



Split() String method in Java with examples

Difficulty Level : Easy • Last Updated : 25 Sep, 2021

The **string split()** method breaks a given string around matches of the given regular expression. After splitting against the given regular expression, this method returns a char array.

Example:

```
Input String: 016-78967
Regular Expression: -
Output : {"016", "78967"}
```

Following are the two variants of the split() method in Java:

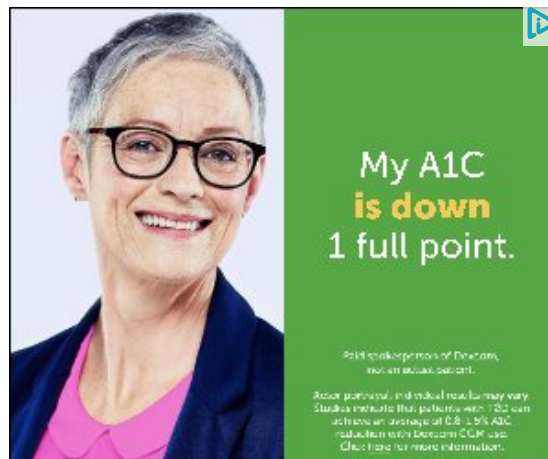
Attention reader! Don't stop learning now. Get hold of all the important **Java Foundation** and Collections concepts with the **Fundamentals of Java and Java Collections Course** at a student-friendly price and become industry ready. To complete your preparation from learning a language to DS Algo and many more, please refer **Complete Interview Preparation Course**.

1. Public String [] split (String regex, int limit)

Parameters:

- regex – a delimiting regular expression
- Limit – the resulting threshold

Returns: An array of strings is computed by splitting the given string.



Throws: *PatternSyntaxException* – if the provided regular expression's syntax is invalid.

The limit parameter can have 3 values:

- **limit > 0** – If this is the case, then the pattern will be applied at most limit-1 times, the resulting array's length will not be more than n, and the resulting array's last entry will contain all input beyond the last matched pattern.
- **limit < 0** – In this case, the pattern will be applied as many times as possible, and the resulting array can be of any size.
- **limit = 0** – In this case, the pattern will be applied as many times as possible, the resulting array can be of any size, and trailing empty strings will be discarded.

Here's how it works:

Let the string that is to be split is – **geekss@for@geekss**

| Regex | Limit | Result |
|-------|-------|-----------------------------|
| @ | 2 | {"geekss", "for@geekss"} |
| @ | 5 | {"geekss", "for", "geekss"} |

| Regex | Limit | Result |
|-------|-------|-----------------------------------|
| @ | -2 | {"geekss", "for", "geekss"} |
| s | 5 | {"geek", "", "@for@geek", "", ""} |
| s | -2 | {"geek", "", "@for@geek", "", ""} |
| s | 0 | {"geek", "", "@for@geek"} |

Following are the Java example codes to demonstrate the working of split()

Example 1:

Java

```
// Java program to demonstrate working of split(regex,
// limit) with small limit.
public class GFG {
    public static void main(String args[])
    {
        String str = "geekss@for@geekss";
        String[] arrOfStr = str.split("@", 2);

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

```
geekss
for@geekss
```

Example 2:

Java

```
// Java program to demonstrate working of split(regex,
// limit) with high limit.
public class GFG {
```

```

public static void main(String args[])
{
    String str = "geekss@for@geekss";
    String[] arrOfStr = str.split("@", 5);

    for (String a : arrOfStr)
        System.out.println(a);
}

```

Output

```

geekss
for
geekss

```

Example 3:



Java

```

// Java program to demonstrate working of split(regex,
// limit) with negative limit.
public class GFG {
    public static void main(String args[])
    {
        String str = "geekss@for@geekss";
        String[] arrOfStr = str.split("@", -2);

        for (String a : arrOfStr)
            System.out.println(a);
    }
}

```

```
}
```

Output

```
geekss
for
geekss
```

Example 4:

Java

```
// Java program to demonstrate working of split(regex,
// limit) with high limit.
public class GFG {
    public static void main(String args[])
    {
        String str = "geekss@for@geekss";
        String[] arrOfStr = str.split("s", 5);

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

```
geek

@for@geek
```

Example 5:

Java

```
// Java program to demonstrate working of split(regex,
// limit) with negative limit.
public class GFG {
    public static void main(String args[])
    {
        String str = "geekss@for@geekss";
        String[] arrOfStr = str.split("s", -2);

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

```
}  
}
```

Output

```
geek
```

```
@for@geek
```

Example 6:

Java

```
// Java program to demonstrate working of split(regex,  
// limit) with 0 limit.  
public class GFG {  
    public static void main(String args[])  
    {  
        String str = "geekss@for@geekss";  
        String[] arrOfStr = str.split("s", 0);  
  
        for (String a : arrOfStr)  
            System.out.println(a);  
    }  
}
```

Output

```
geek
```

```
@for@geek
```

2. public String[] split(String regex)

This variant of the split method takes a regular expression as a parameter and breaks the given string around matches of this regular expression regex. Here, by default limit is 0.

Parameters: *regex* – a delimiting regular expression

Returns: An array of strings is computed by splitting the given string.

Throws: *PatternSyntaxException* – if the provided regular expression's syntax is invalid.

Here are some working example codes:

Example 1:

Java

