Strings in Java are immutable (cannot be changed). When a change is made to a String object in Java, a completely new String object is created each time.

**String** in Java is an object that represents a sequence of char values. A String can be created in two ways:

1. Using a literal
2. Using ‘new’ keyword

String str1 = “Welcome”; // Using literal

String str2 = new String(”Welcome”); // Using new keyword

String Pool

Java String pool refers to a collection of Strings that are stored in heap memory. In this, whenever a new object is created, the String pool first checks whether the object is already present in the pool or not.

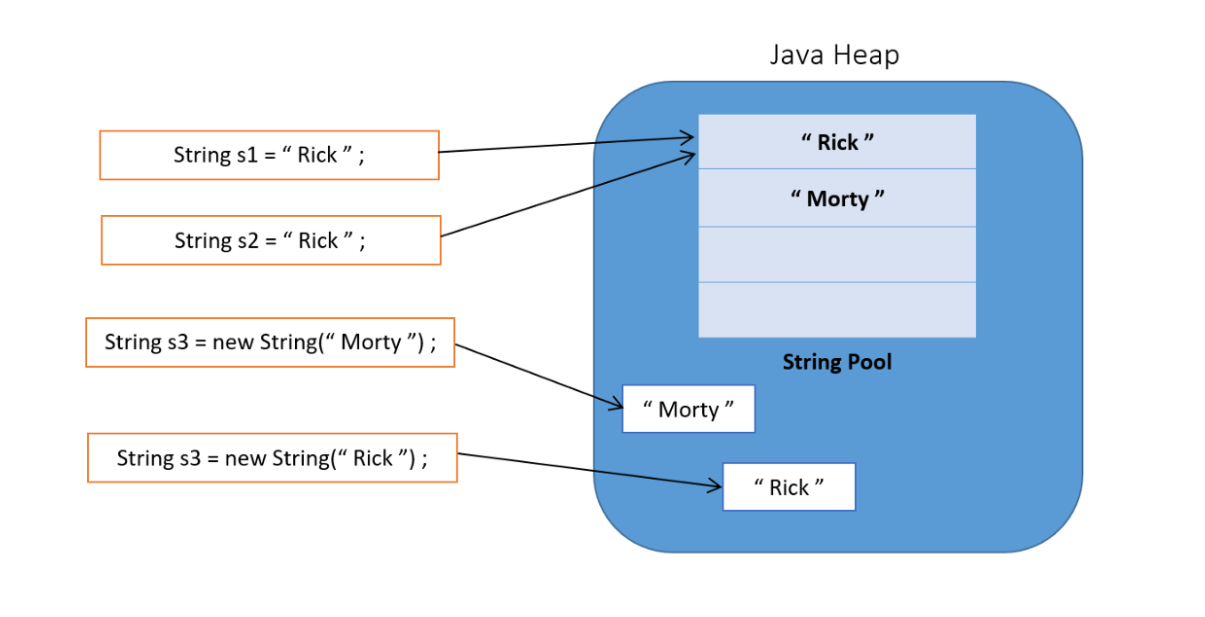
String str1 = "abc";

String str2 = "abc";

System.out.println(str1 == str2);

System.out.println(str1 == "abc");

When using the new operator, a new String object will be created each time in the heap memory. But when using a literal object, if the object already exists in the heap, a new object will not be created, and the reference variable will be pointed to the existing object. When using the String literal, the String object is stored in something called the String Pool in Java. It’s located inside the heap memory, and it helps to save a lot of space for Java Runtime even though it takes more time to create the String. So if you want to create a new String object every time, you should use the new operator to create a String, or if you want to save the heap memory, use the literal string to create a String.



When using the new operator, if the object does not exist in the String Pool, first it will be created in the String pool, and then in the heap memory as well. The created string object reference always points to the heap area object. But if it already exists in the String Pool, then only in the heap memory the String object will be created. This is known to be a Java interview question.

## **What if I changed a string after initializing it?!**

Since they are immutable, the string object will reference another place in the memory and the previous object will be available for the garage collector(if it’s not assigned to another string variable).

String str1 = "Java is fun!";

String str2 = str1;

str2 = "Strings are immutable"; // str2 is no longer points to "Java is fun!"

System.out.println(str1); // Java is fun!

System.out.println(str2); // Strings are immutable

