| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BreakIterator.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/text/Bidi.html)   [**NEXT CLASS**](http://docs.google.com/java/text/CharacterIterator.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/BreakIterator.html)    [**NO FRAMES**](http://docs.google.com/BreakIterator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#1t3h5sf) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#3rdcrjn) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

## **java.text**

Class BreakIterator

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.text.BreakIterator**

**All Implemented Interfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

public abstract class **BreakIterator**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

The BreakIterator class implements methods for finding the location of boundaries in text. Instances of BreakIterator maintain a current position and scan over text returning the index of characters where boundaries occur. Internally, BreakIterator scans text using a CharacterIterator, and is thus able to scan text held by any object implementing that protocol. A StringCharacterIterator is used to scan String objects passed to setText.

You use the factory methods provided by this class to create instances of various types of break iterators. In particular, use getWordIterator, getLineIterator, getSentenceIterator, and getCharacterIterator to create BreakIterators that perform word, line, sentence, and character boundary analysis respectively. A single BreakIterator can work only on one unit (word, line, sentence, and so on). You must use a different iterator for each unit boundary analysis you wish to perform.

Line boundary analysis determines where a text string can be broken when line-wrapping. The mechanism correctly handles punctuation and hyphenated words. Actual line breaking needs to also consider the available line width and is handled by higher-level software.

Sentence boundary analysis allows selection with correct interpretation of periods within numbers and abbreviations, and trailing punctuation marks such as quotation marks and parentheses.

Word boundary analysis is used by search and replace functions, as well as within text editing applications that allow the user to select words with a double click. Word selection provides correct interpretation of punctuation marks within and following words. Characters that are not part of a word, such as symbols or punctuation marks, have word-breaks on both sides.

Character boundary analysis allows users to interact with characters as they expect to, for example, when moving the cursor through a text string. Character boundary analysis provides correct navigation through character strings, regardless of how the character is stored. The boundaries returned may be those of supplementary characters, combining character sequences, or ligature clusters. For example, an accented character might be stored as a base character and a diacritical mark. What users consider to be a character can differ between languages.

The BreakIterator instances returned by the factory methods of this class are intended for use with natural languages only, not for programming language text. It is however possible to define subclasses that tokenize a programming language.

**Examples**:

Creating and using text boundaries:

public static void main(String args[]) {  
 if (args.length == 1) {  
 String stringToExamine = args[0];  
 //print each word in order  
 BreakIterator boundary = BreakIterator.getWordInstance();  
 boundary.setText(stringToExamine);  
 printEachForward(boundary, stringToExamine);  
 //print each sentence in reverse order  
 boundary = BreakIterator.getSentenceInstance(Locale.US);  
 boundary.setText(stringToExamine);  
 printEachBackward(boundary, stringToExamine);  
 printFirst(boundary, stringToExamine);  
 printLast(boundary, stringToExamine);  
 }  
 }

Print each element in order:

public static void printEachForward(BreakIterator boundary, String source) {  
 int start = boundary.first();  
 for (int end = boundary.next();  
 end != BreakIterator.DONE;  
 start = end, end = boundary.next()) {  
 System.out.println(source.substring(start,end));  
 }  
 }

Print each element in reverse order:

public static void printEachBackward(BreakIterator boundary, String source) {  
 int end = boundary.last();  
 for (int start = boundary.previous();  
 start != BreakIterator.DONE;  
 end = start, start = boundary.previous()) {  
 System.out.println(source.substring(start,end));  
 }  
 }

Print first element:

public static void printFirst(BreakIterator boundary, String source) {  
 int start = boundary.first();  
 int end = boundary.next();  
 System.out.println(source.substring(start,end));  
 }

Print last element:

public static void printLast(BreakIterator boundary, String source) {  
 int end = boundary.last();  
 int start = boundary.previous();  
 System.out.println(source.substring(start,end));  
 }

Print the element at a specified position:

public static void printAt(BreakIterator boundary, int pos, String source) {  
 int end = boundary.following(pos);  
 int start = boundary.previous();  
 System.out.println(source.substring(start,end));  
 }

Find the next word:

public static int nextWordStartAfter(int pos, String text) {  
 BreakIterator wb = BreakIterator.getWordInstance();  
 wb.setText(text);  
 int last = wb.following(pos);  
 int current = wb.next();  
 while (current != BreakIterator.DONE) {  
 for (int p = last; p < current; p++) {  
 if (Character.isLetter(text.codePointAt(p)))  
 return last;  
 }  
 last = current;  
 current = wb.next();  
 }  
 return BreakIterator.DONE;  
 }

