

Username: Pralay Patoria **Book:** The C++ Standard Library: A Tutorial and Reference, Second Edition. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

14.8. The Regex ECMAScript Grammar

The default grammar of the regex library is a “modified ECMAScript” grammar (see [ECMAScript](#)), which is a much more powerful grammar than all the other grammars available. [Table 14.3](#) lists the most important special expressions with their meanings.

Table 14.3. Common Regex Expressions for the Default (ECMAScript) Grammar

Expression	Meaning
.	Any character except newline
[...]	One of the characters ... (may contain ranges)
[^...]	None of the characters ... (may contain ranges)
[[: <i>charclass</i> :]]	A character of the specified character class <i>charclass</i> (see Table 14.4)
\n, \t, \f, \r, \v	A newline, tabulator, form feed, carriage return, or vertical tab
\xhh, \uhhh	A hexadecimal or Unicode character
\d, \D, \s, \S, \w, \W	A shortcut for a character of a character class (see Table 14.4)
*	The previous character or group any times
?	The previous character or group optional (none or one times)
+	The previous character or group at least one time
{ <i>n</i> }	The previous character or group <i>n</i> times
{ <i>n</i> , }	The previous character or group at least <i>n</i> times
{ <i>n</i> , <i>m</i> }	The previous character or group at least <i>n</i> and at most <i>m</i> times
... ...	The pattern before or the pattern after
(...)	Grouping
\1, \2, \3, ...	The <i>n</i> th group (first group has index 1)
\b	A positive word boundary (beginning or end of a word)
\B	A negative word boundary (no beginning or end of a word)
^	The beginning of a line (includes beginning of all characters)
\$	The end of a line (includes end of all characters)

Inside the bracket expressions, you can specify any combination of characters (including special characters), character ranges (for example, `[0-9a-z]`), and character classes (for example, `[[:digit:]]`). A leading `^` negates the whole expression, so the whole bracket expression means “any character except ...”. [Table 14.4](#) lists the possible character classes of regular expressions. Note that the basic classes correspond to the convenience functions for character classifications in [Section 16.4.4, page 895](#). However, the one-letter shortcuts are supported only by regular expressions. The character class escape sequences are supported only by the ECMAScript grammar.

Table 14.4. Character Classes and Corresponding Escape Sequences (ECMAScript)

Character Class	Shortcut	Esc.	Effect
<code>[[:alnum:]]</code>			A letter or a digit (equivalent to <code>[[:alpha:]][[:digit:]]</code>)
<code>[[:alpha:]]</code>			A letter
<code>[[:blank:]]</code>			A space or a tab
<code>[[:cntrl:]]</code>			A control character
<code>[[:digit:]]</code>	<code>[[:d:]]</code>	<code>\d</code>	A digit
		<code>\D</code>	Not a digit (equivalent to <code>^[[:digit:]]</code>)
<code>[[:graph:]]</code>			A printable, nonspace character (equivalent to <code>[[:alnum:]][[:punct:]]</code>)
<code>[[:lower:]]</code>			A lowercase letter
<code>[[:print:]]</code>			A printable character (including whitespaces)
<code>[[:punct:]]</code>			A punctuation character (that is, it is printable but is not a space, digit, or letter)
<code>[[:space:]]</code>	<code>[[:s:]]</code>	<code>\s</code>	A space character
		<code>\S</code>	Not a space character (equivalent to <code>^[[:space:]]</code>)
<code>[[:upper:]]</code>			An uppercase letter
<code>[[:xdigit:]]</code>			A hexadecimal digit
	<code>[[:w:]]</code>	<code>\w</code>	A letter, digit, or underscore (equivalent to <code>[[:alpha:]][[:digit:]]_</code>)
		<code>\W</code>	Not a letter or a digit or an underscore (equivalent to <code>^[[:alpha:]][[:digit:]]_</code>)

Here are some examples:

[Click here to view code image](#)

```

[_[[:alpha:]][_[:alnum:]]*           // a C++ identifier
(.\|\\n)*                          // any number of any character (including
newlines)
[123]?[0-9]\\.[1]?[0-9]\\.[0-9]{2} // a date in the first century of 2000
                                   // (German format, for example 24.12.2010)

```