

[\[目次\]](#)

## 利用可能な正規表現

使える(かも知れない)正規表現についての解説です。すべての動作を確認することは難しく、またライブラリの更新により動作が変更になることもあります。最終的にはご自分でご確認ください。

関連: [正規表現ライブラリ](#)

### 基本要素

¥	退避修飾(エスケープ) 正規表現記号の有効/無効の制御 ¥の次にある正規表現記号を普通の文字として扱います。またアルファベット文字と組み合わせて特殊な意味を持たせたりします。
	選択子 パターンの論理和
(...)	式集合(グループ) パターンをグループ化。
[...]	文字集合(文字クラス) キャラクタクラス。

### 文字集合(キャラクタクラス)

[...] の中に以下のものが指定可能です。

...	[ABC] はAかBかCのどれかにマッチします。
^...	否定 [^ABC] はAとBとC以外の任意の1文字にマッチします。
x-y	範囲 [A-Z] は、「A」から「Z」までの文字のどれか1つとマッチします。
[...]	(鬼) 文字集合内文字集合
..&&..	(鬼) 積演算
[:xxxx:]	(鬼) <a href="#">POSIXブラケット</a>
[:^xxxx:]	(鬼) <a href="#">POSIXブラケット</a> (否定)

(鬼) はbregonig.dllのみ

### 量指定子(量子)

最小一致 (無欲)	最大一致 (欲張り)	
*?	*	直前のパターンの0回以上の繰り返し
+?	+	直前のパターンの1回以上の繰り返し
??	?	直前のパターンが0回または1回現われる
{n}?	{n}	直前のパターンの n 回の繰り返し
{n,}?	{n,}	直前のパターンの n 回以上の繰り返し
{n,m}?	{n,m}	直前のパターンが n 回以上、m 回以下

正規表現Aを[A-Z]\*PROC、正規表現Bを[A-Z]\*?PROC とします。

SAKURA\_COLLBACKPROC\_BREXP\_PROC

という文字列中で、最初にマッチするのは以下ようになります。

Aの場合: SAKURA\_CALLBACKPROC\_BREXP\_PROC

Bの場合: SAKURA\_CALLBACKPROC

## 文字

¥t	水平タブコード(HT,TAB)
¥n	ラインフィード(LF)
¥r	キャリッジリターン(CR)
¥b	後退空白／バックスペース(BS) []の中でのみ有効
¥f	フォームフィード／改ページ(FF)
¥a	鐘／アラーム(BEL)
¥e	退避修飾／エスケープコード(ESC)
¥ooo	oに8進数で文字コードを指定する(oは1～3桁)
¥xHH	Hに16進数で文字コードを指定する(Hは1～2桁)
¥x[HHHH]	(鬼) 拡張16進数表現(Hは1～4桁)
¥c[	コントロール文字([はコントロール文字)
¥Q	(鬼) ¥Eに至るまで 正規表現演算子(正規表現記号)を抑制します
¥E	(鬼) 正規表現演算子(正規表現記号)の抑制状態を終端します

(鬼) はbregonig.dllのみ

## 文字種

.	¥nを除く任意の1文字 [^¥n]と同じ (Unicode版) 改行を除く任意の1文字。[^¥r¥n]と同じ *1
¥w	単語の構成文字 [0-9A-Za-z]と同じ (鬼) 2バイト文字も含む
¥W	単語の構成文字以外
¥s	空白類文字 [ ¥t¥v¥x0a¥x0d]と同じ (鬼) [:space:]と同じ
¥S	空白類文字以外
¥d	10進数字 (Unicode版) 2バイト文字=全角数字も含む
¥D	10進数字以外
¥p{property-name}	(鬼) <u>キャラクタプロパティ</u>
¥p{^property-name}	(鬼) <u>キャラクタプロパティ</u> (否定)
¥P{property-name}	

(鬼) はbregonig.dllのみ

(Unicode版) はUnicode版のサクラエディタ

## 位置(錨)

^	行頭
\$	行末
¥b	単語の境界 []の中ではバックスペースの意味になります
¥B	単語の境界以外
¥A	文字列の最初
¥Z	文字列の最後(最後が改行ならば改行の直前)
¥z	(鬼) 文字列の最後
¥G	照合開始位置

(鬼) はbregonig.dllのみ

## 後方参照と部分式呼び出し

¥n	番号指定による後方参照 ( )でグループ化した文字列を、¥n(nは1以上の整数)で参照します。
¥k<name> ¥k' name'	(鬼) 名前指定による後方参照
¥k<name+n> ¥k<name-n> ¥k' name+n' ¥k' name-n'	(鬼) ネストレベル付き後方参照
¥g<name> ¥g' name'	(鬼) 名前指定部分式呼出し
¥g<n> ¥g' n'	(鬼) 番号指定部分式呼出し

