**Python Strings**

[00:00:00.00] [MUSIC PLAYING]

[00:00:08.76] RYAN AHMED: Hello, everyone, and welcome to this lesson on Python strings. In Python, a string is a sequence of characters that are being closed by single or double quotation marks. Recall that we touched base on strings before when we introduced the print function. Here are the learning objectives of this lesson.

[00:00:29.37] Define Python string and understand their use case. Concatenate two or more strings together. Apply string methods such as split to split strings into a series of words, and upper to convert strings from lowercase to uppercase. So let's head over to our Jupyter notebook and get started.

[00:00:50.41] [MUSIC PLAYING]

[00:00:57.65] All right. So right now we are in the Jupyter notebook titled Python Strings. So in Python, a string is a sequence of characters. Strings are simply enclosed by single or double quotation marks. Here is the syntax of defining a simple Python string simply you specify the name of the string. Here I call it welcome\_message as an example. And then you say equals, and then you add here quotation marks. And here I used double quotation marks.

[00:01:30.54] You can, of course, use single quotation marks and it's going to do the same job for you. And then you specify what you want to print here as part of that string, and that is going to be Hello and Welcome to Python Programming Fundamentals Course. So this is simply the string content and then this is the syntax to define a simple string using double quotation marks.

[00:01:55.93] So let's go ahead and move into the code and show you in details how we can define our first string and also learn how to apply string methods as well. So here, again, I specified welcome\_message, and then I specified the string. And if you press Shift and Enter, that is going to run or execute the sim and you should be able to see that this is simply my Python string.

[00:02:24.09] If you wanted to perform a sanity check and obtain the data type for that string all you need to do is to say type, you open parentheses, and then you specify the name of the string. Here I called it welcome\_message. I'm just going to write welcome\_message in here. You press your Enter. Here we go. And you'll get the data type, and that is going to be STR, which stands for string.

[00:02:47.81] Once you have your data in the form of a Python string, it unlocks a world of possibilities for you. For example, you can apply multiple methods on Python strings. For instance, I can go and say, well, I want to grab that string here and I want it to convert all of its characters into uppercase. To do that you just say welcome\_message message and then you say .upper and you open np parentheses here.

[00:03:18.35] And you press your Enter. Here we go. You will see that you have been able to convert that simply message into uppercase. What if you grab your string and you apply the split method on it? So if you grab the welcome\_message.split and you open parentheses, the split method is going to divide up the string into a bunch of words. And we're going to place these words in a Python list.

[00:03:51.12] If you recall, we covered the Python list in the past, and Python lists are defined with a square bracket. So if you press Shift and Enter, here we go. Basically what I've done here as if I said, grab my Python string, split it into simple words, and put them in a Python list. So the list is going to be called Words.

[00:04:21.38] And here I have the first word, and this is the Hello, and then comma, and then I have the and, comma, and then Welcome to Python Programming Fundamental Scores. And please note that this operation is quite useful-- actually pretty important as well, especially if you're doing sentiment analysis on data. For example, you can get feedback from users regarding the service, and then you can take the message in the form of a string and then you split it into a series of words.

[00:04:53.34] And then you can run an AI or machine learning algorithm on it to extract sentiment to get an idea, are my customers happy or not? And if they are not happy, why? Can I get a lot of information just by simply analyzing the words that is getting-- or I'm receiving from my clients? OK. Next, if you wanted to check the data type of Words-- and if you recall, this was a Python list. I can go ahead and confirm that. If you press Shift-Enter, here we go, and you will get it simply that Words is a list or in a list data type, list format.

[00:05:29.03] OK. Next I can go ahead and use the split method as well. And instead of splitting my Python strings based on this space-- so basically what I've done here is I got my Python string and I split it based on the spaces. So if you don't specify anything at all here within parentheses, by default it's going to be split according to the spaces between the words.

[00:05:55.28] Alternatively, you can go ahead and specify which letter, which character you would like to split upon. For example, if I say welcome\_message and then you say .split and then you open parentheses and you specify the and here character and you press Shift-Enter, here we go. What you notice is simply now I have been able to split my string into two words or two sentences.

[00:06:22.20] One of them is going to be Hello, and there was an and here, and basically I ended up with Welcome to Python Programming Fundamental Course. So if you go up, all I've done here as if I said please split on the and, so I ended up with Hello and then Welcome to Python Programming Fundamentals Course. That would be the second item in my list. All right.

[00:06:42.38] Next, what I could do as well is I can go ahead and combine two strings together as follows. First I'm going to define first\_name equals to Sarah, so that would be my first string. And then I'm going to define another string. I'm going to call it last\_name and then I'm going to put David in it. And then if I would like to combine or concatenate these two strings together, I can simply say first\_name plus last\_name, and then whatever outputs I'm going to get out of the concatenated string I'm going to put it in full\_name.

[00:07:16.37] So if you press Shift and Enter right now, here we go. What you notice is now I have Sarah and then David, and both of them are concatenated or combined together in one string. So you can just say simply say the-- or use the positive sign to combine or concatenate two strings together. What you notice here is the first\_name and last\_name have been put just back to back next to each other without adding a space here.

[00:07:42.45] So if I would like to add a space in between I can just simply say first\_name plus, and I can artificially add a space in here, and then I say plus last\_name. So if you press Shift-Enter, here we go. Now we end up with Sarah, space, David, and that's simply all I have for this lesson. I hope you enjoyed it. In the next lesson, I'm going to have our practice opportunity so please stay tuned. I hope you enjoyed this lesson, and see you in the next one.

[00:08:12.38] [MUSIC PLAYING]