

Introduction

The **java.lang.String** class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class. Strings are constant, their values cannot be changed after they are created

Class declaration

Following is the declaration for **java.lang.String** class:

```
public final class String
    extends Object
    implements Serializable, Comparable<String>, CharSequence
```

Field

Following are the fields for **java.lang.String** class:

- **static Comparator<String> CASE_INSENSITIVE_ORDER** -- This is a Comparator that orders String objects as by `compareToIgnoreCase`.

Class constructors

S.N.	Constructor & Description
1	String This initializes a newly created String object so that it represents an empty character sequence.
2	Stringbyte[]bytes This constructs a new String by decoding the specified array of bytes using the platform's default charset.
3	Stringbyte[]bytes, Charsetcharset This constructs a new String by decoding the specified array of bytes using the specified charset.
4	Stringbyte[]bytes, intoffset, intlength This constructs a new String by decoding the specified subarray of bytes using the platform's default charset
5	Stringbyte[]bytes, intoffset, intlength, Charsetcharset This constructs a new String by decoding the specified subarray of bytes using the specified charset.
6	

String*byte[]bytes, intoffset, intlength, StringcharsetName*

This constructs a new String by decoding the specified subarray of bytes using the specified charset.

7

String*byte[]bytes, StringcharsetName*

This constructs a new String by decoding the specified array of bytes using the specified charset.

8

String*char[]value*

This allocates a new String so that it represents the sequence of characters currently contained in the character array argument.

9

String*char[]value, intoffset, intcount*

This allocates a new String that contains characters from a subarray of the character array argument.

10

String*int[]codePoints, intoffset, intcount*

This allocates a new String that contains characters from a subarray of the Unicode code point array argument.

11

String*Stringoriginal*

This initializes a newly created String object so that it represents the same sequence of characters as the argument; in other words, the newly created string is a copy of the argument string.

12

String*StringBufferbuffer*

This allocates a new string that contains the sequence of characters currently contained in the string buffer argument.

13

String*StringBuilderbuilder*

This allocates a new string that contains the sequence of characters currently contained in the string builder argument.

Class methods

S.N.	Method & Description
------	----------------------

1

[char charAt\(intindex\)](#)

This method returns the char value at the specified index.

2

[int codePointAt\(int index\)](#)

This method returns the character *Unicode codepoint* at the specified index.

3

[int codePointBefore\(int index\)](#)

This method returns the character *Unicode codepoint* before the specified index.

4

[int codePointCount\(int beginIndex, int endIndex\)](#)

This method returns the number of Unicode code points in the specified text range of this String.

5

[int compareTo\(String anotherString\)](#)

This method compares two strings lexicographically.

6

[int compareToIgnoreCase\(String str\)](#)

This method compares two strings lexicographically, ignoring case differences.

7

[String concat\(String str\)](#)

This method concatenates the specified string to the end of this string.

8

[boolean contains\(CharSequence\)](#)

This method returns true if and only if this string contains the specified sequence of char values.

9

[boolean contentEquals\(CharSequence\)](#)

This method compares this string to the specified CharSequence.

10

[boolean contentEquals\(StringBuffer\)](#)

This method compares this string to the specified StringBuffer.

11

[static String copyValueOf\(char\[\] data\)](#)

This method returns a String that represents the character sequence in the array specified.

12

[static String copyValueOf\(char\[\] data, int offset, int count\)](#)

This method returns a String that represents the character sequence in the array specified.

13

[boolean endsWithStringsuffix](#)

This method tests if this string ends with the specified suffix.

14

[boolean equalsObjectanObject](#)

This method compares this string to the specified object.

15

[boolean equalsIgnoreCaseStringanotherString](#)

This method compares this String to another String, ignoring case considerations.

16

[static String formatLocalel, Stringformat, Object... args](#)

This method returns a formatted string using the specified locale, format string, and arguments.

17

[static String formatStringformat, Object... args](#)

This method returns a formatted string using the specified format string and arguments.

18

[byte\[\] getBytes](#)

This method encodes this String into a sequence of bytes using the platform's default charset, storing the result into a new byte array.

19

[byte\[\] getBytesCharsetcharset](#)

This method encodes this String into a sequence of bytes using the given charset, storing the result into a new byte array.

20

[byte\[\] getBytesStringcharsetName](#)

This method encodes this String into a sequence of bytes using the named charset, storing the result into a new byte array.

21

[void getCharsintsrcBegin, intsrcEnd, char\[\]dst, intdstBegin](#)

This method copies characters from this string into the destination character array.

