



Chatgpt discussion link on this topic: <https://chatgpt.com/share/807e7321-3cbe-4f0d-addd-b852ad5d6e72>

Strings in Java

Last Updated : 07 Jul, 2024

In the given example only one object will be created. Firstly JVM will not find any string object with the value "Welcome" in the string constant pool, so it will create a new object. After that it will find the string with the value "Welcome" in the pool, it will not create a new object but will return the reference to the same instance. In this article, we will learn about Java Strings.

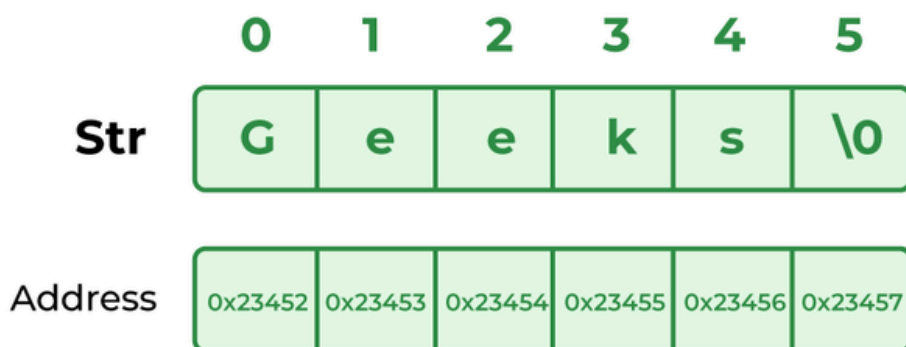
Encoding - done
Encryption

What are Strings in Java?

Strings are the type of objects that can store the character of values and in Java, every character is stored in 16 bits i.e using UTF 16-bit encoding. A string acts the same as an array of characters in Java.

Example:

```
String name = "Geeks";
```



String Example in Java

Below is an example of a String in Java:

Java

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#)

Got It !

```
public class StringExample {  
    // Main Function  
    public static void main(String args[])  
    {  
        String str = new String("example");  
        // creating Java string by new keyword  
        // this statement create two object i.e  
        // first the object is created in heap  
        // memory area and second the object is  
        // created in String constant pool.  
  
        System.out.println(str);  
    }  
}
```

Output

example

Ways of Creating a String

There are two ways to create a string in Java:

- String Literal
- Using new Keyword

Syntax:

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

String s = new String("example")
= "example"

1. String literal

To make Java more memory efficient (because no new objects are created if it exists already in the string constant pool).

Example:

```
String demoString = "GeeksforGeeks";
```

2. Using new keyword

- String s = new String("Welcome");

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

constant pool. The variable `s` will refer to the object in the heap (non-pool)

Example:

```
String demoString = new String ("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: It is a sequence of characters. In Java, objects of `String` are immutable which means a constant and cannot be changed once created.

CharSequence Interface

`CharSequence` Interface is used for representing the sequence of Characters in Java.

Classes that are implemented using the `CharSequence` interface are mentioned below and these provides much of functionality like `substring`, `lastoccurrence`, `first occurrence`, `concatenate`, `toupper`, `tolower` etc.

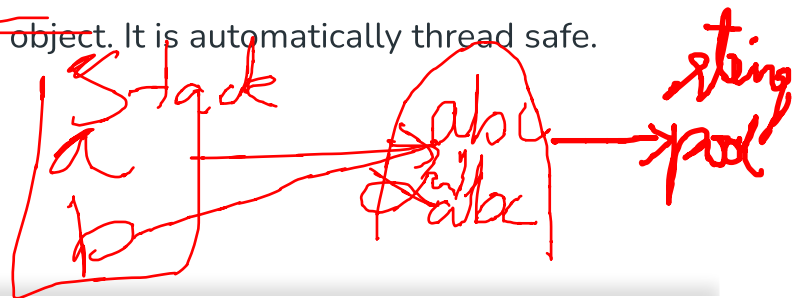
1. `String`
2. `StringBuffer`
3. `StringBuilder`

1. String

`String` is an immutable class which means a constant and cannot be changed once created and if wish to change , we need to create an new object and even the functionality it provides like `toupper`, `tolower`, etc all these return a new object , its not modify the original object. It is automatically thread safe.

Syntax

