

Regular Expression Cheat Sheet

Syntax (Anchors)

<code>^</code>	Matches the beginning of input	<code>^a</code> : “alice” not “banana” <code>^</code> : “a line\nand break” => 2 match
<code>\$</code>	Matches the end of input.	<code>a\$</code> : “obama” not “name”
<code>.</code>	Matches any single character except a newline	<code>a.b</code> : “a0b”, “acb” not “ab”
<code>\b</code>	Word boundary	<code>\bdog\b</code> : “dog”, “dog ” not “doggy”
<code>\B</code>	Not word boundary	<code>\Bdog\B</code> : “doggy”, “adogg” not “dog”

<code>\s</code>	Whitespace (space, tab, newline,...)	<code>\sc</code> : matches “a cat” not “xcat”
<code>\S</code>	Not white space	<code>\Sc</code> : matches “cat” not “a cat”
<code>\d</code>	Digit	<code>\d</code> : matches “1” in “1abc”
<code>\D</code>	Not digit	<code>\D</code> : matches “b” in “123b”
<code>\w</code>	Word (alphanumeric character including <code>_</code>)	<code>\w</code> : matches “a” in “a%”
<code>\W</code>	Not word	<code>\W</code> : matches “%” in “a%”

Syntax (Quantifiers)

<code>*</code>	Matches the preceding character 0 or more times	<code>zo*</code> : “z” and “zoo”
<code>+</code>	Matches the preceding character 1 or more times	<code>zo+</code> : “zoo” not “z”
<code>?</code>	Matches the preceding character 0 or 1 time	<code>ca?r</code> : only “car” and “cr”
<code>{n}</code>	Matches exactly <i>n</i> times.	<code>o{2}</code> : “foooood” not “bob”
<code>{n,}</code>	Matches at least <i>n</i> times	<code>o{2,}</code> : “foooood” not “bob”
<code>{n,m}</code>	Matches the preceding character <i>n</i> to <i>m</i> times	<code>o{2,3}</code> : “food”, “foood” not “fod”

Syntax (Group & Range)

	or	(z w)o : “zo” and “wo”
(...)	Matches subexpression and remembers the match	(ab): “abc” -> 1st group match “ab”
(?...)	Matches subexpression and do not remembers the match (the match doesn’t in array)	(ab): “abc” -> No group
(?=...)	Matches only if there is a following pattern	win(?=7): “win” in “win7” not “win95”
(?!...)	Matches only if there is not a following pattern	win(?!=7): “win8” not “win7”
[...]	Range	c[ma-c]: “cm”, “cb” not “cd” c[A-E0-5]: “cB”, “c1” not c6
[^...]	Not in range	c[^ab]: “cc” not “ca”

Syntax (Modifier)

g	global match (find all)	/is/g : “this is” => two match
i	Case-i-nse-nsitive	/my/i : “My” and “my”
m	Multiple line	/is/m : “this is”

Replacement

\$&	insert the whole regex match	"1a2b".replace(/\d+/g, "(\$&") => "(1)a(2)b"
\$1,... \$9	Insert the text matched by one of the first 9 capturing groups.	"abc".replace(/(a)(b)(c)/g, "\$3\$2\$1") => "cba"
\$`	Insert the part of the subject string to the <i>left</i> of the regex match	"abc".replace(/b/, "\$`") => "aac"
\$'	Insert the part of the subject string to the <i>right</i> of the regex match	"abc".replace(/b/, "\$'") => "acc"