```
// Java program to demonstrate working of split()
public class GFG {
    public static void main(String args[])
    {
        String str = "GeeksforGeeks:A Computer Science Portal";
        String[] arrOfStr = str.split(":");

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

```
GeeksforGeeks
A Computer Science Portal
```

Example 2:

Java

```
// Java program to demonstrate working of split()
public class GFG {
    public static void main(String args[])
    {
        String str = "GeeksforGeeksforStudents";
        String[] arrOfStr = str.split("for");

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

```
Geeks
Geeks
Students
```

It can be seen in the above example that the pattern/regular expression "for" is applied twice (because "for" is present two times in the string to be split)

Example 3:

Java

```
// Java program to demonstrate working of split()
public class GFG {
    public static void main(String args[])
    {
        String str = "Geeks for Geeks";
        String[] arrOfStr = str.split(" ");

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

```
Geeks
for
Geeks
```


Example 4:

Java

```
// Java program to demonstrate working of split()
public class GFG {
    public static void main(String args[])
    {
        String str = "Geekssss";
        String[] arrOfStr = str.split("s");

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

Geek

In the above example that trailing empty strings are not included in the resulting array arrOfStr.

Example 5:

Java

```
// Java program to demonstrate working of split()
public class GFG {
    public static void main(String args[])
    {
        String str = "GeeksforforGeeksfor ";
        String[] arrOfStr = str.split("for");

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

Geeks

Geeks

In the above example, the trailing spaces (hence not empty string) become a string in the resulting array `arrOfStr`.

Example 6:

Java

```
// Java program to demonstrate working of split()
// using regular expressions
public class GFG {
    public static void main(String args[])
    {
        String str = "word1, word2 word3@word4?word5.word6";
        String[] arrOfStr = str.split("[, ?.@]+");

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```

Output

```
word1
word2
word3
word4
word5
word6
```

In the above example, words are separated whenever either of the characters specified in the set is encountered.

This article is contributed by **Vaibhav Bajpai**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks. Please write comments if you find anything incorrect or you want to share more information about the topic discussed above.



Like

< Previous

length vs length() in Java

Next >

**Java String trim() method
with Example**

RECOMMENDED ARTICLES

Page :

Article Contributed By :



GeeksforGeeks

Vote for difficulty

Current difficulty : [Easy](#)

Easy

Normal

Medium

Hard

Expert

Improved By : [AlokSingh2](#), [nishkarshgandhi](#)

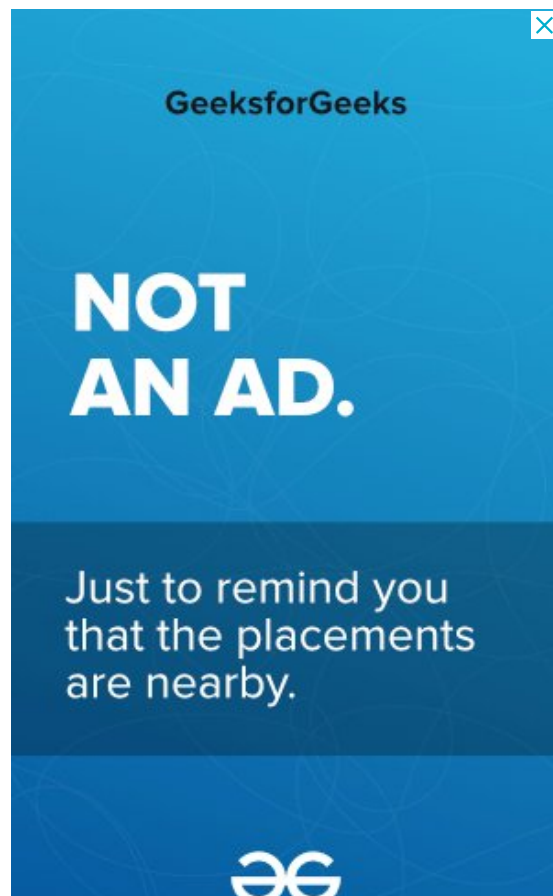
Article Tags : [Java-Functions](#), [Java-lang package](#),
[java-regular-expression](#), [Java-Strings](#), [Java](#)

Practice Tags : [Java-Strings](#), [Java](#)

Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments



Start Preparing

WHAT'S NEW



DSA Live Classes for Working Professionals

[View Details](#)



Competitive Programming Live Classes for Students

[View Details](#)



Complete Interview Preparation Course

[View Details](#)





MOST POPULAR IN JAVA

[Object Oriented Programming \(OOPs\) Concept in Java](#)

[HashMap in Java with Examples](#)

[Initialize an ArrayList in Java](#)

[Multidimensional Arrays in Java](#)

[How to iterate any Map in Java](#)

MORE RELATED ARTICLES IN JAVA

[Stack Class in Java](#)

[Polymorphism in Java](#)

[ArrayList in Java](#)

[LinkedList in Java](#)

[Overriding in Java](#)



 5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

 feedback@geeksforgeeks.org



Company

[About Us](#)

[Careers](#)

[Privacy Policy](#)

Learn

[Algorithms](#)

[Data Structures](#)

[Languages](#)

[Contact Us](#)
[Copyright Policy](#)

Web Development

[Web Tutorials](#)
[HTML](#)
[CSS](#)
[JavaScript](#)
[Bootstrap](#)

[CS Subjects](#)
[Video Tutorials](#)

Contribute

[Write an Article](#)
[Write Interview Experience](#)
[Internships](#)
[Videos](#)