

# Java Strings

# Java String

In Java, **String objects are immutable**. Immutable simply means unmodifiable or unchangeable.

Once String object is created its data or state **can't be changed** but a new String object is created.

## **Testimmutablestring.java**

```
class Testimmutablestring{  
    public static void main(String args[]){  
        String s="Indian";  
        s.concat(" Team");//concat() method appends the string at the end  
        System.out.println(s);//will print Indian because strings are immutable objects  
    }  
}
```

# Java String

String is basically an object that represents sequence of char values. An array of characters works same as Java string.

For example:

```
1.char[] ch={'j','a','v','a','w','o','r','l','d'};  
2.String s=new String(ch);
```

is same as:

```
1.String s="javaworld";
```

# Java String compare

There are three ways to compare String in Java:

- 1.By Using **equals()** Method
- 2.By Using **==** Operator
- 3.By **compareTo()** Method

# String Concatenation

In Java, String concatenation forms a new String that is the combination of multiple strings. There are two ways to concatenate strings in Java:

- 1.By **+** (String concatenation) operator
- 2.By **concat()** method

# Substring in Java

You can get substring from the given String object by one of the two methods:

1. **public String substring(int startIndex)**

2 **public String substring(int startIndex, int endIndex)**

.



# Java String format()

```
public class FormatExample{  
  
    public static void main(String[] args) {  
        String name = "Java";  
        String sf1 = String.format("name is %s", name);  
        String sf2 = String.format("value is %f", 72.7834);  
  
        // returns 12 char fractional part filling with 0 and total len 17  
        String sf3 = String.format("value is %17.12f", 32.33434);  
  
        System.out.println(sf1);  
        System.out.println(sf2);  
        System.out.println(sf3);  
    }  
}
```

```
name is      Java  
value is 72.783400  
value is 32.334340000000
```

Apart from formatting, we can set width, padding etc. of any value.

```
public class FormatExample3 {  
    public static void main(String[] args) {  
        String sf0 = String.format("%d", 101);  
        String sf1 = String.format("|%10d|", 101);  
        String sf2 = String.format("|%-10d|", 101);  
        String sf3 = String.format("|%d|", 101);  
        String sf4 = String.format("|%010d|", 101);  
  
        System.out.println(sf0);  
        System.out.println(sf1);  
        System.out.println(sf2);  
        System.out.println(sf3);  
        System.out.println(sf4);  
    }  
}
```

```
101  
|          101|  
|101        |  
|101|       |  
|0000000101|
```



# Java String Class Others Methods

The **java.lang.String** class provides a lot of built-in methods that are used to manipulate **string in Java**.

By the help of these methods, we can perform operations on String objects such as trimming, concatenating, converting, comparing, replacing strings etc.

Let's try a few of them..

# Java StringBuffer Class

Java StringBuffer class is used to create mutable (modifiable) String objects. The StringBuffer class in Java is the same as String class except it is mutable i.e. it can be changed.

`append(String s)`

It is used to append the specified string with this string. The `append()` method is overloaded like `append(char)`, `append(boolean)`, `append(int)`, `append(float)`, `append(double)` etc.

`insert(int offset, String s)`

It is used to insert the specified string with this string at the specified position. The `insert()` method is overloaded like `insert(int, char)`, `insert(int, boolean)`, `insert(int, int)`, `insert(int, float)`, `insert(int, double)` etc.

`replace(int startIndex, int endIndex, String str)`

It is used to replace the string from specified `startIndex` and `endIndex`.

`delete(int startIndex, int endIndex)`

It is used to delete the string from specified `startIndex` and `endIndex`.

`reverse()`

is used to reverse the string.

# Java StringBuilder Class

Java StringBuilder class is used to create mutable (modifiable) String.

The Java StringBuilder class is same as StringBuffer class except that it is **non-synchronized**.