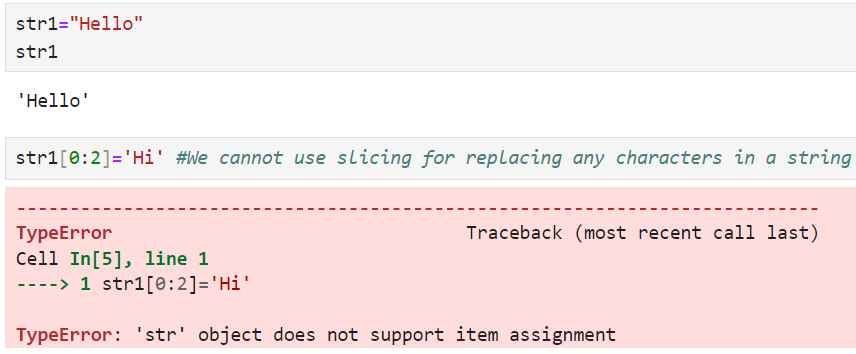
Day: 6

November 7, 2023

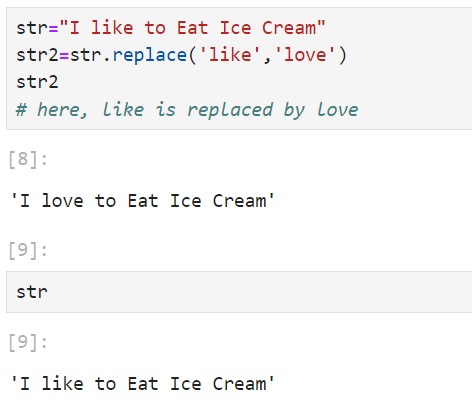
**Update and Delete String**

Strings are immutable which means elements of a string cannot be changed once they are assigned with any values.



Since we cannot update the string, it can be replaced by using following syntax:

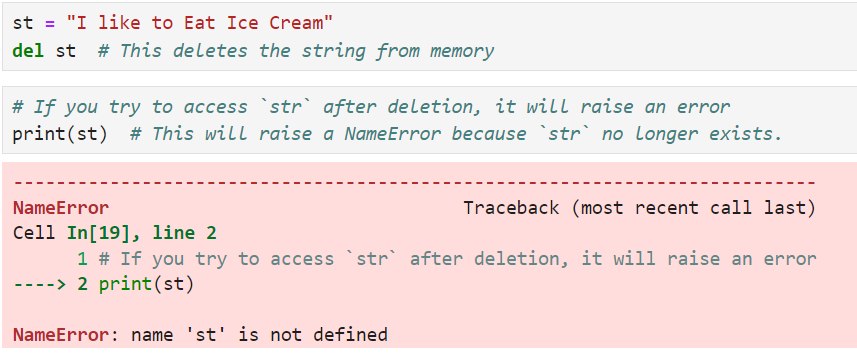
string.replace (‘old value’, ‘new value to replace old value’)



The string "I like to Eat Ice Cream" is still present in the original variable str when it is printed. This is so because strings in Python are immutable, and the replace() method does not change the original string. Rather, it applies the modifications and returns a fresh string.

We can also replace the number but it should be passed as string using single or double quotes.

**Deleting a string:**

In Python, strings are immutable, meaning you cannot directly modify or delete a part of a string. However, you can delete the entire string or remove it from memory using the **del** statement. Here’s how:

**String Concatenation:**

String concatenation means joining 2 or more strings. It is possible to join strings using:

**+ operator:**

****

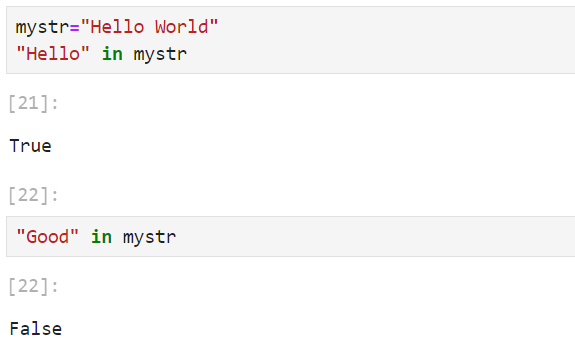
**String Membership:**

String membership in Python allows you to check whether a substring exists within a string using the **in**  and **not in** operators.

**Using in Operator:**

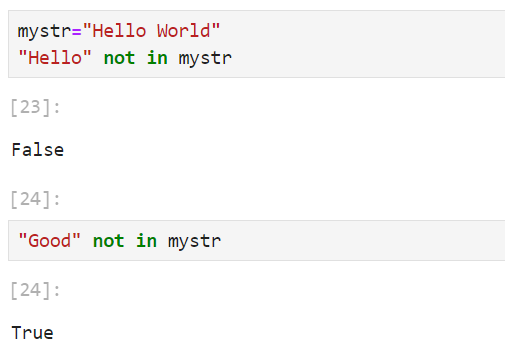
This operator returns True if a substring is present in a string. Otherwise, it returns False

**Example:**



**Using not in Operator:**

Returns True if any substring/character is not part of the string. Otherwise, returns False.



**String Partitioning:**

The partition() method searches for a specified string and splits the string into 3 parts:

* The first element contains the part before the argument string
* The second element contains the argument string
* The third element contains the part after the argument string.

**Example:**

