

Characters

The following expressions will match single characters. For more information see Microsoft's article on [Character Classes](#).

Characters

Ordinary characters Characters other than `. $ ^ { [(|) * + ? \` match themselves.

`.` Matches any character excluding the line feed. Includes the line feed in single-line mode.

`[abc]` A character class (may contain more than one character). Matches any character that is contained within the brackets, in no particular order.

`[^abc]` The opposite of `[]`. Matches all characters not contained within the brackets.

`[a-z]` Character range: Matches any single character in the range from first (a) to last (z).

`\w` Matches an alpha-numeric character (a-z, A-Z, 0-9, and underscore).

`\W` The opposite of `\w`. Matches any non-alphanumeric character.

`\d` Matches a decimal character (0-9).

`\D` The opposite of `\d`. Matches any non-decimal character.

`\s` Matches a character of whitespace (space, tab, carriage return, line feed).

`\S` The opposite of `\s`. Matches any non-whitespace character.

`\r` Matches a carriage return.

`\n` Matches a new line (line feed).

`\f` Matches a form feed.

Characters (cont)

`\t` Matches a tab.

`\v` Matches a vertical tab.

`\a` Matches a bell character.

`\b` In a character class, matches a backspace.

`\e` Matches an escape.

`\040` Uses octal representation to specify a character (octal consists of up to three digits).

`\x20` Uses hexadecimal representation to specify a character (hex consists of exactly two digits).

`\c0003` Matches the specified 4-digit ASCII control character.

`\u0020` Matches a Unicode character by using hexadecimal representation (exactly four digits).

`\p{name}` Matches any single character in the Unicode general category or named block specified by name.

`\P{name}` Matches any single character that is not in the Unicode general category or named block specified by name.

`\` In front of any of the special characters (`. $ ^ { [(|) * + ? \`), this will match the character itself.

Assertions

The following expressions specify the location to search for a match, but do not match anything themselves.



Assertions

<code>^</code>	The match must start at the beginning of the string (or beginning of the line in multiline mode).
<code>\$</code>	The match must occur at the end of the string or before <code>\n</code> at the end of the string (or end of the line in multiline mode).
<code>\A</code>	The match must occur at the start of the string.
<code>\Z</code>	The match must occur at the end of the string or before <code>\n</code> at the end of the string.
<code>\z</code>	The match must occur at the end of the string.
<code>\G</code>	The match must occur at the point where the previous match ended.
<code>\b</code>	Asserts a boundary between word and non-word characters.
<code>\B</code>	The opposite of <code>\b</code> . Asserts a location that is not a boundary between word and non-word characters.
<code>(?=pattern)</code>	Asserts that the specified pattern exists immediately after this location. Known as a positive lookahead.
<code>(?!pattern)</code>	Asserts that the specified pattern does not exist immediately after this location. Known as a negative lookahead.
<code>(?<=pattern)</code>	Asserts that the specified pattern exists immediately before this location. Known as a positive lookbehind.
<code>(?!pattern)</code>	Asserts that the specified pattern does not exist immediately before this location. Known as a negative lookbehind.