22

[int hashCode](#)

This method returns a hash code for this string.

23

[int indexOfintch](#)

This method returns the index within this string of the first occurrence of the specified character.

24

[int indexOf\(intch, intfromIndex\)](#)

This method returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.

25

[int indexOfStringstr](#)

This method returns the index within this string of the first occurrence of the specified substring.

26

[int indexOfStringstr, intfromIndex](#)

This method returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

27

[String intern](#)

This method returns a canonical representation for the string object.

28

[boolean isEmpty](#)

This method returns true if, and only if, length is 0.

29

[int lastIndexOf\(intch\)](#)

This method returns the index within this string of the last occurrence of the specified character.

30

[int lastIndexOf\(intch, intfromIndex\)](#)

This method returns the index within this string of the last occurrence of the specified character, searching backward starting at the specified index.

31

[int lastIndexOfStringstr](#)

This method returns the index within this string of the rightmost occurrence of the specified substring.

32

[int lastIndexOfStringstr, intfromIndex](#)

This method returns the index within this string of the last occurrence of the specified substring, searching backward starting at the specified index.

33

[int length](#)

This method returns the length of this string.

34

[boolean matchesStringregex](#)

This method tells whether or not this string matches the given regular expression.

35

[int offsetByCodePoints\(int index, int codePointOffset\)](#)

This method returns the index within this String that is offset from the given index by codePointOffset code points.

36

[boolean regionMatches\(boolean ignoreCase, int toffset, String other, int toffset, int len\)](#)

This method tests if two string regions are equal with case ignored.

37

[boolean regionMatches\(int toffset, String other, int toffset, int len\)](#)

This method tests if two string regions are equal.

38

[String replace\(char oldChar, char newChar\)](#)

This method returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.

39

[String replace\(CharSequence target, CharSequence replacement\)](#)

This method replaces each substring of this string that matches the literal target sequence with the specified literal replacement sequence.

40

[String replaceAll\(String regex, String replacement\)](#)

This method replaces each substring of this string that matches the given regular expression with the given replacement.

41

[String replaceFirst\(String regex, String replacement\)](#)

This method replaces the first substring of this string that matches the given regular expression with the given replacement.

42

[String\[\] split\(String regex\)](#)

This method splits this string around matches of the given regular expression.

43

[String\[\] split\(String regex, int limit\)](#)

This method splits this string around matches of the given regular expression.

44

[boolean startsWith\(String prefix\)](#)

This method tests if this string starts with the specified prefix.

45

[boolean startsWith\(String prefix, int toffset\)](#)

This method tests if the substring of this string beginning at the specified index starts with the specified prefix.

46

[CharSequence subSequence\(intbeginIndex, intendIndex\)](#)

This method returns a new character sequence that is a subsequence of this sequence.

47

[String substring\(intbeginIndex\)](#)

This method returns a new string that is a substring of this string.

48

[String substring\(intbeginIndex, intendIndex\)](#)

This method returns a new string that is a substring of this string.

49

[char\[\] toCharArray](#)

This method converts this string to a new character array.

50

[String toLowerCase](#)

This method converts all of the characters in this String to lower case using the rules of the default locale.

51

[String toLowerCase\(Localelocale\)](#)

This method converts all of the characters in this String to lower case using the rules of the given Locale.

52

[String toString](#)

This method returns the string itself.

53

[String toUpperCase](#)

This method converts all of the characters in this String to upper case using the rules of the default locale.

54

[String toUpperCase\(Localelocale\)](#)

This method converts all of the characters in this String to upper case using the rules of the given Locale.

55

[String trim](#)

This method returns a copy of the string, with leading and trailing whitespace omitted.

56

[static String valueOf\(booleanb\)](#)

This method returns the string representation of the boolean argument.

57

[static String valueOfcharc](#)

This method returns the string representation of the char argument.

58

[static String valueOfchar\[\]data](#)

This method returns the string representation of the char array argument.

59

[static String valueOfchar\[\]data, intoffset, intcount](#)

This method Returns the string representation of a specific subarray of the char array argument.

60

[static String valueOfdoubled](#)

This method returns the string representation of the double argument.

61

[static String valueOffloatf](#)

This method returns the string representation of the float argument.

62

[static String valueOfinti](#)

This method returns the string representation of the int argument.

63

[static String valueOflongl](#)

This method returns the string representation of the long argument.

64

[static String valueOfObjectobj](#)

This method returns the string representation of the Object argument.