**Table 4.3 Special characters in regular expressions.**

|  |  |
| --- | --- |
| **Character** | **Meaning** |
| \ | For characters that are usually treated literally, indicates that the next character is special and not to be interpreted literally.  For example, /b/ matches the character 'b'. By placing a backslash in front of b, that is by using /\b/, the character becomes special to mean match a word boundary.  -or-  For characters that are usually treated specially, indicates that the next character is not special and should be interpreted literally.  For example, \* is a special character that means 0 or more occurrences of the preceding character should be matched; for example, /a\*/ means match 0 or more a's. To match \* literally, precede the it with a backslash; for example, /a\\*/ matches 'a\*'. |
| ^ | Matches beginning of input or line.  For example, /^A/ does not match the 'A' in "an A," but does match it in "An A." |
| $ | Matches end of input or line.  For example, /t$/ does not match the 't' in "eater", but does match it in "eat" |
| \* | Matches the preceding character 0 or more times.  For example, /bo\*/ matches 'boooo' in "A ghost booooed" and 'b' in "A bird warbled", but nothing in "A goat grunted". |
| + | Matches the preceding character 1 or more times. Equivalent to {1,}.  For example, /a+/ matches the 'a' in "candy" and all the a's in "caaaaaaandy." |
| ? | Matches the preceding character 0 or 1 time.  For example, /e?le?/ matches the 'el' in "angel" and the 'le' in "angle." |
| . | (The decimal point) matches any single character except the newline character.  For example, /.n/ matches 'an' and 'on' in "nay, an apple is on the tree", but not 'nay'. |
| (x) | Matches 'x' and remembers the match.  For example, /(foo)/ matches and remembers 'foo' in "foo bar." The matched substring can be recalled from the resulting array's elements [1], ..., [n], or from the predefined RegExp object's properties $1*, ...,* $9. |
| x|y | Matches either 'x' or 'y'.  For example, /green|red/ matches 'green' in "green apple" and 'red' in "red apple." |
| {n} | Where n is a positive integer. Matches exactly n occurrences of the preceding character.  For example, /a{2}/ doesn't match the 'a' in "candy," but it matches all of the a's in "caandy," and the first two a's in "caaandy." |
| {n,} | Where n is a positive integer. Matches at least n occurrences of the preceding character.  For example, /a{2,} doesn't match the 'a' in "candy", but matches all of the a's in "caandy" and in "caaaaaaandy." |
| {n,m} | Where n and m are positive integers. Matches at least n and at most m occurrences of the preceding character.  For example, /a{1,3}/ matches nothing in "cndy", the 'a' in "candy," the first two a's in "caandy," and the first three a's in "caaaaaaandy" Notice that when matching "caaaaaaandy", the match is "aaa", even though the original string had more a's in it. |
| [xyz] | A character set. Matches any one of the enclosed characters. You can specify a range of characters by using a hyphen.  For example, [abcd] is the same as [a-c]. They match the 'b' in "brisket" and the 'c' in "ache". |
| [^xyz] | A negated or complemented character set. That is, it matches anything that is not enclosed in the brackets. You can specify a range of characters by using a hyphen.  For example, [^abc] is the same as [^a-c]. They initially match 'r' in "brisket" and 'h' in "chop." |
| [\b] | Matches a backspace. (Not to be confused with \b.) |
| \b | Matches a word boundary, such as a space. (Not to be confused with [\b].)  For example, /\bn\w/ matches the 'no' in "noonday";/\wy\b/ matches the 'ly' in "possibly yesterday." |
| \B | Matches a non-word boundary.  For example, /\w\Bn/ matches 'on' in "noonday", and /y\B\w/ matches 'ye' in "possibly yesterday." |
| \c*X* | Where *X* is a control character. Matches a control character in a string.  For example, /\cM/ matches control-M in a string. |
| \d | Matches a digit character. Equivalent to [0-9].  For example, /\d/ or /[0-9]/ matches '2' in "B2 is the suite number." |
| \D | Matches any non-digit character. Equivalent to [^0-9].  For example, /\D/ or /[^0-9]/ matches 'B' in "B2 is the suite number." |
| \f | Matches a form-feed. |
| \n | Matches a linefeed. |
| \r | Matches a carriage return. |
| \s | Matches a single white space character, including space, tab, form feed, line feed. Equivalent to [ \f\n\r\t\v].  for example, /\s\w\*/ matches ' bar' in "foo bar." |
| \S | Matches a single character other than white space. Equivalent to [^ \f\n\r\t\v].  For example, /\S/\w\* matches 'foo' in "foo bar." |
| \t | Matches a tab |
| \v | Matches a vertical tab. |
| \w | Matches any alphanumeric character including the underscore. Equivalent to [A-Za-z0-9\_].  For example, /\w/ matches 'a' in "apple," '5' in "$5.28," and '3' in "3D." |
| \W | Matches any non-word character. Equivalent to [^A-Za-z0-9\_].  For example, /\W/ or /[^$A-Za-z0-9\_]/ matches '%' in "50%." |