Character	Description
g	global match
i	ignore case
gi	both global match and ignore cas

Methods that use Regular Expressions

Method	Type	Description
exec	RegExp	Executes a search for a match in a string. It returns an array of information.
test	RegExp	Tests for a match in a string. It returns true or false.
match	String	Executes a search for a match in a string. It returns an array of information or null on a mismatch.
search	String	Tests for a match in a string. It returns the index of the match, or -1 if the search fails.
replace	String	Executes a search for a match in a string, and replaces the matched substring with a replacement substring.
split	String	Uses a regular expression or a fixed string to break a string into an array of substrings.

Pattern Syntax

Character	Meaning
\	Indicates next character should *not* be interpreted literally if it normally is, and *should* be interpreted literally if it normally isn't.
^	Matches beginning of input or line.
$	Matches end of input or line.
*	Matches 0 or more instances of preceding character.
	Matches 1 or more instances of preceding character.
?	Matches 0 or 1 instances of preceding character.
.	Matches any single character other than the newline character.
(x)	Matches *x* and remembers the match.
x|y	Matches either *x* or *y*.
{n}	Matches exactly *n* instances of preceding character (where *n* is an integer).
{n,}	Matches at least *n* instances of preceding character (where *n* is an integer).
{n,m}	Matches it least *n* and at most *m* instances of preceding character (where *n* and *m* are integers).
[xyz]	Matches any one of enclosed characters (specify range using hyphen, such as [0-9].
[^xyz]	Matches any character not enclosed (specify range using hyphen, such as [^0-9].
[\b]	Matches a backspace.
\b	Matches a word boundary, such as a space.
\B	Matches a nonword boundary.
\cX	Matches a control character, *X*.
\d	Matches a digit character (same as [0-9]).
\D	Matches a nondigit character (same as [^0-9]).
\f	Matches a form feed.
\n	Matches a line feed.
\r	Matches a carriage return.
\s	Matches a single white space character, including space, tab, form feed, and line feed (same as [\f\n\r\t\v]).
\S	Matches a single non-white-space character (same as [^\f\n\r\t\v]).
\t	Matches a tab.
\v	Matches a vertical tab.
\w	Matches any alphanumeric character, including the underscore (same as [A-Za-z0-9_]).
\W	Matches any nonword character (same as [^A-Za-z0-9_]).
\n	A reference to the last substring matching the *n*th parenthetical (where *n* is a positive integer).
\ooctal \xhex	Matches an octal or hexadecimal escape value (for embedding ASCII codes).

Property	Description
$1, ..., $9	Parenthesized substring matches, if any.
constructor	Specifies the function that creates an object's prototype.
global	Whether or not to test the regular expression against all possible matches in a string, or only against the first.
ignoreCase	Whether or not to ignore case while attempting a match in a string.
$_input	The string against which a regular expression is matched.
lastIndex	The index at which to start the next match.
$&lastMatch	The last matched characters.
$lastParen	The last parenthesized substring match, if any.
$`leftContext	The substring preceding the most recent match.
$*multiline	Whether or not to search in strings across multiple lines.
prototype	Allows the addition of properties to all objects.
$'rightContext	The substring following the most recent match.
source	The text of the pattern.

Method Summary

Method	Description
compile	Compiles a regular expression object.
exec	Executes a search for a match in its string parameter.
test	Tests for a match in its string parameter.
toSource	Returns an object literal representing the specified object; you can use this value to create a new object. Overrides the Object.toSource method.
toString	Returns a string representing the specified object. Overrides the Object.toString method.
valueOf	Returns the primitive value of the specified object. Overrides the Object.valueOf method.

Creating the Regular Expression Object

Type	Usage	Example	Note
Literal way	/pattern/flags	var objRegex = /ab?	Do not use quotation marks to indicate strings.
Constructor method	new RegExp("pattern"[, "flags"])	var objRegex = new RegExp("ab", "i")	The normal escape rules apply (using the \ character).