(The iterator returned by BreakIterator.getWordInstance() is unique in that the break positions it returns don't represent both the start and end of the thing being iterated over. That is, a sentence-break iterator returns breaks that each represent the end of one sentence and the beginning of the next. With the word-break iterator, the characters between two boundaries might be a word, or they might be the punctuation or whitespace between two words. The above code uses a simple heuristic to determine which boundary is the beginning of a word: If the characters between this boundary and the next boundary include at least one letter (this can be an alphabetical letter, a CJK ideograph, a Hangul syllable, a Kana character, etc.), then the text between this boundary and the next is a word; otherwise, it's the material between words.)

**See Also:**[CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html)

| **Field Summary** | |
| --- | --- |
| static int | [**DONE**](http://docs.google.com/java/text/BreakIterator.html#DONE)            DONE is returned by previous(), next(), next(int), preceding(int) and following(int) when either the first or last text boundary has been reached. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**BreakIterator**](http://docs.google.com/java/text/BreakIterator.html#BreakIterator())()            Constructor. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/text/BreakIterator.html#clone())()            Create a copy of this iterator |
| abstract  int | [**current**](http://docs.google.com/java/text/BreakIterator.html#current())()            Returns character index of the text boundary that was most recently returned by next(), next(int), previous(), first(), last(), following(int) or preceding(int). |
| abstract  int | [**first**](http://docs.google.com/java/text/BreakIterator.html#first())()            Returns the first boundary. |
| abstract  int | [**following**](http://docs.google.com/java/text/BreakIterator.html#following(int))(int offset)            Returns the first boundary following the specified character offset. |
| static [Locale](http://docs.google.com/java/util/Locale.html)[] | [**getAvailableLocales**](http://docs.google.com/java/text/BreakIterator.html#getAvailableLocales())()            Returns an array of all locales for which the get\*Instance methods of this class can return localized instances. |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getCharacterInstance**](http://docs.google.com/java/text/BreakIterator.html#getCharacterInstance())()            Returns a new BreakIterator instance for [character breaks](#3dy6vkm) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()). |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getCharacterInstance**](http://docs.google.com/java/text/BreakIterator.html#getCharacterInstance(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a new BreakIterator instance for [character breaks](#3dy6vkm) for the given locale. |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getLineInstance**](http://docs.google.com/java/text/BreakIterator.html#getLineInstance())()            Returns a new BreakIterator instance for [line breaks](#3znysh7) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()). |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getLineInstance**](http://docs.google.com/java/text/BreakIterator.html#getLineInstance(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a new BreakIterator instance for [line breaks](#3znysh7) for the given locale. |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getSentenceInstance**](http://docs.google.com/java/text/BreakIterator.html#getSentenceInstance())()            Returns a new BreakIterator instance for [sentence breaks](#2et92p0) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()). |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getSentenceInstance**](http://docs.google.com/java/text/BreakIterator.html#getSentenceInstance(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a new BreakIterator instance for [sentence breaks](#2et92p0) for the given locale. |
| abstract  [CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) | [**getText**](http://docs.google.com/java/text/BreakIterator.html#getText())()            Get the text being scanned |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getWordInstance**](http://docs.google.com/java/text/BreakIterator.html#getWordInstance())()            Returns a new BreakIterator instance for [word breaks](#tyjcwt) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()). |
| static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) | [**getWordInstance**](http://docs.google.com/java/text/BreakIterator.html#getWordInstance(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) locale)            Returns a new BreakIterator instance for [word breaks](#tyjcwt) for the given locale. |
| boolean | [**isBoundary**](http://docs.google.com/java/text/BreakIterator.html#isBoundary(int))(int offset)            Returns true if the specified character offset is a text boundary. |
| abstract  int | [**last**](http://docs.google.com/java/text/BreakIterator.html#last())()            Returns the last boundary. |
| abstract  int | [**next**](http://docs.google.com/java/text/BreakIterator.html#next())()            Returns the boundary following the current boundary. |
| abstract  int | [**next**](http://docs.google.com/java/text/BreakIterator.html#next(int))(int n)            Returns the nth boundary from the current boundary. |
| int | [**preceding**](http://docs.google.com/java/text/BreakIterator.html#preceding(int))(int offset)            Returns the last boundary preceding the specified character offset. |
| abstract  int | [**previous**](http://docs.google.com/java/text/BreakIterator.html#previous())()            Returns the boundary preceding the current boundary. |
| abstract  void | [**setText**](http://docs.google.com/java/text/BreakIterator.html#setText(java.text.CharacterIterator))([CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) newText)            Set a new text for scanning. |
| void | [**setText**](http://docs.google.com/java/text/BreakIterator.html#setText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) newText)            Set a new text string to be scanned. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### DONE

