

■PHP正規表現 (Perl互換) : PCRE

作成: 2010.10.25/2015.7.27修正 大角

●PCREの構文

[関数]

```

preg_match (“/[正規表現パターン]/[修飾子]”, [入力文字列], [一致した内容 (配列)], [flags], [offset]);
preg_match_all (“/[正規表現パターン]/[修飾子]”, [入力文字列], [一致した全ての内容 (配列)], [flags], [offset]);
preg_replace (“/[正規表現パターン]/[修飾子]”, [置換する文字列又は配列], [検索・置換対象となる文字列又は配列], [limit], [count]);
preg_filter (“/[正規表現パターン]/[修飾子]”, [置換する文字列又は配列], [検索・置換対象となる文字列又は配列], [limit], [count]);
preg_replace_callback (“/[正規表現パターン]/[修飾子]”, [コールバック関数], [検索・置換対象となる文字列又は配列], [limit], [count]);
preg_split (“/[正規表現パターン]/[修飾子]”, [入力文字列], [limit], [flags]);
preg_grep (“/[正規表現パターン]/[修飾子]”, [入力の配列], [flags]);

```

他

正規表現パターンのデリミタ	PCRE 関数を使うときには、パターンを delimiters で囲まなければなりません。英数字、バックスラッシュ、空白文字以外の任意の文字をデリミタとして使うことができます。よく使われる文字は、スラッシュ (/)、ハッシュ記号 (#) およびチルダ (~) です。
---------------	---

●正規表現パターン

[メタ文字]

¥	次のメタ文字をエスケープ	
^	行の先頭にマッチ	
.	任意の文字にマッチ(但し改行は除く)	
\$	行の終端にマッチ(若しくは終端の改行の前)	
	代替	
()	グループ化	
[]	文字クラス	[A-Z] … 大文字アルファベット全てを表わす
[^]	指定文字クラス以外(最初の文字に用いた場合のみ)	[^a-z] … 小文字のアルファベット以外全てを表わす

[量指定子]

*	0 回以上のマッチ	
+	1 回以上のマッチ	
?	1 回若しくは 0 回のマッチ	
{n}	ちょうど n 回のマッチ	
{n, }	n 回以上のマッチ	
{n, m}	n 回以上 m 回以下のマッチ	

[最短一致]

- 問題がないのであれば、マッチしないほうを選ぶ。
- デフォルトでは、パターンで行われる量指定は“貪欲”です、つまりそれはパターンの残りの部分が可能な範囲で、(始めた地点から)可能な限り多くを先にあるパターンでマッチさせます。
- もし最小回数でのマッチを行いたいのであれば、量指定子の後ろに “?” を続けます。意味は変更されずに“貪欲さ”だけを変更できます。

*?	0 回以上の貪欲でないマッチ	
+	1 回以上の貪欲でないマッチ	
??	0 回若しくは 1 回の貪欲でないマッチ	
{n}?	ちょうど n 回の貪欲でないマッチ	
{n, }?	n 回以上の貪欲でないマッチ	
{n, m}?	n 回以上 m 回以下の貪欲でないマッチ	

[例]: データが「1234」の時、「^123?4\$」と「^123??4\$」の違い
「123??」→「12」にマッチ、「123?」→「123」にマッチ

[強欲]

- デフォルトでは、パターンのうちの量指定された一部によってパターン全体がマッチに失敗したとき、Perl はバックトラックを行います。しかしこの振る舞いは望まれないこともあります。その為 Perl は“強欲な(possessive)”量指定形式も提供しています。

**	0 回以上のマッチでなおかつ全く戻らない	
++	1 回以上のマッチでなおかつ全く戻らない	
?+	0 回若しくは 1 回のマッチでなおかつ全く戻らない	
{n}+	ちょうど n 回のマッチでなおかつ全く戻らない (冗長)	
{n, }+	n 回以上のマッチでなおかつ全く戻らない	
{n, m}+	n 回以上 m 回以下のマッチでなおかつ全く戻らない	

[例]: /“(?”[“¥¥”+]¥¥)*+”/ /“(?”(?”[“¥¥”+]¥¥)*”/

最後のクオートがマッチしなかったときバックトラックは役に立ちません。
(バックトラックするべきでない場所のヒントを php に与えるのに非常に便利)

[エスケープシーケンス]

¥t	タブ	(水平タブ: HT, TAB)
¥n	改行	(LF, NL)
¥r	復帰	(CR)
¥f	フォームフィード	(FF)
¥a	アラーム (ベル)	(BEL)
¥e	エスケープ (troff 的)	(troff 的) (ESC)
¥033	8進文字	
¥x1B	16進文字	
¥x[263a]	ロング16進文字	
¥cK	制御文字	
¥N{name}	名前付きユニコード文字	
¥l	次の文字を小文字に (vi 的)	
¥u	次の文字を大文字に (vi 的)	
¥L	¥E まで小文字に (vi 的)	
¥U	¥E まで大文字に (vi 的)	
¥E	変更の終端 (vi 的)	

