String Methods:

charAt()

* The **Java String class charAt()** *method returns*a char value at the given index number*.*
* The index number starts from 0 and goes to n-1, where n is the length of the string. It returns **StringIndexOutOfBoundsException,** if the given index number is greater than or equal to this string length or a negative number.
* Syntax:

public char charAt(int index)

1. Program1: // Print Characters Presented at Odd Positions by Using the charAt() Method
2. Program2: // Counting Frequency of a character in a String by Using the charAt() Method
3. Program3: // Counting Frequency each character in a String by Using the charAt() Method

compareTo()

The **Java String class compareTo()** method compares the given string with the current string lexicographically. It returns a positive number, negative number, or 0.

It compares strings on the basis of the Unicode value of each character in the strings.

If the first string is lexicographically greater than the second string, it returns a positive number (difference of character value). If the first string is less than the second string lexicographically, it returns a negative number, and if the first string is lexicographically equal to the second string, it returns 0.

1. **if** s1 > s2, it returns positive number
2. **if** s1 < s2, it returns negative number
3. **if** s1 == s2, it returns 0

Syntax

1. **public** **int** compareTo(String anotherString)

**Example**:

**public** **class** CompareToExample

{

**public** **static** **void** main(String args[])

{

String s1="hello";

String s2="hello";

String s3="meklo";

String s4="hemlo";

String s5="flag";

System.out.println(s1.compareTo(s2));//0 because both are equal

System.out.println(s1.compareTo(s3));//-5 because "h" is 5 times lower than "m"

System.out.println(s1.compareTo(s4));//-1 because "l" is 1 times lower than "m"

System.out.println(s1.compareTo(s5));//2 because "h" is 2 times greater than "f"

}

}

# **Java String concat**

The **Java String class concat()** method combines specified string at the end of this string. It returns a combined string. It is like appending another string.

### Signature

The signature of the string concat() method is given below:

1. **public** String concat(String anotherString)

### Parameter

**anotherString** : another string i.e., to be combined at the end of this string.

### Returns

combined string

# **Java String contains()**

The **Java String class contains()** method searches the sequence of characters in this string. It returns true if the sequence of char values is found in this string otherwise returns false.

### Signature

The signature of string contains() method is given below:

1. **public** **boolean** contains(CharSequence sequence)

### Parameter

**sequence** : specifies the sequence of characters to be searched.

### Returns

**true** if the sequence of char value exists, otherwise **false**.