

Strings Class

```
String greeting = "Hello world!";
```

OR

```
public class StringDemo {  
  
    public static void main(String args[]) {  
        char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.' };  
        String helloString = new String(helloArray);  
        System.out.println( helloString );  
    }  
}
```

This will produce the following result

Output

hello.

Strings Class

Another Example

```
public class StringDemo {  
  
    public static void main(String args[]) {  
        String palindrome = "Dot saw I was Tod";  
        int len = palindrome.length();  
        System.out.println( "String Length is : " + len );  
    }  
}
```

Strings Class

Concatenating Strings

The String class includes a method for concatenating two strings

```
string1.concat(string2);
```

This returns a new string that is string1 with string2 added to it at the end. You can also use the concat() method with string literals, as in

```
"My name is ".concat("Shoukat");
```

Strings are more commonly concatenated with the + operator, as in –

```
"Hello," + " world" + "!"  
which results in –
```

```
"Hello, world!"
```

```
public class StringDemo {  
  
    public static void main(String args[]) {  
        String string1 = "saw I was ";  
        System.out.println("Dot " + string1 + "Tod");  
    }  
}
```

Strings Class

printf() and format()

Example

```
System.out.printf("The value of the float variable is " +  
    "%f, while the value of the integer " +  
    "variable is %d, and the string " +  
    "is %s", floatVar, intVar, stringVar);
```

You can write –

```
String fs;
```

```
fs = String.format("The value of the float variable is " +  
    "%f, while the value of the integer " +  
    "variable is %d, and the string " +  
    "is %s", floatVar, intVar, stringVar);
```

```
System.out.println(fs);
```

char charAt(int index)

Returns the character at the specified index.

int compareTo(Object o)

Compares this String to another Object.

int compareTo(String anotherString)

Compares two strings lexicographically.

int compareToIgnoreCase(String str)

Compares two strings lexicographically, ignoring case differences.

String concat(String str)

Concatenates the specified string to the end of this string.

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

Returns true if and only if this String represents the same sequence of characters as the specified StringBuffer.

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

Returns a String that represents the character sequence in the array specified.

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

Returns a String that represents the character sequence in the array specified.

[boolean endsWith\(String suffix\)](#)

Tests if this string ends with the specified suffix.

[boolean equals\(Object anObject\)](#)

Compares this string to the specified object.

[boolean equalsIgnoreCase\(String anotherString\)](#)

Compares this String to another String, ignoring case considerations.

[byte getBytes\(\)](#)

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

[byte\[\] getBytes\(String charsetName\)](#)

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

[void getChars\(int srcBegin, int srcEnd, char\[\] dst, int dstBegin\)](#)

Copies characters from this string into the destination character array.

[int hashCode\(\)](#)

Returns a hash code for this string.

[int indexOf\(int ch\)](#)

Returns the index within this string of the first occurrence of the specified character.

[int indexOf\(int ch, int fromIndex\)](#)

Returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.

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

Returns the index within this string of the first occurrence of the specified substring.

[int indexOf\(String str, int fromIndex\)](#)

Returns the index within this string of the first occurrence of the specified substring, starting at the specified index.

[String intern\(\)](#)

Returns a canonical representation for the string object.

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

Returns the index within this string of the last occurrence of the specified character.

[int lastIndexOf\(int ch, int fromIndex\)](#)

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

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

Returns the index within this string of the rightmost occurrence of the specified substring.

[int lastIndexOf\(String str, int fromIndex\)](#)

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

[int length\(\)](#)

Returns the length of this string.

boolean matches(String regex)

Tells whether or not this string matches the given regular expression.

boolean regionMatches(boolean ignoreCase, int toffset, String other, int ooffset, int len)

Tests if two string regions are equal.

boolean regionMatches(int toffset, String other, int ooffset, int len)

Tests if two string regions are equal.

String replace(char oldChar, char newChar)

Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.

String replaceAll(String regex, String replacement)

Replaces each substring of this string that matches the given regular expression with the given replacement.

String replaceFirst(String regex, String replacement)

Replaces the first substring of this string that matches the given regular expression with the given replacement.

String[] split(String regex)

Splits this string around matches of the given regular expression.

String[] split(String regex, int limit)

Splits this string around matches of the given regular expression.

boolean startsWith(String prefix)

Tests if this string starts with the specified prefix.

boolean startsWith(String prefix, int toffset)

Tests if this string starts with the specified prefix beginning a specified index.

CharSequence subSequence(int beginIndex, int endIndex)

Returns a new character sequence that is a subsequence of this sequence.

String substring(int beginIndex)

Returns a new string that is a substring of this string.

String substring(int beginIndex, int endIndex)

Returns a new string that is a substring of this string.

char[] toCharArray()

Converts this string to a new character array.

String toLowerCase()

Converts all of the characters in this String to lower case using the rules of the default locale.

String toLowerCase(Locale locale)

Converts all of the characters in this String to lower case using the rules of the given Locale.

String toString()

This object (which is already a string!) is itself returned.

String toUpperCase()

Converts all of the characters in this String to upper case using the rules of the default locale.

String toUpperCase(Locale locale)

Converts all of the characters in this String to upper case using the rules of the given Locale.

String trim()

Returns a copy of the string, with leading and trailing whitespace omitted.

static String valueOf(primitive data type x)

Returns the string representation of the passed data type argument.