

### 1. charAt(int index)

Returns the character at the given index.

ex.

```
String a = "abc";  
a.charAt(0) ==> 'a';
```

If it exceeds the range then it will give the `StringIndexOutOfBoundsException`.

ex.

```
String a = "abc";  
a.charAt(-5) ==> StringIndexOutOfBoundsException  
a.charAt(100) ==> --||--
```

### 2. compareTo()

Returns -1 if comparison is false otherwise 0

ex.

```
String a = "abc";  
String b = "bcd";  
String c = "abc";  
  
a.compareTo(b) ==> -1  
a.compareTo(c) ==> 0
```

### 3. concat()

Returns the new object of concatenated string.

ex.

```
String a = "abc";  
String b = "bcd";  
  
a.concat(b) ==> "abcbcd";
```

### 4. contains(String str)

Returns boolean "true" if given string is there. otherwise "false".

ex.

```
String a = "abc";  
a.contains("b") ==> true  
a.contains("") ==> true  
a.contains("z") ==> false
```

### 5. endsWith(String str)

Returns boolean true if our string ends with str. otherwise false.

Ex.

```
String a = "abc";  
a.endsWith("bc"); ==> true  
a.endsWith("zfasdfs") ==> false  
a.endsWith("") ==> true
```

## 6. startsWith(String str)

Returns boolean true if our string starts with str. otherwise false.

Ex.

```
String a = "abc";  
a.startsWith("ab"); => true  
a.startsWith("zfasdfs") => false  
a.startsWith("") => true
```

## 7. equals(String str)

Returns boolean if given string is equals on which we are calling on it.

Ex.

```
String a = "abc";  
String b = "bcd";  
String c = "abc";  
  
a.equals(b) => false  
a.equals(c) => true
```

## 7. equalsIgnoreCase(String str)

Returns boolean if given string is equals on which we are calling on it by ignoring the case.

Ex.

```
String a = "abc";  
String b = "bCd";  
String c = "Abc";  
  
a.equals(b) => false  
a.equals(c) => true
```

## 8. format()

The java string format() method returns the formatted string by given locale, format and arguments.

Ex.

```
String name="abcc";  
String.format("name is %s",name);  => name is abcc
```

## 9. getBytes()

-----

The java string getBytes() method returns the byte array of the string. In other words, it returns sequence of bytes.

Ex.

```
String name = "fuck"  
name.getBytes() => { 102, 117, 99, 107 }
```

## 10. getChars(int start, int end, char[] ch, int begin)

The java string getChars() method copies the content of this string into specified char array.

start => start index in given string.  
end => end index in given string.  
ch => in which array we have to fill.  
begin => start index in ch array.

Ex

```
String str = new String("parimal");  
char[] ch = new char[10];  
str.getChars(2, 5, ch, 0);  
System.out.println(ch);
```

## 11. indexOf(String str)

Returns the first occurrence index of given str in the main string.  
It returns -1 if didn't get.

Ex.

```
String str = "abc";  
str.indexOf("a") => 0  
str.indexOf("") => 0  
str.indexOf("p") => -1  
str.indexOf("bc") => 1
```

## 12. intern()

The java string intern() method returns the interned string.  
It returns the canonical representation of string.

Ex.

```
String s1 = "Javatpoint";  
String s2 = s1.intern();  
String s3 = new String("Javatpoint");  
String s4 = s3.intern();  
  
System.out.println(s1==s2) => True  
System.out.println(s1==s3) => False  
System.out.println(s1==s4) => True  
System.out.println(s2==s3) => False  
System.out.println(s2==s4) => True  
System.out.println(s3==s4) => False
```

## 13. isEmpty(String str)

Returns the boolean true if the given string is empty.  
otherwise false.

Ex.

```
String str = "pa"  
str.isEmpty() => false;  
  
str = ""  
str.isEmpty() => true;
```

#### 14. join( delimiter, string seperated with comma )

The java string join() method returns a string joined with given delimiter. In string join method, delimiter is copied for each elements.

