

Java – Replace all occurrences of a substring in a String

Java – Replace all Occurrences of a Substring in a String

To replace the all occurrences of a string `old_string` in `str1` with `new_string`, you can use `str1.replaceAll(old_string, new_string)` function.

In your application, when working on string processing or some cleaning, you may need to replace some or all occurrences of words with others.

In this tutorial, we will learn the syntax of `String.replaceAll()` function and how to use it to replace all occurrences of a string in another string.

The syntax of `String.replaceAll()` is given below.

```
public String replaceAll(String regex, String replacement)
```

For `regex`, you can provide the old string you would like to replace. You may provide a string constant or some regex, based on your Java application's requirement.

For `replacement`, provide the new string you would like to replace with.

Example 1 – Replace all occurrences with New string

In this example, we shall take three strings: `str1`, `old_string` and `new_string`. We shall replace all the occurrences of the string `old_string` with `new_string` in `str1`.

ReplaceAllOccurrences.java

```
/*
 * Java Example Program, to replace all occurrences of a substring with another in a string
 */

public class ReplaceAllOccurrences {

    public static void main(String[] args) {
        String str1 = "Hi! Good morning. Have a Good day.";
        String str2 = str1.replaceAll("Good", "Nice");
        System.out.println(str2);
    }
}
```

```

String old_string = "Good";
String new_string = "Very-Good";

//replace all occurrences
String resultStr = str1.replaceAll(old_string, new_string);

System.out.println(resultStr);
}
}

```

Run the above Java program in console or your favorite IDE, and you shall see an output similar to the following.

Output

```
Hi! Very-Good morning. Have a Very-Good day.
```

All the occurrences of old string are replaced with new string.

Example 2 – Replace all occurrences with New string – Negative Scenario

In this example, we shall take three strings: str1, old_string and new_string. We shall try to replace all the occurrences of the string **old_string** with **new_string** in **str1**. But **old_string** is not present in **str1**.

ReplaceAllOccurrences.java

```

/**
 * Java Example Program, to replace all occurrences of a substring with another in a string
 */

public class ReplaceAllOccurrences {

    public static void main(String[] args) {
        String str1 = "Hi! Good morning. Have a Good day.";
        String old_string = "Bad";
        String new_string = "Very-Good";

        //replace all occurrences
        String resultStr = str1.replaceAll(old_string, new_string);

        System.out.println(resultStr);
    }
}

```

Run the above program.

Output

```
Hi! Good morning. Have a Good day.
```

Hi! Good morning. Have a Good day.

Since the old_string is not present in str1, replaceAll() returns str1 unchanged.

Conclusion

In this [Java Tutorial](#), we have learned how to replace the first occurrence of a sub-string with another in a string.

- ◆ [Java Tutorial](#)

[Java String Operations](#)

- ◆ [Java String](#)
- ◆ [Java String Operations](#)
- ◆ [Java - Print String to Console](#)
- ◆ [Java - Read String from Console](#)
- ◆ [Java - Concatente two Strings](#)
- ◆ [Java - Check if Strings are Equal](#)
- ◆ [Java - Find Index of First Occurrence of Substring](#)
- ◆ [Java - Find Index of Nth Occurrence of Substring](#)
- ◆ [Java - Replace First Occurrence of Substring](#)
- ⇒ **[Java - Replace All Occurrences of Substring](#)**
- ◆ [Java - Reverse a String](#)
- ◆ [Java - Split String](#)
- ◆ [Java - Trim String](#)