**String Methods**

1. **int length():**Returns the number of characters in the String.

"GeeksforGeeks".length();  // returns 13

1. [**Char charAt(int i)**](https://www.geeksforgeeks.org/java-string-charat-method-example/)**:**Returns the character at ith index.

"GeeksforGeeks".charAt(3); // returns  ‘k’

1. [**String substring (int i)**](https://www.geeksforgeeks.org/substring-in-java/)**:**Return the substring from the ith index character to end.

"GeeksforGeeks".substring(3); // returns “ksforGeeks”

1. [**String substring (int i, int j)**](https://www.geeksforgeeks.org/substring-in-java/)**:**Returns the substring from i to j-1 index.

"GeeksforGeeks".substring(2, 5); // returns “eks”

1. [**String concat( String str)**](https://www.geeksforgeeks.org/java-string-concat-examples/)**:**Concatenates specified string to the end of this string.
2. String s1 = ”Geeks”;
3. String s2 = ”forGeeks”;
4. String output = s1.concat(s2); // returns “GeeksforGeeks”
5. [**int indexOf (String s)**](https://www.geeksforgeeks.org/java-string-indexof/)**:**Returns the index within the string of the first occurrence of the specified string.
6. String s = ”Learn Share Learn”;
7. int output = s.indexOf(“Share”); // returns 6
8. [**int indexOf (String s, int i)**](https://www.geeksforgeeks.org/java-string-indexof/)**:**Returns the index within the string of the first occurrence of the specified string, starting at the specified index.
9. String s = ”Learn Share Learn”;
10. int output = s.indexOf("ea",3);// returns 13
11. [**Int lastIndexOf( String s)**](https://www.geeksforgeeks.org/java-lang-string-lastindexof-method/)**:**Returns the index within the string of the last occurrence of the specified string.
12. String s = ”Learn Share Learn”;
13. int output = s.lastIndexOf("a"); // returns 14
14. **boolean equals( Object otherObj):**Compares this string to the specified object.
15. Boolean out = “Geeks”.equals(“Geeks”); // returns true
16. Boolean out = “Geeks”.equals(“geeks”); // returns false
17. [**boolean  equalsIgnoreCase (String anotherString)**](https://www.geeksforgeeks.org/equalsignorecase-in-java/)**:**Compares string to another string, ignoring case considerations.
18. Boolean out= “Geeks”.equalsIgnoreCase(“Geeks”); // returns true

Boolean out = “Geeks”.equalsIgnoreCase(“geeks”); // returns true

1. [**int compareTo( String anotherString)**](https://www.geeksforgeeks.org/java-lang-string-compareto/)**:**Compares two string lexicographically.
2. int out = s1.compareTo(s2);  // where s1 ans s2 are
3. // strings to be compared
4. This returns difference s1-s2. If :
5. out < 0 // s1 comes before s2
6. out = 0 // s1 and s2 are equal.
7. out > 0 // s1 comes after s2.
8. **int compareToIgnoreCase( String anotherString):**Compares two string lexicographically, ignoring case considerations.
9. int out = s1.compareToIgnoreCase(s2);
10. // where s1 ans s2 are
11. // strings to be compared
12. This returns difference s1-s2. If :
13. out < 0 // s1 comes before s2
14. out = 0 // s1 and s2 are equal.
15. out > 0 // s1 comes after s2.

*Note- In this case, it will not consider case of a letter (it will ignore whether it is uppercase or lowercase).*

1. [**String toLowerCase()**](https://www.geeksforgeeks.org/java-string-tolowercase-examples/)**:**Converts all the characters in the String to lower case.
2. String word1 = “HeLLo”;
3. String word3 = word1.toLowerCase(); // returns “hello"
4. [**String toUpperCase()**](https://www.geeksforgeeks.org/java-touppercase-examples/)**:**Converts all the characters in the String to upper case.
5. String word1 = “HeLLo”;
6. String word2 = word1.toUpperCase(); // returns “HELLO”
7. [**String trim()**](https://www.geeksforgeeks.org/java-string-trim-method-example/)**:**Returns the copy of the String, by removing whitespaces at both ends. It does not affect whitespaces in the middle.
8. String word1 = “ Learn Share Learn “;
9. String word2 = word1.trim(); // returns “Learn Share Learn”
10. [**String replace (char oldChar, char newChar)**](https://www.geeksforgeeks.org/java-lang-string-replace-method-java/)**:**Returns new string by replacing all occurrences of *oldChar*with *newChar.*
11. String s1 = “feeksforfeeks“;
12. String s2 = “feeksforfeeks”.replace(‘f’ ,’g’); // returns “geeksgorgeeks”

*Note:- s1 is still feeksforfeeks and s2 is geeksgorgeeks*