

Regexp:	Matches:
"a.b"	a followed by any character followed by b.
"a\\.b"	a followed by a literal period followed by b.
"ac?b"	a followed by at most one c followed by b; thus, "ab" or "acb" but not "ac" or "adb".
"ac*b"	a followed by any number of instances of c, followed by b; thus, "ab" or "acb" or "acccb" but not "ac" or "adb".
"ac+b"	a followed by one or more instances of c, followed by b; thus, "acb" or "accb" but not "ab" or "ac" or "adb".
"a[xyz]b"	a followed by any of the characters x or y or z, followed by b; thus, "axb" or "ayb" or "azb" but not "ab" or "aab".
"a[x0-9]b"	a followed by either x or characters in the range 0-9, followed by b; thus, "axb" or "a0b" or "a4b" but not "ab" or "aab".
"a[^xyz]b"	a followed by any character that is not x or y or z, followed by b; thus, "adb" or "aeb" but not "axb" or "ayb" or "azb".
"a[^x0-9]b"	a followed by any character that is not x or in the range 0-9, followed by b; thus, "adb" or "aeb" but not "axb" or "a0b" or "a4b".
"^a"	a at the beginning of a string; thus, "acb" or "accb" but not "bcb" or "bab".
"a\$"	a at the end of a string; thus, "bca" or "ba" but not "bac" or "cab".

Wildcard:	Supported in:	Matches:
\	All	Escape next character. Toggles whether following punctuation is treated as a wildcard or not. Following letters or digits are interpreted in various different ways depending on the program.
.	All	Any character.
^	All	Beginning of line
\$	All	End of line
[...]	All	Any of the characters between the brackets
[^...]	All	Any of characters <i>except those</i> between the brackets.
*	All	Accept any number of repetitions of the previous element.
?	egrep/Emacs, Perl/Python	Accept zero or one instances of the previous element.
+	egrep/Emacs, Perl/Python	Accept one or more instances of the previous element.
{n}	egrep, Perl/Python; as \{n\} in Emacs	Accept exactly n repetitions of the previous element. Not supported by some older regexp engines.
{n,}	egrep, Perl/Python; as \{n\} in Emacs	Accept n or more repetitions of the previous element. Not supported by some older regexp engines.
{m,n}	egrep, Perl/Python; as \{n\} in Emacs	Accept at least m and at most n repetitions of the previous element. Not supported by some older regexp engines.
	egrep, Perl/Python; as \ in Emacs	Accept the element to the left or the element to the right. This is usually used with some form of pattern-grouping delimiters.
(...)	Perl/Python; as \(...\) in older versions.	Treat this pattern as a group (in newer regexp engines like Perl and Pythons). Older regexp engines such as those in Emacs and grep require \(...\).