String Tokeniger

- The java.util.StringTokenizer class is used to break strings into tokens (words, numbers, operators, or whatever).
- Has been replaced by regular expression tools.
  - A more powerful solution is to use regular expressions, and the easiest way to do that is use the java.util.Scaner class, the String split(..) method, or the Pattern and Matcher classes

- A StringTokenizer constructor takes a string to break into tokens and returns a StringTokenizer object for that string.
- Each time its nextToken() method is called, it returns the next token in that string.
- If you don't specify the delimiters (separator characters), blanks are the default.

Constructors

- StringTokenizer st = new StringTokenizer(s);
  - Creates a StringTokenizer for the String s that uses whitespace (blanks, tabs, newlines, returns, form feeds) as delimiters.
- StringTokenizer st = new StringTokenizer(s, d);
  - Creates a StringTokenizer for the String s using delimiters from the String d.
- StringTokenizer st = new StringTokenizer(s, d, f);
  - Creates a StringTokenizer for the String s using delimiters from the String d.
  - If the boolean f is true, each delimiter character will also be returned as a token.

Common Methods

Assume that st is a StringTokenizer

- st.hasMoreTokens() -- Returns true if there are more tokens.
- st.nextToken() -- Returns the next token as a String.
- st.countTokens() –
  - Returns the int number of tokens.
  - This can be used to allocate an array before starting
  - it can be inefficient for long strings because it has to scan the string once just to get this number.
  - Using a Vector and converting it to an array at the end may be a better choice.

Example: Find the longest word in a String

```
import java.util.StringTokenizer;;


public class StringToken {


static String s = "Batch=Java: HCL FRESHER" + "Vendor=BlueLotus;";


public static void main(String args[]) {
StringTokenizer st = new StringTokenizer(s, "=;");
while (st.hasMoreTokens()) {
String key = st.nextToken();
System.out.println(key);
}
}
}
```