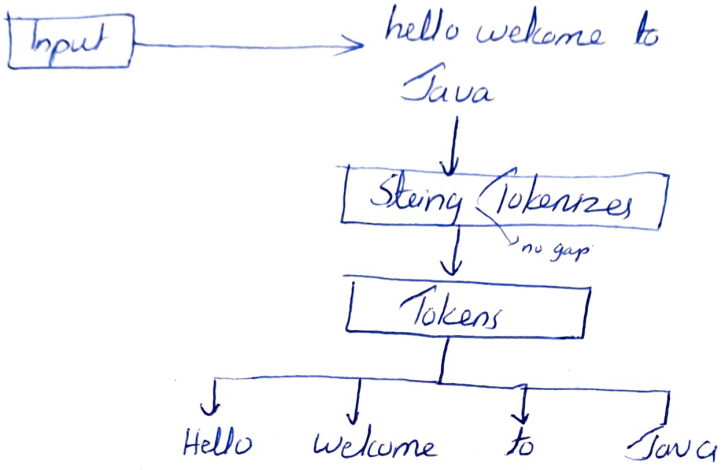
StringTokenizer in JAVA

→ java.util.StringTokenizer class allows you to break string into tokens.
  → it is the simple way to break a string.
→ it doesn't provide the facility to differentiate numbers, quoted strings, identifiers etc. like StreamTokenizer class.
→ In the StringTokenizer class, the delimiters can be provided at the time of creation or one by one to the tokens.



Constructors of StringTokenizer class

sts — is the string to be tokenized.

1. StringTokenizer(String sts);
   → default delimiters like newline, space, tab, carriage return and form feed (lf), used to move the cursor to the start of the next logical page
   → it creates StringTokenizer with specified string
   → This constructor is implemented to perform tokenization of a particular string that is being provided in the parameter.

2. StringTokenizer(String sts, String delimiters);
   → This constructor is implemented to perform string tokenization based on the delimiters provided by the user in the argument.

```
StringTokenizer st = new StringTokenizer("techvidan article on StringTokenizer", " ,");
while (st.hasMoreTokens())
    System.out.println(st.nextToken());
```

o/p
techvidan
article
on
StringTokenizer

3. StringTokenizer(String str, String delimiter, boolean flag)

→ This constructor is implemented to perform string tokenization based on the delimiter and has additional functionality to display delimiter also.

→ if flag == false, delimiter characters serve to seperate tokens

i/p : if string → "hello greeks" and Delimiter is " ", then

o/p : tokens are "hello" and "greeks"

→ if flag == True, delimiter characters are considered to be tokens.

i/p : string → "hello greeks" and Delimiter is " ", then

o/p : Tokens → "hello", " ", and "greeks".

Methods of StringTokenizer class

1. boolean hasMoreTokens() — it checks if there is more tokens available
   → it returns true if more tokens are available in the tokenizer string otherwise returns false.
2. String nextToken() — it returns the next token from the StringTokenizer object.
   (ie, Returns the next token as a string)
3. String nextToken(String delim) — it returns the next token as a string based on the delimiter
4. boolean hasMoreElements() — it is the same as hasMoreTokens() method
   → returns true if one as more tokens remain in the string & returns false if there are none.
5. Object nextElement() — it is the same as nextToken() but its return type is Object.
   (ie, Returns the next token as an object)
6. int countTokens() — it returns the total no. of tokens

→ StringTokenizer implements Enumeration, the hasMoreElements() & nextElement() methods are also implemented. and they act the same as hasMoreTokens() & nextToken() respectively.

```
import java.util.*;
import java.io.*;
public class StringTokenizerExample
{ public static void main(String args[]) throws IOException
  { String myString = "welcome to StringTokenizer Tutorial, article on
                       StringTokenizer class";
    StringTokenizer st = new StringTokenizer(myString); //constructor 1
    int numberOfTokens;
    numberOfTokens = st.countTokens();
    System.out.println("The no. of tokens in this string is :" + numberOfTokens);
    System.out.println("all the token with constructor 1:");
    while(st.hasMoreTokens())
    { System.out.println("next token =" + st.nextToken());
    }
    StringTokenizer st1 = new StringTokenizer(myString, ",");
    System.out.println("all the token with constructor 2:");
    while(st1.hasMoreTokens())
    { System.out.println(st1.nextToken());
    }
    StringTokenizer st2 = new StringTokenizer(myString, ",", true);
    System.out.println("all the token with constructor 3 (delimiter ',') and
                        also printing delimiter:");
    while(st2.hasMoreTokens())
    { System.out.println(st2.nextToken());
    }
  }
```

o/p

```
The no. of tokens in this string
 is : 8
all the token with constructor 1:
next token = welcome
 next token = to
next token = StringTokenizer
next token = Tutorial,
next token = article
next token = on
next token = StringTokenizer
next token = class
```

```
all the token with constructor 2:
Welcome to StringTokenizer Tutorial
article on StringTokenizer class
```

```
all the token with constructor 3
(delimiter ,) and also
printing delimiter:
Welcome to StringTokenizer Tutorial
,
article on StringTokenizer class
```

```
all the token with
constructor 3 without
delimiter:
Welcome to StringTokenizer Tutorial
article on StringTokenizer
class
```

```
StringTokenizer st3 = new StringTokenizer(mystring, ",", false);
System.out.println("all the token with constructor without delimiter:");
while (st3.hasMoreTokens())
{ System.out.println(st3.nextToken());
}
}
}
```

---

eg: String nextToken(String delim)

```
import java.util.*;
public class Test
{ public static void main(String[] args[])
  { StringTokenizer st = new StringTokenizer("my,name,is,khan");
    //printing next token
    System.out.println("Next token is :" + st.nextToken(","));
  }
}
```

o/p

Next token is : my