¥Q	¥E までパターンメタ文字の無効化(Quote)	
[ブラケット内のみ]		
¥b	バックスペース	(0x08)
【文字クラス及び他の特殊なエスケープ】		
¥w	単語 文字にマッチ (英数字及び“_”)	
¥W	非“単語”文字にマッチ	
¥s	空白文字にマッチ	
¥S	非空白文字にマッチ	
¥d	数字にマッチ	
¥D	非数字にマッチ	
¥pP	名前属性 P にマッチ. 長い名前であれば ¥p[Prop]	
¥PP	P以外にマッチ	
¥X	拡張 Unicode “複合文字シーケンス (combining character sequence)” のマッチ, (?¥PM¥pM*) と等価	
¥C	1つの C 文字 (8進数) にマッチ, Unicode 環境でも同じ. 補足: 文字を UTF-8 バイト列へと変換するので, 壊れた UTF-8 片となるかもしれません. lookbehind はサポートしていません	
¥1	指定したグループへの後方参照. '1' には正の整数を指定できません	
¥g1	指定した若しくは前のグループへの後方参照	
¥g[-1]	数値は前のバッファを示す負の値にもできます, また 任意で安全にパースするために波括弧で括弧することもできます.	
¥g[name]	名前指定の後方参照	
¥k<name>	名前指定の後方参照	
¥K	¥K の左にある物を保持, \$& に含めない	
¥v	垂直空白	
¥V	垂直空白以外	
¥h	水平空白	
¥H	水平空白	
¥R	行区切り	

【アサーション(言明)】

・以下のゼロ幅のアサーションを定義しています:(Perl)

¥b	単語境界にマッチ	
¥B	単語境界以外にマッチ	
¥A	文字列の開始にのみマッチ	
¥Z	文字列の終端若しくは終端の改行前にのみマッチ	
¥z	文字列の終端にのみマッチ	
¥G	前回のマッチの 末尾にマッチ (前回の正規表現実行のマッチ終端位置)	

例:「1234567」のデータの 場合の「/¥G¥d{2}/」の時(all:g、^1234567→12^34567→1234^567→123456^7 のように「^」の位置が変化

【グループ化】

(?pattern)	部分正規表現のグルーピング(キャプチャなし)	
(?=pattern)	patternがこの位置の右側に存在する場合にマッチ(肯定の先読み)...位置にマッチ 例:「Windows (?=98 Me XP)」→「Windows 2000」はマッチしない	
(?!pattern)	patternがこの位置の右に存在しない場合にマッチ(否定先読み)	
(?<pattern)	patternがこの位置の左に存在する場合にマッチ(肯定戻り読み)	
(?<!pattern)	patternがこの位置の左に存在しない場合にマッチ(否定戻り読み)	
(?>pattern)	マッチ文字列に対するバックトラックを禁止する	

●パターン修飾子

i	この修飾子を設定すると、パターンの中の文字は 大文字にも小文字にもマッチします。
m	デフォルトで、PCRE は、検索対象文字列を(実際には複数行からなる 場合でも)単一の行からなるとして処理します。「行頭」メタ文字 (^) は、対象文字列の最初にしかマッチしません。一方、「行末」メタ文字 (\$) は、文字列の最後、または (D 修飾子が設定されていない場合) 最後にある改行記号の前だけにしかマッチしません。この動作は Perl と同じです。この修飾子を設定すると、「行頭」および「行末」メタ文字は 対象文字列において、文字列の最初と最後に加えて、各改行の直前と直後にそれぞれマッチします。この動作は、Perl の /m 修飾子と同じです。対象文字列の中に “¥n” 文字がない場合や、またはパターンに “または \$” がいない場合は、この修飾子を設定しても意味はありません。
s	この修飾子を設定すると、パターン中のドットメタ文字は 改行を含む全ての文字にマッチします。これを設定しない場合は、改行にはマッチしません。この修飾子は、Perl の /s 修飾子と同じです。[a] のような否定の文字クラスは、この修飾子の設定によらず、常に改行文字にマッチします。
x	この修飾子を設定すると、エスケープするか 文字クラスの内部を除き、パターンの空白文字は完全に無視されます。文字クラスの外にあって、かつエスケープされていない # と次の改行文字の間の文字も無視されます。この動作は、Perl の /x 修飾子と同じであり、複雑なパターンの内部に コメントを記述することが可能となります。しかし、この修飾子は、データ文字にのみ適用されることに注意してください。空白文字をパターンの特殊文字の並びの中、例えば条件付きサブパターン (?< の内部に置くことはできません。
e	この修飾子を設定すると、preg_replace() は、置換文字列において後方参照に関する通常の置換を行った後、PHP コードとして評価し、検索文字列を置換するためにその結果を使用します。置換された後方参照においては、単引用符や二重引用符、バックスラッシュ (¥)および NULL 文字は バックスラッシュでエスケープされます。 (注):PHP 5.5.0 で 非推奨 となり、PHP 7.0.0 で 削除