

StringTokenizer methods in Java with Examples | Set 2

StringTokenizer methods in Java with example :

	/	/		\	\
hasMoreToken		nextToken	countTokens	nextElement	
hasMoreElements					

We strongly recommend to refer below post as a prerequisite for this.

[StringTokenizer class in Java with example | Set 1](#)

Following are the StringTokenizer class methods :

- **hasMoreTokens():** The method **java.util.StringTokenizer.hasMoreTokens()** plays role in testing, if tokens are present for the StringTokenizer's string. Those characters that are considered to be delimiters by the StringTokenizer object are changed to characters in the string delimiter. Then the next token to the current position in the string is returned.

Syntax:

```
public boolean hasMoreTokens()
```

Returns: True if and only if next token to the current position in the string exists, else false.

- **nextToken():** The method **java.util.StringTokenizer.nextToken()** returns next token from the given StringTokenizer.

Syntax:

- ```
public String nextToken()
```

- **Return:** the next token from the given StringTokenizer if present.
- **Throws:** NoSuchElementException - if no more token are left.

- **countTokens():** The method **java.util.StringTokenizer.countTokens()** returns total number of tokens present so that we can use nextToken() method before it gives an exception..

**Syntax:**

- ```
public int countTokens()
```

- **Return :** the number of tokens remaining in the string using the current delimiter set.

filter_none

edit

play_arrow

brightness_4

```
// Program in Java illustrating the methods of StringTokenizer class:
// hasMoreToken    nextToken    countTokens
import java.util.*;
public class NewClass
{
    public static void main(String args[])
```

```

{
    String mydelim = " : ";
    String mystr = "JAVA : Code : String : Tokenizer : Geeks";

    // Use of Constructor 2
    // Here we are passing Delimiter - "mydelim"
    StringTokenizer geeks3 =
        new StringTokenizer(mystr, mydelim);

    // Printing count of tokens and tokens
    int count = geeks3.countTokens();
    System.out.println("Number of tokens : " + count + "\n");
    for (int i = 0; i < count; i++)
        System.out.println("token at [" + i + "] : "
            + geeks3.nextToken());

    // .hasMoreTokens() method checks for more Tokens.
    // Here not working as no Tokens left
    while (geeks3.hasMoreTokens())

        // .nextToken is method is returning next token.
        System.out.println(geeks3.nextToken());
}

```

Output:

```

Number of tokens : 5
token at [0] : JAVA
token at [1] : Code
token at [2] : String
token at [3] : Tokenizer
token at [4] : Geeks

```

- **nextElement():** The method **java.util.StringTokenizer.nextElement()** works similar to nextToken except that it returns Object rather than String. Exists so that this class can implement the Enumeration interface.

Syntax:

- **public Object nextElement()**
- **Return:** the next token from the given StringTokenizer.
- **Throws:** NoSuchElementException - if there are no more tokens left.
- **hasMoreElements():** This method **java.util.StringTokenizer.hasMoreElements()** returns same value as hasMoreToken. It exists so that the class can implement the Enumeration interface.

Syntax:

- **public boolean hasMoreElements()**
- **Return:** true if tokens are present in the string, else false

filter_none

edit

play_arrow brightness_4

```
// Program in Java illustrating the methods of StringTokenizer
// class: hasMoreElements, nextElement and nextElement
import java.util.*;
public class NewClass
{
    public static void main(String args[])
    {
        String mydelim = " : ";
        String mystr = "JAVA : Code : String : Tokenizer : Geeks";

        // Use of Constructor 2
        // Here we are passing Delimiter - "mydelim"
        StringTokenizer geeks =
            new StringTokenizer(mystr, mydelim);

        // .countTokens() method counts no. of tokens present.
        int count = geeks.countTokens();
        System.out.println("Number of tokens : " + count);

        // use of hasMoreElements() - true if tokens are present
        while (geeks.hasMoreElements())

            // use of nextElement() - returns the next token
            System.out.println(geeks.nextElement());
    }
}
```

Output:

```
Number of tokens : 5
JAVA
Code
String
Tokenizer
Geeks
```

Important Points:

- counttoken() method is a good alternative in using the combination hasMoreTokens and nextToken().
- The combination of countTokens and nextToken is used if you are interested in the number of tokens also.

References :

<https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html>

This article is contributed by **Mohit Gupta**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.