

Backslash and Generic Character Types

The third use of backslash is for specifying generic character types. The following are always recognized:

Igor Pattern	PCRE Pattern	Character(s) Matched
<code>\\d</code>	<code>\d</code>	Any decimal digit
<code>\\D</code>	<code>\D</code>	Any character that is not a decimal digit
<code>\\s</code>	<code>\s</code>	Any whitespace character
<code>\\S</code>	<code>\S</code>	Any character that is not a whitespace character
<code>\\w</code>	<code>\w</code>	Any “word” character
<code>\\W</code>	<code>\W</code>	Any “nonword” character

Each pair of escape sequences, such as `\d` and `\D`, partitions the complete set of characters into two disjoint sets. Any given character matches one, and only one, of each pair.

These character type sequences can appear both inside and outside character classes. They each match one character of the appropriate type. If the current matching point is at the end of the subject string, all of them fail, since there is no character to match.

The `\s` whitespace characters are HT (9), LF (10), VT (11), FF (12), CR (13), and space (32). In Igor Pro 6 `\s` did not match the VT character (code 11).

A “word” character is an underscore or any character that is a letter or digit.

Backslash and Simple Assertions

The fourth use of backslash is for certain simple assertions. An assertion specifies a condition that has to be met at a particular point in a match without consuming any characters from the subject string. The use of subpatterns for more complicated assertions is described in **Assertions** on page IV-190. The backslashed assertions are:

Igor Pattern	PCRE Pattern	Character(s) Matched
<code>\\b</code>	<code>\b</code>	At a word boundary
<code>\\B</code>	<code>\B</code>	Not at a word boundary
<code>\\A</code>	<code>\A</code>	At start of subject
<code>\\Z</code>	<code>\Z</code>	At end of subject or before newline at end
<code>\\z</code>	<code>\z</code>	At end of subject
<code>\\G</code>	<code>\G</code>	At first matching position in subject

These assertions may not appear in character classes (but note that `\b` has a different meaning, namely the backspace character, inside a character class).

A word boundary is a position in the subject string where the current character and the previous character do not both match `\w` or `\W` (i.e. one matches `\w` and the other matches `\W`), or the start or end of the string if the first or last character matches `\w`, respectively.

While PCRE defines additional simple assertions (`\A`, `\Z`, `\z` and `\G`), they are not any more useful to Igor’s regular expression commands than the `^` and `$` characters.

Circumflex and Dollar

Outside a character class, in the default matching mode, the circumflex character `^` is an assertion that is true only if the current matching point is at the start of the subject string. Inside a character class, circumflex has an entirely different meaning (see **Character Classes and Brackets** on page IV-182).