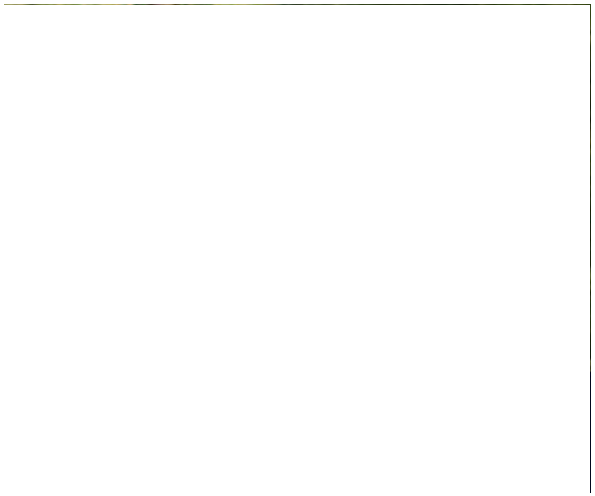
- Collections in Java
- Stream In Java
- Exceptions in Java
- Regular Expressions in Java
- Multithreading in Java
- File Handling in Java with CRUD operations
- Packages In Java

Strings in Java

Strings in Java are Objects that are backed internally by a char array. Since arrays are immutable(cannot grow), Strings are immutable as well. Whenever a change to a String is made, an entirely new String is created.

Below is the basic syntax for declaring a string in **Java programming** language.

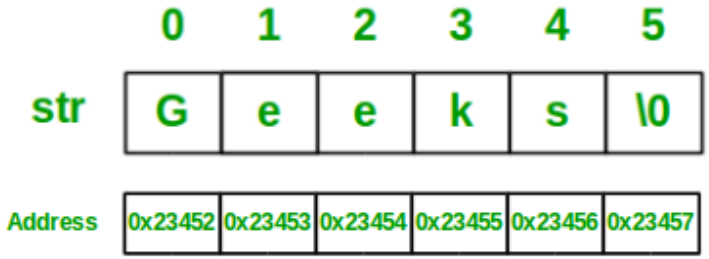
Syntax:



<String_Type> <string_variable> = "<sequence_of_string>";

