



PHP PCRE Cheat Sheet

Functions

<code>preg_match(pattern, subject[, submatches])</code>
<code>preg_match_all(pattern, subject[, submatches])</code>
<code>preg_replace(pattern, replacement, subject)</code>
<code>preg_replace_callback(pattern, callback, subject)</code>
<code>preg_grep(pattern, array)</code>
<code>preg_split(pattern, subject)</code>

Meta Characters

<code>^</code>	Start of subject (or line in multiline mode)
<code>\$</code>	End of subject (or line in multiline mode)
<code>[</code>	Start character class definition
<code>]</code>	End character class definition
<code> </code>	Alternates, eg <code>(a b)</code> matches a or b
<code>(</code>	Start subpattern
<code>)</code>	End subpattern
<code>\</code>	Escape character

Pattern Modifiers

<code>i</code>	Caseless – ignore case
<code>m</code>	Multiline mode - <code>^</code> and <code>\$</code> match start and end of lines
<code>s</code>	Dotall - <code>.</code> class includes newline
<code>x</code>	Extended– comments & whitespace
<code>e</code>	<code>preg_replace</code> only – enables evaluation of replacement as PHP code
<code>S</code>	Extra analysis of pattern
<code>U</code>	Pattern is ungreedy
<code>u</code>	Pattern is treated as UTF-8

Base Character Classes

<code>\w</code>	Any “word” character (a-z 0-9 _)
<code>\W</code>	Any non “word” character
<code>\s</code>	Whitespace (space, tab CRLF)
<code>\S</code>	Any non whitespace character
<code>\d</code>	Digits (0-9)
<code>\D</code>	Any non digit character
<code>.</code>	(Period) – Any character except newline

Quantifiers

<code>n*</code>	Zero or more of n
<code>n+</code>	One or more of n
<code>n?</code>	Zero or one occurrences of n
<code>{n}</code>	n occurrences exactly
<code>{n,}</code>	At least n occurrences
<code>{,m}</code>	At most m occurrences
<code>{n,m}</code>	Between n and m occurrences (inclusive)

Point based assertions

<code>\b</code>	Word boundary
<code>\B</code>	Not a word boundary
<code>\A</code>	Start of subject
<code>\Z</code>	End of subject or newline at end
<code>\z</code>	End of subject
<code>\G</code>	First matching position in subject

Subpattern Modifiers & Assertions

<code>(?:)</code>	Non capturing subpattern	<code>((?:foo fu)bar)</code> matches foobar or fubar without foo or fu appearing as a captured subpattern
<code>(?=)</code>	Positive look ahead assertion	<code>foo(=bar)</code> matches foo when followed by bar
<code>(?!)</code>	Negative look ahead assertion	<code>foo(?!bar)</code> matches foo when <i>not</i> followed by bar
<code>(?<=)</code>	Positive look behind assertion	<code>(?<=foo)bar</code> matches bar when preceded by foo
<code>(?<!)</code>	Negative look behind assertion	<code>(?<!foo)bar</code> matches bar when <i>not</i> preceded by foo
<code>(?>)</code>	Once-only subpatterns	<code>(?>\d+)bar</code> Performance enhancing when bar not present
<code>(?(x))</code>	Conditional subpatterns	<code>(?(3)foo fu)bar</code> Matches foo if 3 rd subpattern has matched, fu if not
<code>(?#)</code>	Comment	<code>(?# Pattern does x y or z)</code>