```
String str= "geeks";  
or
```



StringBuffer is a peer class of **String**, it is mutable in nature and it is thread safe class, we can use it when we have multi threaded environment and shared object of string buffer i.e, used by mutiple thread. As it is thread safe so there is extra overhead, so it is mainly used for multithreaded program.

Syntax:

Thread safety means making sure that when multiple threads work together, they don't interfere with each other and cause errors.

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

→ Strings are automatically thread safe. Since they are immutable

3. StringBuilder

StringBuilder in Java represents an alternative to String and StringBuffer Class, as it creates a mutable sequence of characters and it is not thread safe. It is used only within the thread, so there is no extra overhead, so it is mainly used for single threaded program.

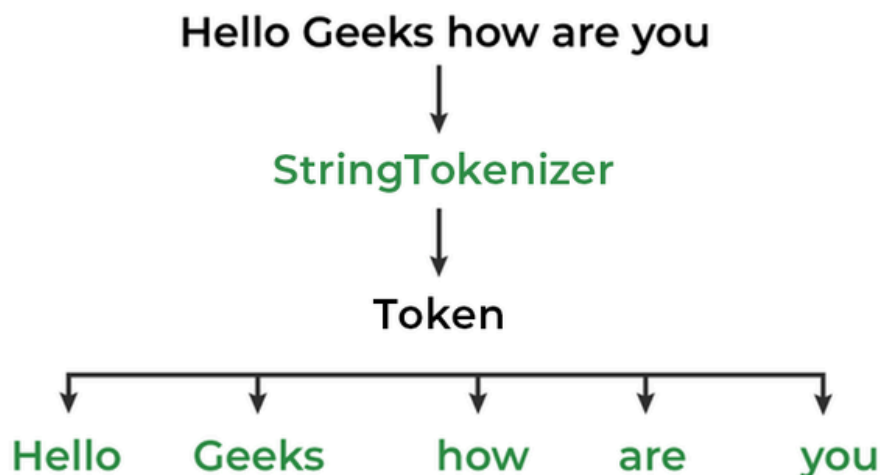
Syntax:

```
StringBuilder demoString = new StringBuilder();
demoString.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

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

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 done with the help of the `StringBuilder` class to append a delimiter after each string, `StringJoiner` provides an easy way to do that without much code to write.

Syntax:

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

Above we saw we can create a string by String Literal.

```
String demoString = "Welcome";
```

Interesting

Here the JVM checks the String Constant Pool. If the string does not exist, then a new string instance is created and placed in a pool. If the string exists, then it will not create a new object. Rather, it will return the reference to the same instance. The cache that stores these string instances is known as the String Constant pool or String Pool. In earlier versions of Java up to JDK 6 String pool was located inside PermGen(Permanent Generation) space. But in JDK 7 it is moved to the main heap area.

Immutable String in Java

In Java, string objects are immutable. Immutable simply means unmodifiable or unchangeable. Once a string object is created its data or state can't be changed but a new string object is created.

Below is the implementation of the topic:

Java

```
// Java Program to demonstrate Immutable String in Java
import java.io.*;

class GFG {
    public static void main(String[] args)
    {
```

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

```
        // This will print Sachin
        // because strings are
        // immutable objects
        System.out.println(s);
    }
}
```

Output

Sachin

Here Sachin is not changed but a new object is created with “Sachin Tendulkar”. That is why a string is known as immutable.

As you can see in the given figure that two objects are created but s reference variable still refers to “Sachin” and not to “Sachin Tendulkar”. But if we explicitly assign it to the reference variable, it will refer to the “Sachin Tendulkar” object.

For Example:

Java

```
// Java Program to demonstrate Explicitly assigned strings
import java.io.*;

class GFG {
    public static void main(String[] args)
    {
        String name = "Sachin";
        name = name.concat(" Tendulkar");
        System.out.println(name);
    }
}
```

Output

Sachin Tendulkar

Memory Allotment of String

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

String literal.

Example:

```
String demoString = "Geeks";
```

```
Example: String c="abc";
String e = "abc";
String f = "abc";
System.out.println(c);//abc
System.out.println(e);//abc
System.out.println(f);//abc
c = c + "d";
System.out.println(c);//abcd
System.out.println(e);//abc
System.out.println(f);//abc
```

The string can also be declared using a **new** operator i.e. dynamically allocated. In case of String are dynamically allocated they are assigned a new memory location in the heap. This string will not be added to the String constant pool.

Example:

```
String demoString = new String("Geeks");
```

If you want to store this string in the constant pool then you will need to "intern" it.

Example:

```
String internedString = demoString.intern();
// this will add the string to string constant pool.
```

It is preferred to use String literals as it allows JVM to optimize memory allocation.

An example that shows how to declare a String

Java

```
// 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 name = "GeeksforGeeks";

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

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

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

Output

```
String name = GeeksforGeeks  
String newString = GeeksforGeeks
```

Note: String Object is created in Heap area and Literals are stored in special memory area known as string constant pool.

Why did the String pool move from PermGen to the normal heap area?

PermGen space is limited, the default size is just 64 MB. it was a problem with creating and storing too many string objects in PermGen space. That's why the String pool was moved to a larger heap area. To make Java more memory efficient, the concept of string literal is used. By the use of the 'new' keyword, The JVM will create a new string object in the normal heap area even if the same string object is present in the string pool.

For example:

```
String demoString = new String("Bhubaneswar");
```

Let us have a look at the concept with a Java program and visualize the actual JVM memory structure:

Below is the implementation of the above approach:

Java

```
// Java program and visualize the  
// actual JVM memory structure  
// mentioned in image  
class StringStorage {  
    public static void main(String args[])
```

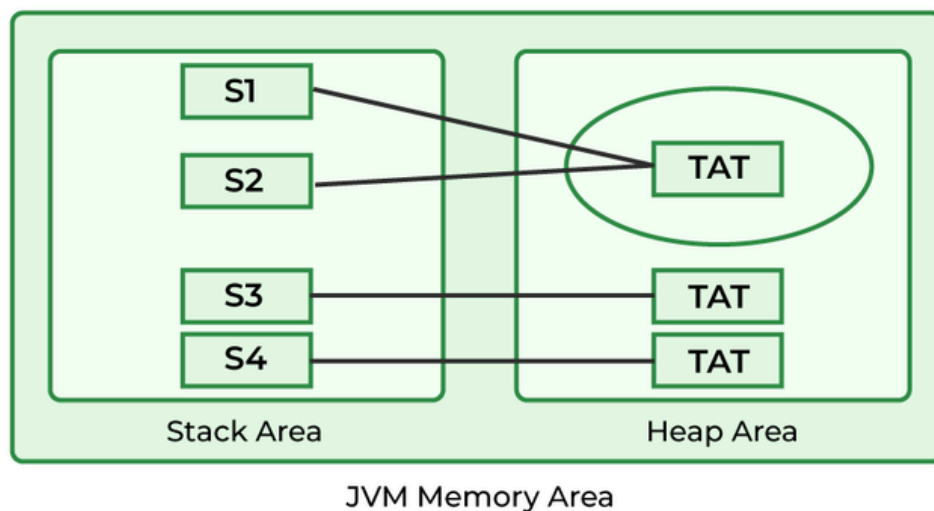
We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).


```
String s4 = new String("TAT");

// Printing all the Strings
System.out.println(s1);
System.out.println(s2);
System.out.println(s3);
System.out.println(s4);
    }
}
```

Output

TAT
TAT
TAT
TAT



String Pool in Java

Note: All objects in Java are stored in a heap. The reference variable is to the object stored in the stack area or they can be contained in other objects which puts them in the heap area also.

Example 1:

Java

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

```
// Construct String from subset of char array

// Driver Class
class GFG {
    // main function
    public static void main(String args[])
    {
        byte ascii[] = { 71, 70, 71 };

        String firstString = new String(ascii);
        System.out.println(firstString);

        String secondString = new String(ascii, 1, 2);
        System.out.println(secondString);
    }
}
```

Output

GFG

FG

Example 2:

Java

```
// Construct one string from another

class GFG {
    public static void main(String args[])
    {
        char characters[] = { 'G', 'f', 'g' };

        String firstString = new String(characters);
        String secondString = new String(firstString);

        System.out.println(firstString);
        System.out.println(secondString);
    }
}
```

Output

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

Frequently Asked Questions

1. What are strings in Java?

Strings are the types of objects which can store characters as elements.

2. Why string objects are immutable in Java?

Because java uses the concept of string literal. Suppose there are 5 reference variables, all refer to one object "Sachin". If one reference variable changes the value of the object, it will be affected by all the reference variables. That is why string objects are immutable in Java.

• 3. Why Java uses the concept of string literal?

To make Java more memory efficient (because no new objects are created if it exists already in the string constant pool).

"The DSA course helped me a lot in clearing the interview rounds. It was really very helpful in setting a strong foundation for my problem-solving skills. Really a great investment, the passion Sandeep sir has towards DSA/teaching is what made the huge difference." - **Gaurav | Placed at Amazon**

Before you move on to the world of development, **master the fundamentals of DSA** on which every advanced algorithm is built upon. Choose your preferred language and start learning today:

[DSA In JAVA/C++](#)

[DSA In Python](#)

[DSA In JavaScript](#)

Trusted by Millions, Taught by One- Join the best DSA Course Today!

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

R Risha... [+ Follow](#)

515

Previous Article

[Bitwise Operators in Java](#)

Next Article

[String class in Java](#)

Similar Reads

Number of common base strings for two strings

Given two strings s1 and s2, we need to find number of common base strings of two. A substring of a string s is called base string if repeated concatenatio...

[6 min read](#)

Count of distinct Strings possible by swapping prefixes of pairs of String...

Given an array string[] consisting of N numeric strings of length M, the task is to find the number of distinct strings that can be generated by selecting any...

[6 min read](#)

Distinct strings such that they contains given strings as sub-sequences

Given two strings str1 and str2 of lengths M and N respectively. The task is to find all the distinct strings of length M + N such that the frequency of any...

[6 min read](#)

Count of Palindrome Strings in given Array of strings

Given an array of strings arr[] of size N where each string consists only of lowercase English letter. The task is to return the count of all palindromic...

[5 min read](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

Given two string S1 and S2 of length L, the task is to count the number of strings of length L, that exists in between S1 and S2, which are...

8 min read

[View More Articles](#)

Article Tags :

[java-StringBuffer](#)

[Java-StringBuilder](#)

[Java-Strings](#)

[Java](#)

[+1 More](#)

Practice Tags :

[Java](#)

[Java-Strings](#)

[Strings](#)



A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305



Company

[About Us](#)

[Legal](#)

[Careers](#)

[In Media](#)

[Contact Us](#)

[Advertise with us](#)

Explore

[Job-A-Thon Hiring Challenge](#)

[Hack-A-Thon](#)

[GfG Weekly Contest](#)

[Offline Classes \(Delhi/NCR\)](#)

[DSA in JAVA/C++](#)

[Master System Design](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning Tutorial
ML Maths
Data Visualisation Tutorial
Pandas Tutorial
NumPy Tutorial
NLP Tutorial
Deep Learning Tutorial

Python Tutorial

Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

DevOps

Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview Questions
Competitive Programming

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
NodeJs
Bootstrap
Tailwind CSS

Computer Science

GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Commerce

Accountancy
Business Studies
Economics
Management
HR Management
Finance
Income Tax

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).

PostgreSQL
PL/SQL
MongoDB

Aptitude Preparation
Puzzles
Company-Wise Preparation
Companies
Colleges

Competitive Exams

JEE Advanced
UGC NET
UPSC
SSC CGL
SBI PO
SBI Clerk
IBPS PO
IBPS Clerk

Free Online Tools

Typing Test
Image Editor
Code Formatters
Code Converters
Currency Converter
Random Number Generator
Random Password Generator

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Share your Experiences
Internships

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy](#).