Example:

```
String str = "Geeks";
```

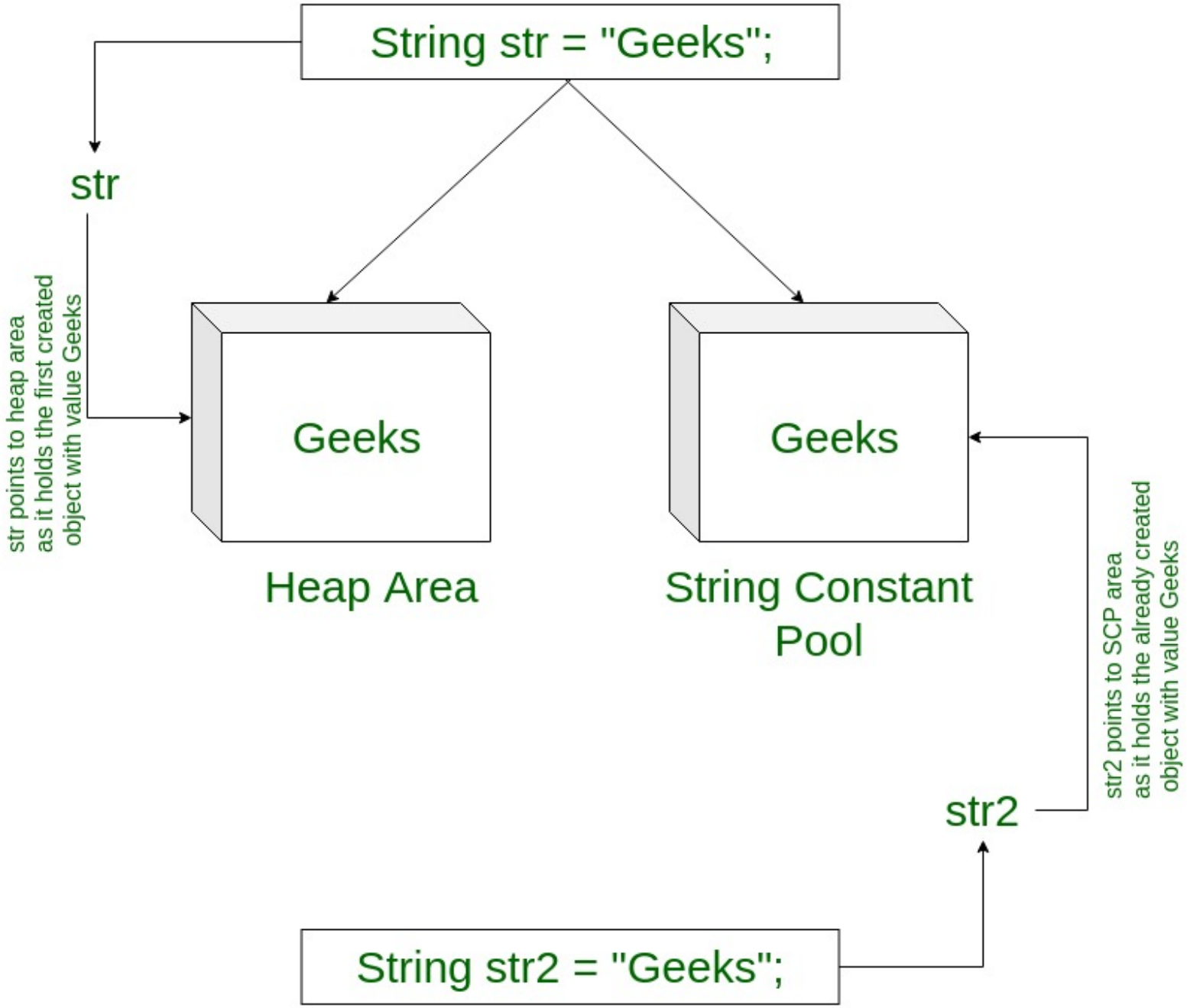


Memory allotment of String

Whenever a String Object is created, two objects will be created- one in the Heap Area and one in the String constant pool and the String object reference always points to heap area object.

For example:

```
String str = "Geeks";
```


String str = "Geeks";
str
str points to heap area
as it holds the first created
object with value Geeks
Geeks
Heap Area
Geeks
String Constant
Pool
str2 points to SCP area
as it holds the already created
object with value Geeks
str2
String str2 = "Geeks";

An Example that shows how to declare String

```
// Java code to illustrate String

import java.io.*;
import java.lang.*;

class Test {
    public static void main(String[] args)
    {
        // Declare String without using new operator
        String s = "GeeksforGeeks";

        // Prints the String.
        System.out.println("String s = " + s);

        // Declare String using new operator
        String s1 = new String("GeeksforGeeks");

        // Prints the String.
        System.out.println("String s1 = " + s1);
    }
}
```

Output:

```
String s = GeeksforGeeks
String s1 = GeeksforGeeks
```

Interfaces and Classes in Strings in Java

- CharBuffer: This class implements the CharSequence interface. This class is used to allow character buffers to be used in place of CharSequences. An example of such usage is the regular-expression package java.util.regex.

- String: String is a sequence of characters. In java, objects of String are immutable which means a constant and cannot be changed once created.

  Creating a String

  There are two ways to create string in Java:

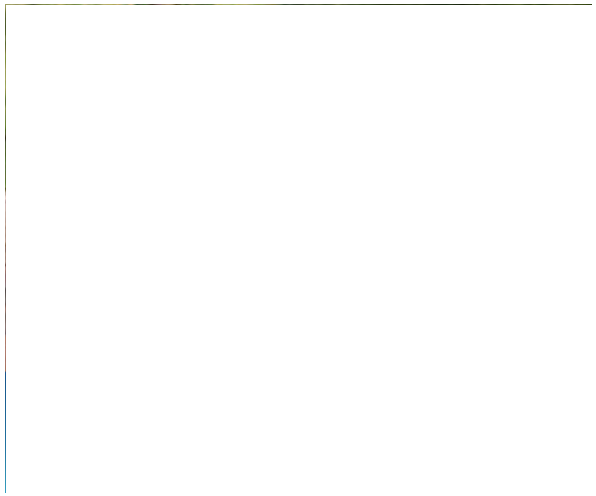
  - ***String literal***

    ```
    String s = “GeeksforGeeks”;
    ```

  - **Using *new* keyword**

    ```
    String s = new String (“GeeksforGeeks”);
    ```

- StringBuffer: **StringBuffer** is a peer class of **String** that provides much of the functionality of strings. String represents fixed-length, immutable character sequences while StringBuffer represents growable and writable character sequences.



