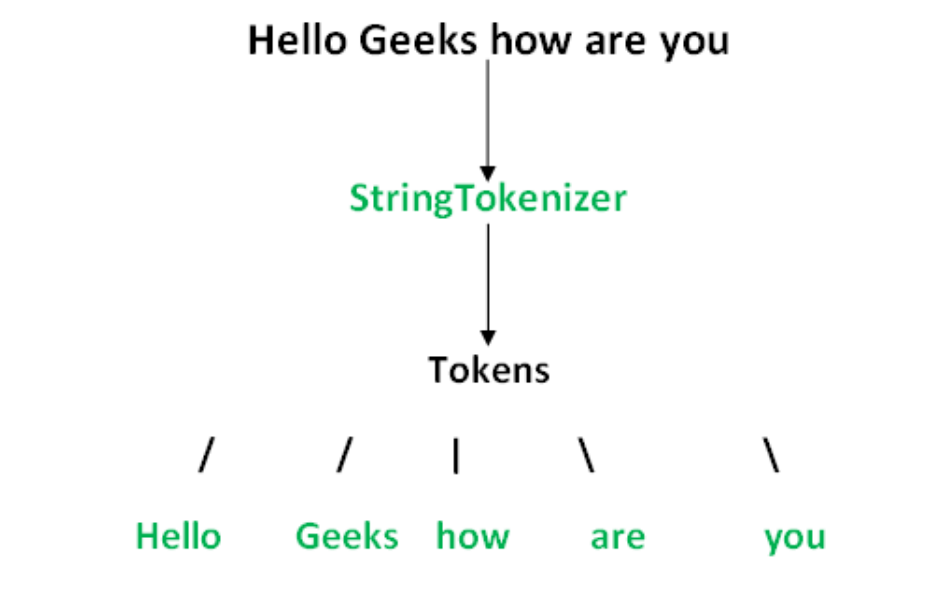
StringTokenizer Class in Java

Difficulty Level : Easy Last Updated : 09 Dec, 2021

StringTokenizer class in Java is used to break a string into tokens. A StringTokenizer object internally maintains a current position within the string to be tokenized. Some operations advance this current position past the characters processed. A token is returned by taking a substring of the string that was used to create the StringTokenizer object.

Illustration:



Constructors of StringToken: Let us consider 'str' is the string to be tokenized

1. **StringTokenizer(String str):** default delimiters like newline, space, tab, carriage return, and form feed.
2. **StringTokenizer(String str, String delim):** delim is a set of delimiters that are used to tokenize the given string.
3. **StringTokenizer(String str, String delim, boolean flag):** The first two parameters have the same meaning wherein The flag
   serves the following purpose.

3.1: If the flag is false, delimiter characters serve to separate tokens

Example:

```
Input : if string --> "hello geeks" and Delimiter is " ", then
Output:  tokens are "hello" and "geeks".
```

3.2: If the flag is true, delimiter characters are considered to be tokens.

Example:

```
Input : String --> is "hello geeks"and Delimiter is " ", then
Output: Tokens --> "hello", " " and "geeks".
```

Methods Of StringTokenizer Class

Method	Action Performed
countTokens()	Returns the total number of tokens present
hasMoreToken()	Tests if tokens are present for the StringTokenizer's string
nextElement()	Returns an Object rather than String
hasMoreElements()	Returns the same value as hasMoreToken
nextToken()	Returns the next token from the given StringTokenizer.

Implementation:

```
// Java Program to Illustrate StringTokenizer Class

// Importing requieed classes
import java.util.*;

// Main class
public class GFG {

    // Main driver method
    public static void main(String args[])
    {

        // Constructor 1
        System.out.println("Using Constructor 1 - ");

        // Creating object of class inside main() method
        StringTokenizer st1 = new StringTokenizer(
```

```
            "Hello Geeks How are you", " ");

        // Condition holds true till there is single token
        // remaining using hasMoreTokens() method
        while (st1.hasMoreTokens())

            // Getting next tokens
            System.out.println(st1.nextToken());

        // Constructor 2
        System.out.println("Using Constructor 2 - ");

        // Again creating object of class inside main()
        // method
        StringTokenizer st2 = new StringTokenizer(
            "JAVA : Code : String", " :");

        // If tokens are present
        while (st2.hasMoreTokens())

            // Print all tokens
            System.out.println(st2.nextToken());

        // Constructor 3
        System.out.println("Using Constructor 3 - ");

        // Again creating object of class inside main()
        // method
        StringTokenizer st3 = new StringTokenizer(
            "JAVA : Code : String", " :", true);

        while (st3.hasMoreTokens())
            System.out.println(st3.nextToken());
    }
}
```

Output

```
Using Constructor 1 -
Hello
Geeks
How
are
you
Using Constructor 2 -
JAVA
Code
```

```
String
Using Constructor 3 -
JAVA

:

Code

:

String
```

This article is contributed by **Mohit Gupta**. 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.