

RegEx Notes

Examples

RegEx	Matches string		
hello	contains {hello}		
`gray\`	grey\`		matches {gray, grey}
`gr(a\`	e)y\`		contains {gray, grey}
gr[ae]y	contains {gray, grey}		
b[aeiou]bble	contains {babble, bebble, bibble, bobble, bubble}		
`[b-chm-pP]at\`	ot\`	contains {bat, cat, hat, mat, nat, oat, pat, Pat, ot}	
colou?r	contains {color, colour}		
`rege(x(es)?\`	xps?)\`	contains {regex, regexes, regexp, regexps}	
go*gle	contains {ggle, gogle, google, gooogle, goooogle}		
go+gle	contains {gogle, google, gooogle, goooogle....}		
g(oog)+le	contains {google, googoogle, googoogooe....}		

RegEx	Matches string		
<code>z{3}</code>	contains <code>{zzz}</code>		
<code>z{3,6}</code>	contains <code>{zzz, zzzz, zzzzz, zzzzzz}</code>		
<code>z{3,}</code>	contains <code>{zzz, zzzz, zzzzz, ...}</code>		
<code>[Bb]rainf**k</code>	contains <code>{Brainf**k, brainf**k}</code>		
<code>\\d</code>	contains <code>{0,1,2,3,4,5,6,7,8,9}</code>		
<code>1\\d{10}</code>	contains an 11-digit string starting with a 1		
<code>`[2-9]\\</code>	<code>[12]\\d\\</code>	<code>3[0-6]`</code>	contains an integer in the range 2..36 inclusive
<code>Hello\\nworld</code>	contains Hello followed by a newline followed by world		
<code>\\d+(\\.\\d\\d)?</code>	contains a positive integer or a floating point number with exactly two characters after the decimal point.		
<code>[^i*&2@]</code>	contains any character other than an i, asterisk, ampersand, 2, or @		
<code>// [^\\r\\n]*[\\r\\n]</code>	contains a java or C# // comment		
<code>^dog</code>	begins with "dog"		
<code>dog\$</code>	ends with "dog"		
<code>^dog\$</code>	is exactly "dog"		

Components of Regexes

Component	Matches string
-----------	----------------

Component	Matches string
[abc]	a or b or c
[^abc]	any character except a, b or c
[a-zA-Z]	a-z or A-Z
\s	space
\S	anything that's not a space
\w	matches any word character (alphanumeric & underscore).
\W	not word
.	matches any character except line breaks
?	makes the preceding quantifier lazy, causing it to match as few characters as possible.
\A	beginning of a string
\z	end of a string
\Z	depends on the engine
\b	word boundary
\B	not a word boundary

Quantifiers

Text	RegEx
Zero or one	?
Zero or more	*? or *+
One or more	+ or +?
m times	{m}
At least m times	{m, }
At least m, at most n times	{m, n}