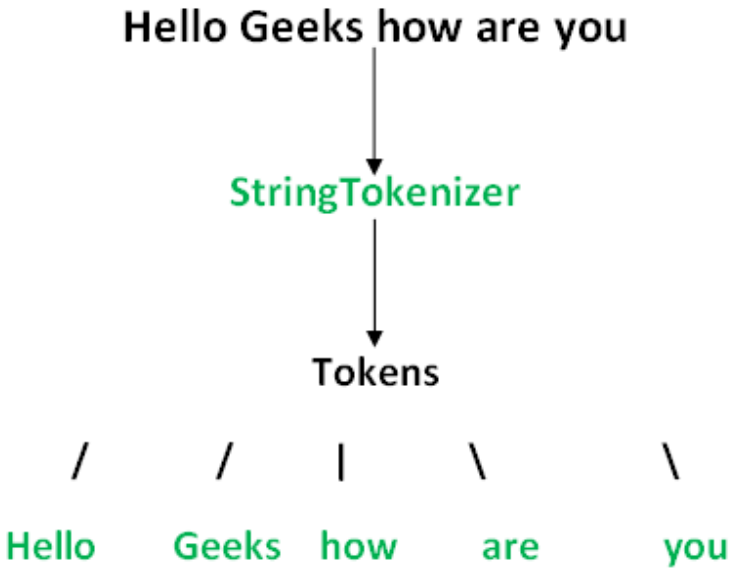
  Syntax:

```
StringBuffer s = new StringBuffer("GeeksforGeeks");
```

- StringBuilder: The **StringBuilder** in Java represents a mutable sequence of characters. Since the String Class in Java creates and immutable sequence of characters, the StringBuilder class provides an alternate to String Class, as it creates a mutable sequence of characters.
  Syntax:

```
StringBuilder str = new StringBuilder();
str.append("GFG");
```

- StringTokenizer: StringTokenizer class in Java is used to break a string into tokens.
  Example:



  A StringTokenizer object internally maintains a current position within the string to be tokenized. Some operations advance this current position past the characters processed.
  A token is returned by taking a substring of the string that was used to create the StringTokenizer object.

- StringJoiner: StringJoiner is a class in *java.util* package which is used to construct a sequence of characters(strings) separated by a delimiter and optionally starting with a supplied prefix and ending with a supplied suffix. Though this can also be with the help of

StringBuilder class to append delimiter after each string, StringJoiner provides an easy way to do that without much code to write.

Syntax:

```
public StringJoiner(CharSequence delimiter)
```

Recommended Posts:

Compare two Strings in Java

Compact Strings in Java 9 with Examples

Numbers in Java (With 0 Prefix and with Strings)

Printing Integer between Strings in Java

Compare two strings lexicographically in Java

How to Initialize and Compare Strings in Java?

Output of Java Programs | Set 52 (Strings Class)

Check whether two Strings are Anagram of each other using HashMap in Java

Swap two Strings without using third user defined variable in Java

CompileTime Vs RunTime Resolution of Strings

Find all palindromic sub-strings of a given string | Set 2

Print all possible strings of length k that can be formed from a set of n characters

Longest Common Substring in an Array of Strings

Number of character corrections in the given strings to make them equal

Java.util.LinkedList.poll(), pollFirst(), pollLast() with examples in Java

RishabhPrabhu
Check out this Author's contributed articles.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Improved By : anthonycathers

Article Tags : Java java-StringBuffer Java-StringBuilder Java-Strings

Practice Tags : Java-Strings Java

15

1.2

Based on **8** vote(s)

To-do Done

Feedback/ Suggest Improvement | Add Notes | Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments