

In this blog, we will cover in depth knowledge of strings in java using this table of content…

 What is a string?

 How strings are stored in java?

 Ways to define strings in java

 String as an array.

So let us jump in to understanding String functions in java, starting with the strings in java…

**Strings In Java**

A string is a sequence of characters or sentences or words, enclosed between double-quotes. To define a string, use a variable with String data type. String in java is present at java.lang.String class and hence it is an object.

There are two ways to create a string in java

 With string literal

A string can be created by enclosing the characters in double-quotes which creates only one object as follows:

|  |
| --- |
| String str = “Hi there, how is it going?”; |

 With new keyword

A string can be created by prefixing the string with *new*keywords. This will create two objects, in the string pool and in the heap memory. Let me show you an example,

|  |
| --- |
| String str = new String(“Hi there, how is it going?”); |

You might get a question, What is String Pool and Heap Memory? well, the string pool is a collection of all the defined strings, and heap memory has all the unique values of those strings. Whenever you define a new string with a previously saved value, the string pool will address the previously stored same value instead of creating a new one or allocating memory in the heap for the new string.

Hope you understood, what are strings in java, let us now learn about strings as an array in java.

**String as an Array**

Strings in java are immutable and shareable, that is they cannot be changed because a string is the same as an array of characters and as the size of the array cannot be changed after definition, the strings cannot be changed as well.

 Here is an example

|  |
| --- |
| String data = ‘Helloo’;   // is as same as   char string[] = {‘H’,’e’,’l’,’l’,’o’,’o’}; String str = new String(string); |

The*java.lang.String*class applies the three interfaces, *Serializable interface*, *Comparable* interface, and *CharSequence* interface.

If you want to create mutable strings, StringBuilder and StringBuilder classes and a StringTokenizer class can be used to divide the string into tokens.

Check out this Complete [Online Java Course](https://www.fita.in/java-training/) by [FITA](https://www.fita.in/). FITA provides a complete Java course where you will be building real-time applications using *Servlets,*[*Hibernate*](https://www.fita.in/hibernate-training-in-chennai/)*Framework,* and [*Spring*](https://www.fita.in/spring-training-in-chennai/) with Aspect Oriented Programming (AOP) architecture,Struts through *JDBC* bundled with, placement support, and certification at an affordable price with an active placement cell, to make you an industry-certified java developer.

**Methods for Strings In Java**

There are many built-in methods for strings in java for calculating the length, removing spaces for replacing characters from the string, converting to lower or upper case, etc, so let’s discuss a few of them, starting with the length() method in java.

 length() method in java

This function or method returns the size of a string. You might get confused with the length variable. The length variable works on arrays to return size such as array.length whereas the length() method works with strings to return the size of the string. Let me clear with an example

|  |
| --- |
| // Example program for demonstrating length() method import java.util.\*; public class Main { // driver method public static void main(String args[]) { char string[] = {‘H’,’e’,’l’,’l’,’o’,’o’}; System.out.println(string.length); String str = “Hi there everybody”; System.out.println(str.length()); } } |

**Output for the above program**

|  |
| --- |
| **6** **18** |

Hope you understood the length method of strings in java for finding the length of the string, let us now learn the compareTo() method of string in java.

 compareTo() method in java

This method compares the two strings with case sensitivity and returns 0 if both are equal, -n for the number of additional characters and n (positive) for the number of missing characters.Here is an example

|  |
| --- |
| // Example program for demonstrating compareTo() method import java.util.\*; public class Main{ public static void main(String args[]){ String s\_1=”hi there”; String s\_2=”HiThere”; String s\_3 = “hi there everybody”; String s\_4=”hola”;   System.out.println(s1.compareTo(s\_2)); System.out.println(s1.compareTo(s\_3)); System.out.println(s4.compareTo(s\_3)); }} |

**Output for the above program**

|  |
| --- |
| **-32** **-10** **6** |

You can also find the length of the string by comparing it with an empty string.

|  |
| --- |
| String str\_1 = “FITA”; String str\_2 = “”; //empty string   //Returns the length of str\_1 in positive str\_1.compareTo(str\_2); //  outputs 4   // Returns the length of str\_1 in negative str2.compareTo(str1); // outputs -4 |

Hope you understood the compareTo() method of strings in java for comparing two strings, let us now learn the concat() method of string in java.

 concat() method in java

The concat() method concatenates or joins two strings, and returns a single string. Here is an example

|  |
| --- |
| // Example program for demonstrating concat() method import java.util.\*; public class Main { public static void main(String args[]) { String s\_1 = “Hi there, “; s\_1 = s\_1.concat(“how is it going?”); System.out.println(s\_1); } } |

**Output for the above program**

|  |
| --- |
| **Hi there, how is it going?** |

Hope you understood the concat() method of strings in  java for comparing two strings, let us now learn theisEmpty() method of string in java.

 IsEmpty() method in java

This method will return true if the parameter passed contains character(s) or is not empty, else returns false.

|  |
| --- |
| // Example program for demonstrating isEmpty() method import java.util.\*; public class Main { public static void main(String args[]) { String s\_1 = “”; String s\_2 = “following?”; System.out.println(s\_1.isEmpty()); // true System.out.println(s\_2.isEmpty()); // false } } |

**Output for the above program**

|  |
| --- |
| **true** **false** |

Hope you understood the isEmpty() method of strings in java to check if the string is empty, let us now learn the trim() method of string in java.

 trim() method in java

The trim method removes the trailing and leading spaces from the string.

|  |
| --- |
| // Example program for demonstrating trim() method import java.util.\*; public class Main { public static void main(String args[]) { String s1 = ”  hola   “;   // without the trim method System.out.println(s1 + “where have you been”);   // with the trim method System.out.println(s1.trim() + “where have you been”); } } |

**Output for the above program**

|  |
| --- |
| **hola   where have you been** **holawhere have you been** |

Hope you understood the trim() method of strings in java to remove spaces from the string, let us now learn the toLowerCase() method of string in java.

 toLowerCase() method in java

This method converts all the characters to lower case characters.

|  |
| --- |
| // Example program for demonstrating toLowerCase() method import java.util.\*; public class Main { public static void main(String args[]) { String s1 = “It’s been raining since morning”; String low\_s1 = s1.toLowerCase(); System.out.println(low\_s1); } } |

**Output for the above program**

|  |
| --- |
| **it’s been raining since morning** |

Hope you understood the toLowerCase() method of strings in java to change the string to lower case, let us now learn the toUpperCase() method of string in java.

 toUpperCase() method in java

This method will convert all the characters to uppercase and returns the new string.

|  |
| --- |
| // Example program for demonstrating toUpperCase() method import java.util.\*; public class Main { public static void main(String args[]) { String s1 = “It’s been raining since morning”; String low\_s1 = s1.toUpperCase(); System.out.println(low\_s1); } } |

**Output for the above program**

|  |
| --- |
| **IT’S BEEN RAINING SINCE MORNING** |

Hope you understood the toUpperCase() method of strings in java to change the string to uppercase, let us now learn the ValueOf() method of string in java.

 The ValueOf() method in java

This static method converts the passed parameter, a boolean, an integer, character, float or double, char array, or an object to a string. Here is the syntax for the various data type

 String.valueOf(boolean b)

 String.valueOf(char c)

 String.valueOf(char[] data)

 String.valueOf(double d)

 String.valueOf(float f)

 String.valueOf(int b)

 String.valueOf(long l)

 String.valueOf(Object o)

Here is an example program

|  |
| --- |
| // Example program for demonstrating ValueOf() method import java.util.\*; public class Main { public static void main(String[] args) { int i = 4; long lng = -2363861L; float flt = 613.2f; double dbl = 713.433d; char chrs[] = { ‘J’, ‘a’, ‘v’, ‘a’,’,’, ‘J’, ‘a’, ‘v’, ‘a’ };   // convert values to strings System.out.println(String.valueOf(i)); System.out.println(String.valueOf(lng)); System.out.println(String.valueOf(flt)); System.out.println(String.valueOf(dbl));   // convert character array to string System.out.println(String.valueOf(chrs)); } } |

**Output for the above program**

|  |
| --- |
| **4** **-2363861** **613.2** **713.433** **Java, Java** |

Hope you understood the ValueOf() method of strings in java to get the value of a variable, let us now learn the replace() method of string in java.

 replace() method in java

This method takes two arguments, one for the character of the string to be replaced and another one for replacing the character of the string with. Example program

|  |
| --- |
| // Example program for demonstrating replace() method import java.util.\*; public class Main { public static void main(String args[]) { String str = “How is your day going?”; String rep\_str = str.replace(‘H’, ‘W’); System.out.println(rep\_str); } } |

**Output for the above program**

|  |
| --- |
| **Wow is your day going?** |

The replace() method can also take strings as the parameters, as follows

|  |
| --- |
| // Example program for demonstrating replace() method import java.util.\*; public class Main { public static void main(String args[]) { String str = “How is your day going?”; String rep\_str = str.replace(“is”, “was”); System.out.println(rep\_str); } } |

**Output for the above program**

|  |
| --- |
| **How was your day going?** |

Hope you understood the replace() method of strings in java to replace one string with the other, let us now learn the contains() method of string in java.

 contains() method in java

This method takes an argument of string or sequence of characters and will return true if the passed argument or string is found in the original string else returns false.

|  |
| --- |
| // Example program for demonstrating contains() method import java.util.\*; public class Main { public static void main(String args[]) { String name = “Get the best of trainings at FITA Academy”; System.out.println(name.contains(“training”)); System.out.println(name.contains(“expensive”)); System.out.println(name.contains(“FITA Academy”)); } } |

**Output for the above program**

|  |
| --- |
| **true** **false** **true** |

Hope you understood the contains() method of strings in java to check if the given string is present in the other string, let us now learn the equals() and equalsIgnoreCase() method of string in java.

 equals() and equalsIgnoreCase() method in java

The equals() method returns true if the given two arguments or strings are equal, else false with case sensitivity whereas equalsIgnoreCase() method will true without case sensitivity for the matched strings.

|  |
| --- |
| // Example program for demonstrating equals() and equalsIgnoreCase() method import java.util.\*; public class Main { public static void main(String args[]) { String str\_1 = “Best JAVA Training at FITA”; String str\_2 = “best java training at fita”; System.out.println(s1.equals(str\_2)); System.out.println(s1.equalsIgnoreCase(str\_2)); } } |

**Output for the above program**

|  |
| --- |
| **true** **false** |

Hope you understood the equals() method and equalsIgnoreCase() method of strings in java to check if the two strings are equal with and without case sensitivity, let us now learn the toCharArray() method of string in java.

 toCharArray() method in java

This method will convert the passed argument or string, to a character array as follows;

|  |
| --- |
| // Example program for demonstrating toCharArray() method import java.util.\*; public class Main { public static void main(String args[]) { String s1 = “Best Java Training at FITA”; char[] ch = s1.toCharArray(); System.out.print(ch); } } |

**Output for the above program**

|  |
| --- |
| **Best Java Training at FITA** |

Hope you understood the toCharArray() method of strings in java to convert string to an array, let us now learn the endWith() method of string in java.

 endsWith() method in java

This method returns true if the passed character or sequence of characters matches the sequence from the end, else returns false. Example Program

|  |
| --- |
| // Example program for demonstrating endWith() method import java.util.\*; public class Main{ public static void main(String args[]) { String str=”On Campus Java Training in Chennai and Bangalore”; System.out.println(str.endsWith(“e”)); System.out.println(str.endsWith(“Chennai”)); System.out.println(str.endsWith(“Bangalore”)); } } |

**Output for the above program**

|  |
| --- |
| **true** **false** **true** |

Hope you understood the endWith() method of strings in java to check if the string ends with the given string, let us now learn the split() method of string in java.

 split() method in java

This method can take upto two arguments a regex and an optional limit, and will return an array of substrings of the given string, divided at the specified regex, with size of limit (if specified).

|  |
| --- |
| // Example program for demonstrating split() method import java.util.\*; import java.util.Arrays; class Main { public static void main(String[] args) { String str = “a,e,i,o,u,1,2,3,3”; String[] ch = str.split(“,”); System.out.println(“result = ” + Arrays.toString(ch)); } } |

**Output for the above program**

|  |
| --- |
| **result = [a, e, i, o, u, 1, 2, 3, 3]** |

Hope you understood the split() method of strings in java to divide the string on the input, let us now learn the charAt() method of string in java.

 charAt() method in java

The charAt() method takes an index argument, and returns the value at the specified index.

|  |
| --- |
| // Example program for demonstrating charAt() method import java.util.\*; class Main { public static void main(String[] args) { String str\_1 = “Learn Java With FITA”; for (int i = 0; i < str\_1.length(); i++) { System.out.println(str\_1.charAt(i)); } } } |

**Output for the above program**

|  |
| --- |
| **L** **e** **a** **r** **n**  **J** **a** **v** **a**  **W** **i** **t** **h**  **F** **I** **T** **A** |