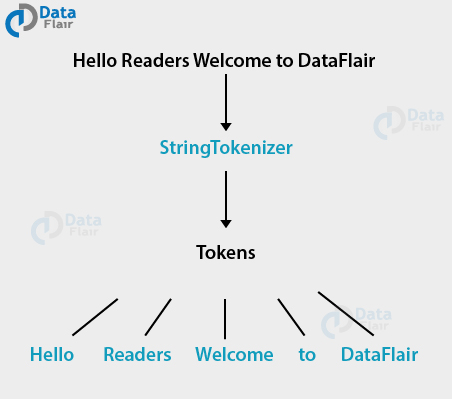
1. **StringTokenizer**

**StringTokenizer in Java** is a class that allows an application to break a string into tokens.

With the help of different StringTokenizer constructors and methods, we can break a string into tokens.

StringTokenizer class is used for creating tokens in Java. It allows an application to break or split into small parts. Each split string part is called Token.



**StringTokenizer Constructors**

* **StringTokenizer(String str)**

str is a string to be tokenized and it considers default delimiters like newline, space, tab, carriage return and form feed which can be further tokenized.

* **StringTokenizer(String str, String delim)**

delim is set of delimiters that are used to tokenize the given string.

* **StringTokenizer(String str, String delim, boolean flag)**

Since the first two parameters have the same meaning. If the flag is false, delimiter characters serve to separate tokens and if the flag is true, delimiter characters are considered to be tokens.

## StringTokenizer Methods

Following are 5 types of ***Methods available in Java*** StringTokenizer:

#### hasMoreTokens()

The method **java.util.StringTokenizer.hasmoreTokens()** plays a role in testing, if tokens are present for the StringTokenizer’s string.

Basically, 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](https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html) if present.

**Throws:** NoSuchElementException – if no more token are left.

#### countTokens()

This method**java.util.StringTokenizer.countTokens()** returns  total number of tokens which are present. Hence, the number further use nextToken() method before it gives an exception and use it.

**Syntax**

public int **countTokens**()

#### nextElement()

This method returns Object rather than String and**java.util.StringTokenizer.nextElements()** works similar to nextToken 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()

Next, in this method **java.util.StringTokenizer.hasMoreElements()** returns same value as hasMoreToken.

**Syntax**

public boolean **hasMoreElements**()

**Return:** True if tokens are present in the string, else false.

**Example program: “StringTokenizerEx.java”**

import java.util.\*;

public class StringTokenizerEx

{

public static void main(String args[])

{

System.out.println("StringTokenizer Constructor 1 - ");

StringTokenizer st1 = new StringTokenizer("Hello students\nWelcome to Java class");

while (st1.hasMoreTokens())

System.out.println(st1.nextToken());

System.out.println("\nStringTokenizer Constructor 2 - ");

StringTokenizer st2 = new StringTokenizer("JAVA : Code : String", " :");

while (st2.hasMoreTokens())

System.out.println(st2.nextToken());

System.out.println("\nStringTokenizer Constructor 3 - ");

StringTokenizer st3 = new StringTokenizer("JAVA\_: Code\_: String\_", "\_: ", true);

System.out.println("Number of tokens=" + st3.countTokens());

while (st3.hasMoreElements())

System.out.println(st3.nextElement());

}

}