(鬼) はbregonig.dllのみ

## 拡張式集合

(?#...)	注釈
(式)	捕獲式集合
(?:式)	非捕獲式集合 (グループ化のみ)
(?<name>式) (?' name' 式)	(鬼) 名前付き捕獲式集合
(?=式)	先読み
(?!式)	否定先読み
(?<=式)	(鬼) 戻り読み
(?<!式)	(鬼) 否定戻り読み
(?>式)	(鬼) 原子的式集合
(?imsx)	孤立オプション i: 大文字小文字照合 m: 複数行 (サクラエディタではデフォルトでon) s: 単一行 (. が ¥n にもマッチ) x: 拡張形式 (空白を無視、# 以降を無視)
(?imsx-imsx)	(鬼) 孤立オプション (bregonig.dllではオプションの否定が可能)
(imsx-imsx:式)	(鬼) 式オプション

(鬼) はbregonig.dllのみ

## 置換で使える参照

「置換後」に指定して使います。

\$n	番号指定参照 ( )でグループ化した文字列を、\$n(nは1以上の整数)で参照します。 サクラエディタでは\$nの代わりに¥nも使用できます。
\${n}	(鬼) (安全な) 番号指定参照 後ろにそのまま続けて数字を書くことができます。
&	マッチした文字列全体
+	(鬼) 最後にマッチした部分文字列
+\$ {name} -\$ {name} [n]	(鬼) 名前指定参照 (Perl 5.10 互換、推奨)
¥k<name> ¥k' name'	(鬼) 名前指定参照 (鬼車準拠)
\$_ {name}	(鬼) 名前指定参照 (独自拡張、暫定仕様のため非推奨)

(鬼) はbregonig.dllのみ

## bregonig.dllへの変更による違い

- **¥w は2バイト文字も含む。**

¥w は [A-Za-z0-9] に加え、2バイト文字も含みます。これに応じて、¥W, ¥b, ¥B の動作も変更になります。

- **[ ] の中に [ を書くときは必ずエスケープが必要。**

文字集合の中で文字集合や積演算が使えるようになる代わりに、[ そのものを書くときは必ず ¥[ と書かなければならなくなりました。

- **¥c¥ の動作が異なる。**

Bregexp.dll では、¥c¥ は Ctrl+¥ を意味していますが、bregonig.dll では、¥c の次の ¥ はエスケープとして解釈されます。Ctrl+¥ を指定するには ¥c¥¥ とする必要があります。(Perl の仕様と異なるため、変更される可能性があります。)

- **置換で ¥ooo の動作が少し異なる。**

Perl の動作に近づけてあります。

- **Unicode版は i オプションの動作が異なる。**

大文字小文字を区別しないの動作が ASCII 範囲外にも影響します。

Unicode版サクラエディタの通常検索・置換も ASCII 範囲外の大文字小文字の考慮をしますが、2つが同じかどうかは未チェックです。

## 改行の検索

サクラエディタで改行(CRLF)を検索する場合は、

¥r¥n で検索して下さい。

改行(CR,LF,CRLFのすべて)を検索する場合は、

[¥r¥n]+ のように指定してください。

改行(CR,LF,CRLF,LF CR)と、最終行の行末を検索する場合は、

¥\$ と指定してください。

### ヒント

正規表現を検索、置換、Grepで利用する場合、「/」をエスケープしたり、「/」で囲ったりする必要はありません。

正規表現キーワードで使う場合は、/text¥/css/k のように「/」をエスケープするか m#text/css#k のように書く必要があります。

### 追加情報

検索、置換、Grepでは、正規表現ライブラリに渡す文字列を以下のようにしています。[DELIMITER] は、0xffのバイナリ表記とします。

Unicode版では U+FFFF です。

検索時のオプションは「m[DELIMITER]Pattern[DELIMITER]km」です。

置換時のオプションは「s[DELIMITER]PatternBefore[DELIMITER]PatternAfter[DELIMITER]km」です。

デリミタ相当の文字を検索・置換するにはエスケープが必要です。

また大文字小文字を区別しない場合は最後に「i」が付加されます。

すべて置換で「すべて置換は置換の繰返し」がOFFの場合は「g」が付加されます。

(mオプションが付いていますが、改行をまたいだ検索はできません。)