

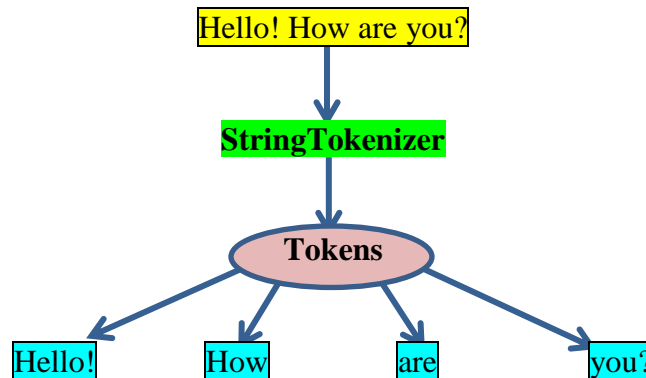
Extraction of words from a given string

Here we will discuss how to extract each word from a given string using the following two concepts:

- **Using StringTokenizer class**
- **Using split() method in Java**

StringTokenizer class and its methods to extract words from a string

The java. util. **StringTokenizer** class allows an application to break a string into tokens.



There are 3 constructors defined in the StringTokenizer class.

StringTokenizer(String str)	creates StringTokenizer with specified string.
StringTokenizer(String str, String delim)	creates StringTokenizer with specified string and delimiter.
StringTokenizer(String str, String delim, boolean returnValue)	creates StringTokenizer with specified string, delimiter and returnValue. If return value is true, delimiter characters are considered to be tokens. If it is false, delimiter characters serve to separate tokens.

There are six methods of StringTokenizer class:

boolean hasMoreTokens()	checks if there is more tokens available.
String nextToken()	returns the next token from the StringTokenizer object.
String nextToken(String delim)	returns the next token based on the delimiter.
boolean hasMoreElements()	same as hasMoreTokens() method.
Object nextElement()	same as nextToken() but its return type is Object.
int countTokens()	returns the total number of tokens.

1. Design a class WordWise to separate words from a sentence and find the frequency of the vowels in each word. [Using StringTokenizer class] Some of the members of the class are given below:

Class name : WordWise

Data members/instance variables:

String str : to store a sentence

Member functions/methods:

- WordWise() : default constructor
- void readsent() : to accept a sentence
- int freq_vowel(String w) : returns the frequency of vowels in the parameterized string w
- void arrange() : displays each word of the sentence in a separate line along with the frequency of vowels for each word by invoking the function freq_vowel()

```
import java.util.*;
class WordWise
{
    String str;
    WordWise() //DEFAULT CONSTRUCTOR
    {
        str=""; //empty string
    }
    void readsent( ) //INPUT METHOD
    {
        Scanner sd=new Scanner(System.in);
        System.out.println("Enter a string");
        str=sd.nextLine( );
    }
    int freq_vowel(String w) //VOWEL CHECK FOR INDIVIDUAL WORD
    {
        int c=0;
        int len=w.length();
        for(int i=0;i<len;i++)
        {
            char ch=Character.toUpperCase(w.charAt(i));
            if(ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')
                c++;
        }
        return c; //RETURNING THE TOTAL NO. OF VOWELS
    }
    void arrange( )
    {
        StringTokenizer st=new StringTokenizer(str);
        int ct=st.countTokens(); //COUNTING THE TOTAL NO. OF TOKENS
        for(int i=1; i<=ct; i++)
        {
            String w=st.nextToken();
            System.out.println(w+"\t"+freq_vowel(w)); //PASSING THE
                                                    //EXTRACTED WORDS
        }
    }
}
```

Split() method of String class

The split() method of String class breaks a given string around matches of the given regular expression (**delimiter**) and returns **an array of strings** holding each word separately. Delimiter is a character designated as separator to split the given string. By default it is a space and denoted by '\s'

For Example:

Input String: "Today is Monday" (here space is the delimiter)

Output: {"Today", "is", "Monday"}

Input String: "Today@is@Monday" and Regular expression: @

Output: {"Today", "is", "Monday"}

Input String: "016-789-67" and Regular expression: -

Output: {"016", "789", "67"}

Syntax:

public String [] split (String R)

- R is a **delimiting regular expression**
- The function will return an array of strings computed by the splitting the given string at the delimiting factor.

Example:

```
String str="Today is Monday AND IT@IS@HOLIDAY";
```

```
String arr1[]=str.split("\s");
```

```
Output: {"Today", "is", "Monday", "AND", "IT@IS@HOLIDAY"}
```

```
String arr2[]=str.split("@");
```

```
Output: {"Today is Monday AND", "IT", "IS", "HOLIDAY"}
```

public String [] split (String R, int L)

- R is a delimiting regular expression
- L is the result threshold (limiting the size of the array, it will be always L-1)
- The function will return an array of strings computed by the splitting the given string at the delimiting factor.

```
String str="Today@is@Monday@AND@HOLIDAY";
```

```
String arr[]=str.split("@",4);
```

```
Output: {"Today", "is", "Monday@AND@HOLIDAY"}
```

L>0 – it means the array length will not be more than n and the last entry will contain all the inputs beyond the last matched pattern

L<0 - it means the array length can be of any size, including the trailing spaces

L=0 - it means the array length can be of any size but the trailing spaces from the string array will be discarded

Default delimiter is space and is denoted by '\s'

Note: If the delimiting factor syntax is invalid then the function will return **PatternSyntaxException**

2. Design a class named SortWords that will accept a string and rearrange all the words in Alphabetical order. [Using split() method] If the string contains non alphabets then they will be placed in front. For example, if the input is TODAY'S DATE IS 6.7.2020 AND TODAY IS FRIDAY, then the output will be like this:

Original string: TODAY'S DATE IS 6.7.2020 AND TODAY IS FRIDAY

Changed string: 6.7.2020 AND DATE FRIDAY IS IS TODAY TODAY'S

```
import java.util.Scanner;
class SortWords
{
    String str;
    String arr[];
    int ln;
    public void accept()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a string with all letters in Caps:");
        str=sc.nextLine().toUpperCase();
    }
    public void splitWords()//extracting each word and rearranging them in alphabetical order
    {
        arr=str.split("\\s"); //each word will be stored in the array
        ln=arr.length;
        String tm;
        for(int i=0;i<ln-1;i++)
        {
            for(int j=i+1;j<ln;j++)
            {
                if(arr[i].compareTo(arr[j])>0)
                {
                    tm=arr[i];
                    arr[i]=arr[j];
                    arr[j]=tm;
                }
            }
        }
    }
    public void display()
    {
        System.out.println("Original string:"+str);
        System.out.print("Changed string:");
        for(int i=0;i<ln;i++)
            System.out.print(arr[i]+" ");
    }
    public static void main(String ar[])
    {
        SortWords sw=new SortWords();
        sw.accept();
        sw.splitWords();
        sw.display();
    }
}
```