public static final int **DONE**

DONE is returned by previous(), next(), next(int), preceding(int) and following(int) when either the first or last text boundary has been reached.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.text.BreakIterator.DONE)

| **Constructor Detail** |
| --- |

### BreakIterator

protected **BreakIterator**()

Constructor. BreakIterator is stateless and has no default behavior.

| **Method Detail** |
| --- |

### clone

public [Object](http://docs.google.com/java/lang/Object.html) **clone**()

Create a copy of this iterator

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**A copy of this**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### first

public abstract int **first**()

Returns the first boundary. The iterator's current position is set to the first text boundary.

**Returns:**The character index of the first text boundary.

### last

public abstract int **last**()

Returns the last boundary. The iterator's current position is set to the last text boundary.

**Returns:**The character index of the last text boundary.

### next

public abstract int **next**(int n)

Returns the nth boundary from the current boundary. If either the first or last text boundary has been reached, it returns BreakIterator.DONE and the current position is set to either the first or last text boundary depending on which one is reached. Otherwise, the iterator's current position is set to the new boundary. For example, if the iterator's current position is the mth text boundary and three more boundaries exist from the current boundary to the last text boundary, the next(2) call will return m + 2. The new text position is set to the (m + 2)th text boundary. A next(4) call would return BreakIterator.DONE and the last text boundary would become the new text position.

**Parameters:**n - which boundary to return. A value of 0 does nothing. Negative values move to previous boundaries and positive values move to later boundaries. **Returns:**The character index of the nth boundary from the current position or BreakIterator.DONE if either first or last text boundary has been reached.

### next

public abstract int **next**()

Returns the boundary following the current boundary. If the current boundary is the last text boundary, it returns BreakIterator.DONE and the iterator's current position is unchanged. Otherwise, the iterator's current position is set to the boundary following the current boundary.

**Returns:**The character index of the next text boundary or BreakIterator.DONE if the current boundary is the last text boundary. Equivalent to next(1).**See Also:**[next(int)](http://docs.google.com/java/text/BreakIterator.html#next(int))

### previous

public abstract int **previous**()

Returns the boundary preceding the current boundary. If the current boundary is the first text boundary, it returns BreakIterator.DONE and the iterator's current position is unchanged. Otherwise, the iterator's current position is set to the boundary preceding the current boundary.

**Returns:**The character index of the previous text boundary or BreakIterator.DONE if the current boundary is the first text boundary.

### following

public abstract int **following**(int offset)

Returns the first boundary following the specified character offset. If the specified offset equals to the last text boundary, it returns BreakIterator.DONE and the iterator's current position is unchanged. Otherwise, the iterator's current position is set to the returned boundary. The value returned is always greater than the offset or the value BreakIterator.DONE.

**Parameters:**offset - the character offset to begin scanning. **Returns:**The first boundary after the specified offset or BreakIterator.DONE if the last text boundary is passed in as the offset. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified offset is less than the first text boundary or greater than the last text boundary.

### preceding

public int **preceding**(int offset)

Returns the last boundary preceding the specified character offset. If the specified offset equals to the first text boundary, it returns BreakIterator.DONE and the iterator's current position is unchanged. Otherwise, the iterator's current position is set to the returned boundary. The value returned is always less than the offset or the value BreakIterator.DONE.

**Parameters:**offset - the characater offset to begin scanning. **Returns:**The last boundary before the specified offset or BreakIterator.DONE if the first text boundary is passed in as the offset. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified offset is less than the first text boundary or greater than the last text boundary.**Since:** 1.2

### isBoundary

public boolean **isBoundary**(int offset)

Returns true if the specified character offset is a text boundary.