Ex.

```
String joinString1=String.join("-", "welcome", "to", "parimal");  
=> welcome-to-parimal
```

#### 15. lastIndexOf()

The java string lastIndexOf() method returns last index of the given character value or substring. If it is not found, it returns -1. The index counter starts from zero.

```
=> lastIndexOf (int ch)  
// returns last index position for the given char value  
  
=> lastIndexOf (int ch, int fromIndex)  
// returns last index position for the given char value and from index  
  
=> lastIndexOf (String substring)  
// returns last index position for the given substring  
  
=> lastIndexOf (String substring, int fromIndex)  
// returns last index position for the given substring and from index
```

Ex.

```
String s1="this is index of example";  
int index1=s1.lastIndexOf('s');  
System.out.println(index1); // 6 but not 3
```

#### 16. length()

Returns the length of the given string.

Ex.

```
String a = "acn"  
a.length() => 3
```

#### 17. replace()

The java string replace() method returns a string replacing all the old char or CharSequence to new char or CharSequence.

Ex.

```
String a = "parimal is good"  
a.replace("p", "&&")  
=> &&arimal is good.  
  
a = "parimal is good and rushikehs is good"  
a.replace("is", "was")  
=> arimal was good and rushikehs was good
```

## 18. split()

The java string split() method splits this string against given regular expression and returns a char array.

Ex.

```
String s = "parimal is good annd others is good."
```

```
s.split("\\s") => { "parimal", "is", "good", "annd", "others", "is", "good" }
```

```
s.split("") => { "p", "a", "r", "i", "m", "a", "l", " ", "i", "s", " ", "g", "o",  
"o", "d", " ", " ", "a", "n", "n", "d", " ", " ", "o", "t", "h", "e", "r",  
"s", " ", " ", "i", "s", " ", " ", "g", "o", "o", "d" }
```

```
String[] ch = s.split("",5) // by providing the limit.
```

```
ch ==> String[5] { "p", "a", "r", "i", "mal is good annd others is good" }
```

## 19. substring(startindex, endindex)

The java string substring() method returns a part of the string.

ex.

```
String str = "parimal"
```

```
str.substring(0,5) => "parim"
```

```
str.substring(3) => "imal"
```

## 20. toCharArray(String str)

The java string toCharArray() method converts this string into character array. It returns a newly created character array, its length is similar to this string and its contents are initialized with the characters of this string.

Ex.

```
String str = "parimal"  
str.toCharArray() => { 'p', 'a', 'r', 'i', 'm', 'a', 'l' }
```

```
String str = "parimal  " // with 3 extra spaces.  
str.toCharArray() => { 'p', 'a', 'r', 'i', 'm', 'a', 'l', ' ', ' ', ' ' }
```

## 21. toUpperCase()

The java string toLowerCase() method returns the string in lowercase letter. In other words, it converts all characters of the string into lower case letter.

ex.

```
String a = "parimal"  
a.toUpperCase() => "PARIMAL"
```

## 21. toLowerCase()

The java string toLowerCase() method returns the string in uppercase letter. In other words, it converts all characters of the string into upper case letter.

ex.

```
String a = "PARIMAL"  
a.toLowerCase() => "parimal"
```

## 22. trim()

trim() method eliminates leading and trailing spaces.

ex.

```
String s = "parimal      "  
s.trim() => "parimal"
```

## 23. valueOf()

The java string valueOf() method converts different types of values into string. By the help of string valueOf() method, you can convert int to string, long to string, boolean to string, character to string, float to string, double to string, object to string and char array to string.

ex.

```
String.valueOf(3) => "3"  
String.valueOf(3.633) => "3.633"  
String.valueOf(true) => "true"  
String.valueOf(false) => "false"  
String.valueOf('f') => "f"
```