

java.util.regex.*

**Pattern**

Static Methods

Pattern compile (String regex)

Pattern compile (String regex, int flags)

boolean matches (String regex, CharSequence input)

Other Public Methods

int flags ()


Matcher matcher (CharSequence input)

String pattern ()

String[] split (CharSequence input)

String[] split (CharSequence input, int limit)

int UNIX_LINES, CASE_INSENSITIVE, COMMENTS, MULTILINE, DOTALL, UNICODE_CASE, CANON_EQ

**Matcher**

Collectors

Matcher appendReplacement (StringBuffer sb, String replacement)

StringBuffer appendTail (StringBuffer sb)

Other Public Methods

int end ()

int end (int group)

boolean find ()

boolean find (int start)

String group ()

String group (int group)

int groupCount ()

boolean lookingAt ()

boolean matches ()

Pattern pattern ()

String replaceAll (String replacement)

String replaceFirst (String replacement)

Matcher reset ()

Matcher reset (CharSequence input)

int start ()

int start (int group)

Typical invocation:

```
Pattern p = Pattern.compile("a(b)");
Matcher m = p.matcher("aaaabab");
boolean b = m.find();
if (b) {
    int start = m.start();
    String s = m.replaceFirst ("foo$1");
    ...
}
```

Line terminators:

- A newline (line feed) character ('\n '),
- A carriage-return character followed immediately by a newline character ('\r\n '),
- A standalone carriage-return character ('\r '),
- A next-line character ('\u0085 '),
- A line-separator character ('\u2028 '), or
- A paragraph-separator character ('\u2029 ').

Modes:

UNIX_LINES mode (?d): only newline characters are line terminators.

DOTALL mode (?s): The regular expression . also matches a line terminator.

MULTILINE mode (?m): the regular expressions ^ and \$ not only match at the beginning and the end, respectively, of the entire input sequence, but also after or just before, respectively, a line terminator.

CASE_INSENSITIVE mode (?i)

Characters	
x	The character x
\\	The backslash character
\xhh	The character with hexadecimal value 0xhh
\uhhhh	The character with hexadecimal value 0xhhhh
\t	The tab character ('\u0009 ')
\n	The newline (line feed) character ('\u000a ')
\r	The carriage-return character ('\u000d ')
\f	The form-feed character ('\u000c ')
\a	The alert (bell) character ('\u0007 ')
\e	The escape character ('\u001B ')
\cx	The control character corresponding to x

Character classes	
[abc]	a, b, or c (simple class)
[^abc]	Any character except a, b, or c (negation)
[a-zA-Z]	a through z or A through Z, inclusive (range)

Predefined character classes	
.	Any character (may or may not match line terminators)
\d	A digit: [0-9]
\D	A non-digit: [^0-9]
\s	A whitespace character: [\t\n\r0B\F\]
\S	A non-whitespace character: [^\s]
\w	A word character: [a-zA-Z_0-9]
\W	A non-word character: [^\w]

Boundary matchers	
^	The beginning of a line
\$	The end of a line
\b	A word boundary
\B	A non-word boundary
\A	The beginning of the input
\G	The end of the previous match
\Z	The end of the input but for the final terminator, if any
\z	The end of the input

Greedy quantifiers	
X?	X, once or not at all
X*	X, zero or more times
X+	X, one or more times
X{n}	X, exactly n times
X{n, }	X, at least n times
X{n, m}	X, at least n but not more than m times

Logical operators	
XY	X followed by Y
X Y	Either X or Y
(X)	X, as a capturing group

Back references	
\n	Whatever the n th capturing group matched