String indices and slices

In security, there are a variety
of reasons we might need to search through a string.
For example, we might need to
locate a username in a security log.
Or, if we learn that
a certain IP address is associated with malware,
we might search for this address in a network log.
And, the first step in being able to use Python in
these ways is learning
about the index of characters in a string.
The index is a number assigned to
every element in a sequence that indicates its position.
In this video, we are discussing strings.
So, the index is
the position of each character in a string.

Let's start with the string "HELLO."
Every character in the string is assigned an index.
In Python, we start counting indices from 0.
So, the character "H" has an index of 0,
and "E" has an index of 1, and so on.
Let's take this into Python and practice using indices.

Placing an index in square brackets after
a string returns the character at that index.
Let's place the index 1 in
square brackets after "HELLO" and run it.
This returned the character "E."
Remember, indices start at 0,
so an index of 1 isn't the first character in the word.

But what if we want it to
return more than just one character?
We can extract a larger part of
a string by specifying a set of indices.
This is called a slice.
When taking a slice from a string,
we specify where the slice
starts and where the slice ends.
So we provide two indices.
The first index is the beginning,
which is included in the output.

The second index is the end,
but it's not included in the final output.
Instead, Python stops
the slice at the element before the second index.
For example, if we wanted to take the letters E-L-L
from "HELLO,"
we would start the interval from the index 1,
but we'd end before the index 4.

Let's try this example and
extract a slice from a string in Python.
Let's type in the string and take the slice starting
at index 1 and ending before index 4.
Now, let's run the code and examine the output.
There's the slice we wanted.

Now that we know how to describe
the location of a character in a string,
let's learn how to search in a string.
To do this, we need to use the index method.
The index method finds the first occurrence of
the input in a string and returns its location.
Let's practice using the index method in Python.

Let's say we want to use the index method
to find the character "E" in the string "HELLO."
We'll locate the first instance of
the character "E." Let's examine this line in more detail.
After writing the string and the index method,
we use the character we want to
find as the argument of the index method.
Remember, the strings in Python are case-sensitive,
so we need to make sure
we use the appropriate case with the index method.
Let's run this code now.
This returned the number 1.
This is because "E" has an index value of 1.

Now, let's explore an example
where a character repeats multiple times in the string.
Let's try searching for
the character "L." We start with similar code as before,
passing the argument "L" instead of "E" to the index method.
Now, let's run this code and investigate the result.
The result is the index 2.
This tells us that the method only
identified the first occurrence of

the character "L" and not the second.
This is an important detail to
notice when working with the index method.

As a security analyst,
learning how to work with indices
allows you to find certain parts in a string.
For example, if you need to find
the location of the @ symbol in an email,
you can use the index method to find
what you're looking for with one line of code.

Now let's turn our attention
to an important property of the strings.
Have you ever heard the expression
"some things never change"?
It might be said about
the comfortable feeling you have with a good friend,
even when you haven't seen them for a long time.
Well, in Python, we can also say this about strings.
Strings are immutable.
In Python, "immutable" means that it
cannot be changed after it's
created and assigned a value.
Let's break this down with an example.

Let's assign the string "HELLO" to the variable my_string.
Now, if we want to change the character "E" to an "A"
so my_string has the value "HALLO,"
then we might be inclined to use index notation.
But here we get an error.
My_string is immutable, so we cannot
make changes like this. And there you have it!

You've just learned how to index and slice into strings.
You've also seen that strings are immutable.
You cannot reassign characters
after a string has been defined.
Coming up, we'll learn
about list operations and see that
lists can be changed with index notation. Meet you there.

Revision #1
Created 2023-12-23 14:12:58 UTC by naruzkurai
Updated 2023-12-27 11:35:24 UTC by naruzkurai