is going to be assigning the third element in my list. That will be company C, and then I'm going to go print company C. And then, one more time, I'm going to go print company D, and that simply is going to conclude the for loop. And then I'm going to exit the full loop afterwards and execute whatever code that comes next after the for loop is done, OK?

[00:05:09.59] All right, so let me go ahead and shift into the code and show you in details how we can write that in code and what kind of outputs, as well, we could generate in there. All right, so first step, I'm going to define a list. I'm going to call it company underscore names equals to, and as I mentioned before, simply, the list contains four different company names, company A, B, C, and D.

[00:05:34.76] So if you go ahead and execute or run the sim, you just press Shift and Enter on your keyboard, and that is going to simply print out the company names. And this is simply the list. So what I wanted to do right now is I would like to print out all the company names listed in there in that list. So a traditional way of doing that, I can simply repeat the print function several times as you can see here.

[00:06:02.66] I can say print, company names, square brackets of index zero. If you recall, we can simply access elements within the list using their index. And in Python, we simply appreciate the index zero, so it's zero indexed. And that's why we stopped. The first element has an index of zero.

[00:06:24.10] So if I see print company names of zero, that is going to generate or print out company A, and then I need to repeat the print operation again. So I'm going to say print company names of one, and that is going to print company B. And then I'm going to print company names of two, of index two, print company C, and then, finally, print company D.

[00:06:45.52] So if you run or execute the sim, simply, what you get here is all the elements in the list. I was able to print them out to the screen. The only problem is I have to write so many lines of code to do that to do pretty much a very simple task, and that's where for loops or loops, in general, comes into play. Let me show you how we can generate the exact same output in a much more efficient way.

[00:07:11.84] So alternatively, I can use for loops to generate the same output, and all I need to do is to say for. You specify a temporary variable. Here, you can simply call it i. You can call it x. You can call it any name you like, and then you use the keyword in. And then you specify the iterable that you would like to iterate on.

[00:07:35.11] For example, here, I added the list, which was company names. You can add a dictionary, for example, if you would like to iterate on. Please, note that there is a colon here that you need to add at the end of for loop, and there is an indentation or right space that you need to include. And simply, you just say print i.

[00:07:55.74] If you pressed Shift and Enter, here we go. Now, we have been able to generate the exact same output that we have done before, but in a lot more efficient way using for loops. OK, what I could also do is I can simply use for loops, as well, to loop over lists containing integers or floating points.

[00:08:17.40] So, for example, here, that list that I had before contained just simply strings in there. But if I have a list that contains integers or floating points, I can simply do the same thing as well. I can use a for loop with the exact same syntax. Let me show you.

[00:08:35.76] Let's assume that I was able to get the company revenues from the list that I got here before. For example, I was able to get the company A revenue and company B revenue, C and D revenues, and I put them in a separate list. I called it company underscore revenues equals to, and then I put, here, that's the revenue for the first company, company A, and this is the revenue for second company, company B, and so on.

[00:09:04.83] So if you press Shift and Enter, here we go. Please, note that this list here contains just simply integers in there. So what I could do next is I can go ahead and create a for loop, and in this for loop, I don't want to just go and print out the elements of my list to the screen.

[00:09:25.97] Alternatively, I would like to go and, perhaps, sum up all the elements, all the revenues that I have to come up with the total sum of all the revenues for all four companies that I have in my list. To do that, I can simply use for loops to do it. So let me show you. First, I'm going to define an accumulator, and I'm going to initialize it to zero.

[00:09:49.42] Here, I'm going to say total underscore revenue, and then I'm going to put zero. Think of an accumulator as a variable that I'm going to update every single iteration or every single loop, and I'm going to add value to it. So I'm going to initialize it to zero, because I'm doing a sum operation. And then I'm going to say for i, and i, again, is a temporary variable in company revenues, which is the list that I defined here before.

[00:10:20.14] Every time, I'm going to grab an element within the list. So for example, in the first iteration, i is going to be equal to 600,000. In the second iteration, i is going to be 900,000, and the third iteration is going to be a million and so on. So every time I'm going to grab the i element within the list, I'm going to assign it to i, and then I'm going to say, please, grab the total revenue, add i to it, and overwrite the previous total revenue, and keep repeating that loop over and over, again, until you simply sum up all the elements that I have in my list.

[00:11:00.73] So let me walk you through a quick iteration within the for loop. So basically, what's going to happen is, in the first iteration, I'm going to first to find total revenue is going to be equal to zero, because that's outside of my for loop. Next, I'm going to say, OK, for i in company revenues, so in the first iteration, i is going to be equal to 600,000. And then I'm going to sum up total revenue, which is zero, plus i, which is in the first iteration, was equal to 600,000.

[00:11:30.46] So now, zero plus 600,000 is going to be equal to 600,000. So now, total revenue before for loop was zero. After the first iteration, it's going to be 900,000, and then, in the second iteration, i is going to be equals to 900,000.

[00:11:48.07] So now, I'm going to sum up the previous total revenue, which was 600,000 plus the new i, which is 900,000, and then I'm going to get the new total revenue. And I will keep repeating that over and over, again, until I sum up all the elements. So if you press Shift and Enter, here we go.

[00:12:06.85] Now, we ended up with-- essentially, that's 3.6 million, basically, and that's kind of the sum of the total revenues that we have in every single company in my list. Of course, alternatively, you can simply go ahead and say sum of company revenues, and that is going to do the same, essentially, output. But I just wanted to show you how we can sum up all the elements and access elements within a list using for loops, and that was pretty straightforward.

[00:12:37.19] Finally, you can also use for loops to loop over a string and print out every character that you have in the string. For example, if I have a message equals to, welcome to Python programming fundamental scores as an example, and if you recall, this was a string. And a string is simply a series of characters. What I could do is, if I can say for character, and character, again, is a temporary variable within for loop in message. So message here is my string.

[00:13:10.86] Every time, go ahead and print character. So simply, in the first iteration, character is going to be equal to w, so I'm going to print w. And then, in the second iteration, I'm going to grab the second element, so character is going to be equal to e. And then I'm going to print e, and then I'm going to grab the l, and then c, and so on.

[00:13:31.18] So if you press Shift, Enter, here we go. Now, we have been able to simply print every single character that you have in your Python string, and that's simply how you can leverage for loops to loop over a list of strings. You can also loop over a list of integers, and I also showed you how to loop over a string in here, and simply, print every single character that I have in my string.

[00:13:57.34] And that's it. That's simply all I have for this lesson. I hope you enjoyed it. In the next lesson, we're going to have our practice opportunity. Please, stay tuned, best of luck, and I'll see you in the next lesson.

[00:14:07.48] [MUSIC PLAYING]