**Parameters:**offset - the character offset to check. **Returns:**true if "offset" is a boundary position, false otherwise.**Since:** 1.2

### current

public abstract int **current**()

Returns character index of the text boundary that was most recently returned by next(), next(int), previous(), first(), last(), following(int) or preceding(int). If any of these methods returns BreakIterator.DONE because either first or last text boundary has been reached, it returns the first or last text boundary depending on which one is reached.

**Returns:**The text boundary returned from the above methods, first or last text boundary.**See Also:**[next()](http://docs.google.com/java/text/BreakIterator.html#next()), [next(int)](http://docs.google.com/java/text/BreakIterator.html#next(int)), [previous()](http://docs.google.com/java/text/BreakIterator.html#previous()), [first()](http://docs.google.com/java/text/BreakIterator.html#first()), [last()](http://docs.google.com/java/text/BreakIterator.html#last()), [following(int)](http://docs.google.com/java/text/BreakIterator.html#following(int)), [preceding(int)](http://docs.google.com/java/text/BreakIterator.html#preceding(int))

### getText

public abstract [CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) **getText**()

Get the text being scanned

**Returns:**the text being scanned

### setText

public void **setText**([String](http://docs.google.com/java/lang/String.html) newText)

Set a new text string to be scanned. The current scan position is reset to first().

**Parameters:**newText - new text to scan.

### setText

public abstract void **setText**([CharacterIterator](http://docs.google.com/java/text/CharacterIterator.html) newText)

Set a new text for scanning. The current scan position is reset to first().

**Parameters:**newText - new text to scan.

### getWordInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getWordInstance**()

Returns a new BreakIterator instance for [word breaks](#tyjcwt) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()).

**Returns:**A break iterator for word breaks

### getWordInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getWordInstance**([Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a new BreakIterator instance for [word breaks](#tyjcwt) for the given locale.

**Parameters:**locale - the desired locale **Returns:**A break iterator for word breaks **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if locale is null

### getLineInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getLineInstance**()

Returns a new BreakIterator instance for [line breaks](#3znysh7) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()).

**Returns:**A break iterator for line breaks

### getLineInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getLineInstance**([Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a new BreakIterator instance for [line breaks](#3znysh7) for the given locale.

**Parameters:**locale - the desired locale **Returns:**A break iterator for line breaks **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if locale is null

### getCharacterInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getCharacterInstance**()

Returns a new BreakIterator instance for [character breaks](#3dy6vkm) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()).

**Returns:**A break iterator for character breaks

### getCharacterInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getCharacterInstance**([Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a new BreakIterator instance for [character breaks](#3dy6vkm) for the given locale.

**Parameters:**locale - the desired locale **Returns:**A break iterator for character breaks **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if locale is null

### getSentenceInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getSentenceInstance**()

Returns a new BreakIterator instance for [sentence breaks](#2et92p0) for the [default locale](http://docs.google.com/java/util/Locale.html#getDefault()).

**Returns:**A break iterator for sentence breaks

### getSentenceInstance

public static [BreakIterator](http://docs.google.com/java/text/BreakIterator.html) **getSentenceInstance**([Locale](http://docs.google.com/java/util/Locale.html) locale)

Returns a new BreakIterator instance for [sentence breaks](#2et92p0) for the given locale.

**Parameters:**locale - the desired locale **Returns:**A break iterator for sentence breaks **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if locale is null

### getAvailableLocales

public static [Locale](http://docs.google.com/java/util/Locale.html)[] **getAvailableLocales**()

Returns an array of all locales for which the get\*Instance methods of this class can return localized instances. The returned array represents the union of locales supported by the Java runtime and by installed [BreakIteratorProvider](http://docs.google.com/java/text/spi/BreakIteratorProvider.html) implementations. It must contain at least a Locale instance equal to [Locale.US](http://docs.google.com/java/util/Locale.html#US).

**Returns:**An array of locales for which localized BreakIterator instances are available.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BreakIterator.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/text/Bidi.html)   [**NEXT CLASS**](http://docs.google.com/java/text/CharacterIterator.html) | [**FRAMES**](http://docs.google.com/index.html?java/text/BreakIterator.html)    [**NO FRAMES**](http://docs.google.com/BreakIterator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#1t3h5sf) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#3rdcrjn) | [CONSTR](#lnxbz9) | [METHOD](#1ksv4uv) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).