List of Regular Expressions

Character	**Result/Use**
Any character	Represents the given character unless otherwise specified.
.	Represents any single character except for a line break or paragraph break. For example, the search term "sh.rt" returns both "shirt" and "short".
^	Only finds the search term if the term is at the beginning of a paragraph. Special objects such as empty fields or character-anchored frames, at the beginning of a paragraph are ignored. Example: "^Peter".
$	Only finds the search term if the term appears at the end of a paragraph. Special objects such as empty fields or character-anchored frames at the end of a paragraph are ignored. Example: "Peter$".
*	Finds zero or more of the characters in front of the "*". For example, "Ab*c" finds "Ac", "Abc", "Abbc", "Abbbc", and so on.
+	Finds one or more of the characters in front of the "+". For example, "AX.+4" finds "AXx4", but not "AX4". The longest possible string that matches this search pattern in a paragraph is always found. If the paragraph contains the string "AX 4 AX4", the entire passage is highlighted.
?	Finds zero or one of the characters in front of the "?". For example, "Texts?" finds "Text" and "Texts" and "x(ab|c)?y" finds "xy", "xaby", or "xcy".
\	Search interprets the special character that follows the "\" as a normal character and not as a regular expression (except for the combinations \n, \t, \>, and \<). For example, "tree\." finds "tree.", not "treed" or "trees".
\n	Represents a line break that was inserted with the Shift+Enter key combination. To change a line break into a paragraph break, enter **\n** in the **Search for** and **Replace with** boxes, and then perform a search and replace. \n in the **Search for** text box stands for a line break that was inserted with the Shift+Enter key combination. \n in the **Replace with** text box stands for a paragraph break that can be entered with the Enter or Return key.
\t	Represents a tab. You can also use this expression in the **Replace with** box.
\>	Only finds the search term if it appears at the end of a word. For example, "book\>" finds "checkbook", but not "bookmark".
\<	Only finds the search term if it appears at the beginning of a word. For example, "\<book" finds "bookmark", but not "checkbook".
^$	Finds an empty paragraph.
^.	Finds the first character of a paragraph.
& or $0	Adds the string that was found by the search criteria in the **Search for** box to the term in the **Replace with** box when you make a replacement. For example, if you enter "window" in the **Search for** box and "&frame" in the **Replace with** box, the word "window" is replaced with "windowframe". You can also enter an "&" in the **Replace with** box to modify the **Attributes** or the **Format** of the string found by the search criteria.
[abc123]	Represents one of the characters that are between the brackets.
[a-e]	Represents any of the characters that are between a and e, including both start and end characters The characters are ordered by their code numbers.
[a-eh-x]	Represents any of the characters that are between a-e and h-x.
[^a-s]	Represents everything that is not between a and s.
\xXXXX	Represents a special character based on its four-digit hexadecimal code (XXXX). The code for the special character depends on the font used. You can view the codes by choosing **Insert - Special Character**.

|	Finds the terms that occur before the "|" and also finds the terms that occur after the "|". For example, "this|that" finds "this" and "that".
{2}	Defines the number of times that the character in front of the opening bracket occurs. For example, "tre{2}" finds and selects "tree".
{1,2}	Defines the minimum and maximum number of times that the character in front of the opening bracket can occur. For example, "tre{1,2}" finds and selects "tre" and "tree".
{1,}	Defines the minimum number of times that the character in front of the opening bracket can occur. For example, "tre{2,}" finds "tree", "treee", and "treeeee".
( )	In the **Search for** box: Defines the characters inside the parentheses as a reference. You can then refer to the first reference in the current expression with "\1", to the second reference with "\2", and so on. For example, if your text contains the number 13487889 and you search using the regular expression (8)7\1\1, "8788" is found. You can also use () to group terms, for example, "a(bc)?d" finds "ad" or "abcd". In the **Replace with** box: Use $ (dollar) instead of \ (backslash) to replace references. Use $0 to replace the whole found string.
[:alpha:]	Represents an alphabetic character. Use [:alpha:]+ to find one of them.
[:digit:]	Represents a decimal digit. Use [:digit:]+ to find one of them.
[:alnum:]	Represents an alphanumeric character ([:alpha:] and [:digit:]).
[:space:]	Represents a space character (but not other whitespace characters).
[:print:]	Represents a printable character.
[:cntrl:]	Represents a nonprinting character.
[:lower:]	Represents a lowercase character if **Match case** is selected in **Options**.
[:upper:]	Represents an uppercase character if **Match case** is selected in **Options.**

Examples

e([:digit:])? -- finds 'e' followed by zero or one digit. Note that currently all named character classes like [:digit:] must be enclosed in parentheses.

^([:digit:])$ -- finds lines or cells with exactly one digit.

You can combine the search terms to form complex searches.

To find three-digit numbers alone in a paragraph

```
^[:digit:]{3}$
```

^ means the match has to be at the start of a paragraph,

[:digit:] matches any decimal digit,

{3} means there must be exactly 3 copies of "digit",

$ means the match must end a paragraph.

Related Topics

Find & Replace

Wiki page about regular expressions in Writer

Wiki page about regular expressions in Calc