Using Strings

by Sophia

WHAT'S COVERED

In this lesson, you will learn about strings and how to use them. Specifically, this lesson covers:

1. More About Strings

In many programming languages, variables are statically typed. This means that a variable is initially declared to have a specific data type, and any value that we later assign to that variable needs to be of that same type. Python, on the other hand, is dynamically typed. The variable doesn't need to be declared to have a particular type, and we even have the ability to change the data type once it has been set. You can have a variable declared as one type, and then later set it to a value of a completely different type.

TRY IT

Directions: For example, copy this code and run it.

```
myVar=1
print(myVar)
myVar="I am now a String"
print(myVar)
```

You should see the following output: `myVar` originally declared as an integer, then switched to a string.

```
1
I am now a String
```

Recall that a String is a string of text. This can come in many different forms:

A string that's an entire sentence	"Hello, world!"
A string that's just one character	"a"
A string with symbols and numbers	"! 3&*# 29"
An empty string	""

KEY CONCEPT

An **empty string** (“”) is a string that is identified but does not contain anything; it is essentially empty.

Also recall that the use of literal strings are surrounded by quotation marks. You can use single quotes (') or double quotes (") when you define strings, like the following:

EXAMPLE

```
myString = 'Hello, world!'
print(myString)
myString = "Hello, world!"
print(myString)
```

The output screen should look the same.

```
Hello, world!
Hello, world!
```

But take care not to mismatch them, as that will not work and will raise an error.

```
myString = "Hello, world!'
print(myString)
```

Then we see an error.

```
  File "/home/main.py", line 1
    myString = "Hello, world!'
               ^
SyntaxError: unterminated string literal (detected at line 1)
```

Being able to switch them can be useful, especially if you want to have one of the quotes as part of the string. Observe what happens if we use a single quote for the string and have a single quote in the string as well:

TRY IT

Directions: Try running the code below.

```
myString1 = 'It's not working'
print(myString1)
```
We see an error.

```
  File "/home/main.py", line 1
    myString1 = 'It's not working'
                                 ^
SyntaxError: unterminated string literal (detected at line 1)
```
That’s not working quite right. The syntax error is pointing at the “s” or the character to the right of the issue. One approach is to switch the outer quotes to double quotes:

TRY IT

Directions: Try running the updated code below.

```
myString1 = "It's working"
print(myString1)
```
Now the output screen should look correct.

```
It's working
```

TERM TO KNOW

Empty String
An empty string is a string that is identified but does not contain anything; it is essentially empty.

---

2. Escaping Strings

The previous example is not an ideal approach, as you could have a string that has both single quotes and double quotes in it.

EXAMPLE
Here is another example of a string using both types of quotes.

```
myDirections = "When you've finished, press the "submit" button"
```
Using single or double quotes to define a string won't work for this example because we have both single (you're) and double quotes.

Rather, the better method is to escape the characters. In doing so, we can use any characters that would generally raise an error. The **escape character** is a backslash \ followed by the character that we want to add. Using the same example from above, we'll add the backslash in front of the single quote:

EXAMPLE

```
myString1 = 'It\'s working'
print(myString1)
```

Here is the output screen.

```
It's working
```

Success with no errors! There are other important escape characters that you should be aware of, like the newline and tab that can be placed in a string.

EXAMPLE

Instead of having these in separate print statements as shown below.

```
print("First Line.")
print("Second Line.")
print("Third Line.")
```

With output like this.

```
First Line.
Second Line.
Third Line.
```

They can be combined in a single print statement using the `\n` for the new line, like this:

```
print("First Line.\nSecond Line.\nThird Line.")
```

Output screen looks similar.

```
First Line.
Second Line.
Third Line.
```

TRY IT

Directions: Try some of these other escape characters to see them in action in the IDE:

Escape Character	Description	Sample Code
\	Placing the \ at the end of a line, the backslash and newline are ignored.	print("line1 \line2 \line3")
\\	Outputs the backslash.	print("\\")
\'	Outputs the single quote.	print("\'")
\"	Outputs the double quote.	print("\"")
\b	The \b stands for backspace and removes the character prior to the \b.	print("Hi\b World!")
\n	The \n adds a new line after each \n.	print("First Line.\nSecond Line.\nThird Line.")
\t	The \t adds a TAB.	print("Hello\t World!")

TERM TO KNOW

Escape Character
Special characters that when added to a string will allow quotes of a string to be escaped.

3. Assigning Multiple Variables

Note that you can also assign many values to multiple variables on a single line. This can be useful in some cases to save some time rather than having them on separate lines.

TRY IT

Directions: Try entering the following code:

```
myString1, myString2, myString3 = "Python", "is", "Fun"
print(myString1)
print(myString2)
print(myString3)
```

You should see the output screen "Python" "is" "Fun", each word on a separate line.

```
Python
is
Fun
```

On the flip side, you can also have it such that all of the variables are assigned to a single value.

Directions: Try entering the following code:

```
myString1 = myString2 = myString3 = "We're the same"
print(myString1)
print(myString2)
print(myString3)
```

You should see the output screen "We're the same" on three separate lines.

```
We're the same
We're the same
We're the same
```

SUMMARY

In this lesson, we learned **more about strings** and how literal string values should use quotes around them. We learned how combining single and double quotes can create syntax issues. We identified that there may be times we need both single and double quotes in a string and in that case, we can **escape strings** by using the escape character. Finally, we learned how to **assign multiple variables** to strings and had a chance to try some out.

Best of luck in your learning!

TERMS TO KNOW

Empty String
An empty string ("") is a string that is identified but does not contain anything; it is essentially empty.

Escape Character
Special characters that when added to a string will allow quotes of a string to be escaped.