the-Automator.com/IntroToRegex Save 25%!

https://regex101.com

Basic RegEx Syntax

Symbol/Pattern	Description	Example
.	Matches any single character except newline	a.b matches acb, aab
^	Matches the start of a string	^abc matches abcdef
$	Matches the end of a string	abc$ matches 123abc
*	Matches zero or more of the preceding element	a* matches a, aaa, aabc
+	Matches one or more of the preceding element	a+ matches a, aa, aaa
?	Matches zero or one of the preceding element	a? matches ``, a
{n}	Matches exactly n occurrences of the preceding element	a{3} matches aaa
{n,}	Matches n or more occurrences of the preceding element	a{3,} matches aaa, aaaa
{n,m}	Matches between n and m occurrences of the preceding element	a{2,4} matches aa, aaa, aaaa
[]	Matches any one of the characters inside the brackets	[abc] matches a, b, c
[^]	Matches any character not inside the brackets	[^abc] matches d, e, f
|	Alternation, matches either the expression before or after the |	a | b matches a, b
()	Groups expressions and captures the matched text	(abc) captures abc
\d	Matches any digit, equivalent to [0-9]	\d matches 1, 2, 3
\D	Matches any non-digit	\D matches a, b, c
\w	Matches any word character (alphanumeric + underscore)	\w matches a, 1, _
\W	Matches any non-word character	\W matches !, @, #
\s	Matches any whitespace character	\s matches space, tab, newline
\S	Matches any non-whitespace character	\S matches a, b, 1
\b	Matches a word boundary	\bword\b matches word in hello word!
\B	Matches a non-word boundary	\Bword\B matches password
[[:alpha:]]	Matches any alphabetic character	[[:alpha:]] matches a, B
[[:digit:]]	Matches any digit, equivalent to \d	[[:digit:]] matches 0, 1, 9
[[:alnum:]]	Matches any alphanumeric character	[[:alnum:]] matches a, 1, A
[[:space:]]	Matches any whitespace character	[[:space:]] matches space, tab, newline
[[:punct:]]	Matches any punctuation character	[[:punct:]] matches !, @, #
[[:lower:]]	Matches any lowercase letter	[[:lower:]] matches a, b, z
[[:upper:]]	Matches any uppercase letter	[[:upper:]] matches A, B, Z
(?P<name>...)	Defines a named group	(?P<word>\w+) captures word as a named group
(?P=name)	Matches the text matched by the named group	(?P<word>\w+)\s(?P=word) matches word word

Advanced Syntax

Syntax/Pattern	Description	Example
(?=...)	Positive lookahead: Asserts that what follows the position is ...	\d(?=abc) matches 1 in 1abc
(?!...)	Negative lookahead: Asserts that what does not follow the position is ...	\d(?!abc) matches 1 in 1def
(?<=...)	Positive lookbehind: Asserts that what precedes the position is ...	(?<=abc)\d matches 1 in abc1
(?<!...)	Negative lookbehind: Asserts that what does not precede the position is ...	(?<!abc)\d matches 1 in def1
(?(id/name)yes|no)	Conditional: Matches yes if the group with id or name exists, otherwise matches no	(a)?b(?(1)c|d) matches abc or bd
(?P<name>...)	Named group: Defines a named capturing group	(?P<word>\w+) matches word as a named group
(?P=name)	Backreference by name: Matches the text matched by the named group	(?P<word>\w+)\s(?P=word) matches word word
\1, \2, ...	Backreference: Matches the text captured by the nth group	(\w+)\s\1 matches word word
(?:...)	Non-capturing group: Groups expressions without capturing	(?:abc) matches abc but does not capture
(?R)	Recursion: Matches the entire pattern recursively	((a|b)\2?|c(?R)) matches nested patterns

Options

Flag	Description	Example
g	Global search (find all matches)	/abc/g matches all occurrences of abc in the string
i	Case-insensitive matching	/abc/i matches ABC, abc, AbC
m	Multiline mode	/^abc/m matches abc at the start of any line
s	Dotall mode (dot matches newline)	/a.b/s matches a\nb
X	Enables more complex regex syntax (additional features)	Ignores whitespace

Autohotkey Gotchas

Regex Flavor	PCRE2	
RegexMatch	Function that allows you to match a regular expression in a string and store it in an object	RegexMatch("this is a test", "t.*?s", &matched)
RegexReplace	Function that allows you to match a regular expression in a string and replace it with something else	RegexReplace("testing and talking to you", ".*ing", "trying")
~=	Shorthand for RegexMatch but doesn't capture the matched pattern	if var ~= "i)test"
Additional Resources	Quick Reference	Beginners Guide to Regular Expressions v1 AHK

Special AHK options

S	Studies the pattern to try improve its performance. This is useful when a particular pattern (especially a complex one) will be executed many times.
`a	Enables recognition of additional newline markers.
`n	Causes a solitary linefeed (`n) to be the only recognized newline marker (see above).
`r	Causes a solitary carriage return (`r) to be the only recognized newline marker (see above).