

## La Classe StringTokenizer

Jusqu'à maintenant, nous avons surtout travaillé avec des mots. Une chaîne de caractères (String) représente aussi une phrase. La classe **StringTokenizer** est très utile pour **décomposer une phrase en morceaux** (comme les mots par exemple). Le StringTokenizer utilise un **séparateur** (le caractère d'espace) pour déterminer quels sont les morceaux.

Pour utiliser cette classe il faut ajouter au tout début du programme l'instruction :  
**import java.util.\*;**

Un StringTokenizer nous permet de parcourir une phrase et d'en utiliser les morceaux un par un.

<code>String phraseAdécouper = "Ceci, est une phrase"</code>  <code>StringTokenizer mots = new StringTokenizer( phraseAdécouper);</code>	séparer la phrase en mots
<code>int compte ;</code> <code>compte = mots.countTokens()</code> détermine le <i>nombre de mots</i> dans la phrase initiale	<b>compte =4</b>
<code>String s1 ;</code> <code>s1 = mots.nextTokens() ;</code> //renvoie le prochain mot dans la phrase  <code>s1 = mots.nextTokens() ;</code> //renvoie le prochain mot dans la phrase	<b>s1 = "Ceci"</b>  <b>s1 = "est"</b>
<code>mots.hasMoreTokens()</code>  détermine s'il y a <i>encore des mots</i> dans la phrase initiale	<b>true</b>

L'exemple ci-dessous affiche tous les mots (un par ligne) dans une phrase :

```
/* Mots.JAVA
 * Affiche les mots d'une phrase un sous l'autre
 * Auteur: votre nom
 * Date: 04-03-30
 */
import java.util.*;
.*;
public class Mots {
    public static void main(String[] args) {

        // déclaration de variables
        String texte = "Bonjour tout le monde!";
        StringTokenizer morceaux = new StringTokenizer(texte) ;

        //Traitement des données
        while (morceaux.hasMoreTokens()) //Regarde s'il reste des
                                         //morceaux dans la phrase
        { // Passe au morceau suivant et affiche le mot
            System.out.println(" - "+morceaux.nextToken());
        }
    }
}
```

**Affichage :**

```
- Bonjour
- tout
- le
- monde!
```