■正規表現サンプル集

（出典：http://hodade.adam.ne.jp/seiki/page.php?chapter_1）

１．正規表現（パターンマッチング）とはなにか？

正規表現は文字列の特徴をパターン化し、特有の記号で表現するものです。
正規表現を使うと、文字列の検索や置換をパターンで行う事ができるので、 多少の違いがある文字列でも、
1 つの検索文字列で検索することができます。

※Kapow で文字を認識させたり、置換させたりする以外に、通常のテキストエディタ等でも使うことができる便利なものですので、覚えて頂けると幸いです。

パターン化の例を一つ、以下に書いてみます。

```
たとえば日付「2010/02/22」
　　↓
これは「数字 スラッシュ 数字 スラッシュ 数字」というパターンである
　　↓
正規表現で表すと「¥d+/¥d+/¥d+」となる
```

一般的に正規表現に当てはまることを「一致する」または「マッチする」と言います。

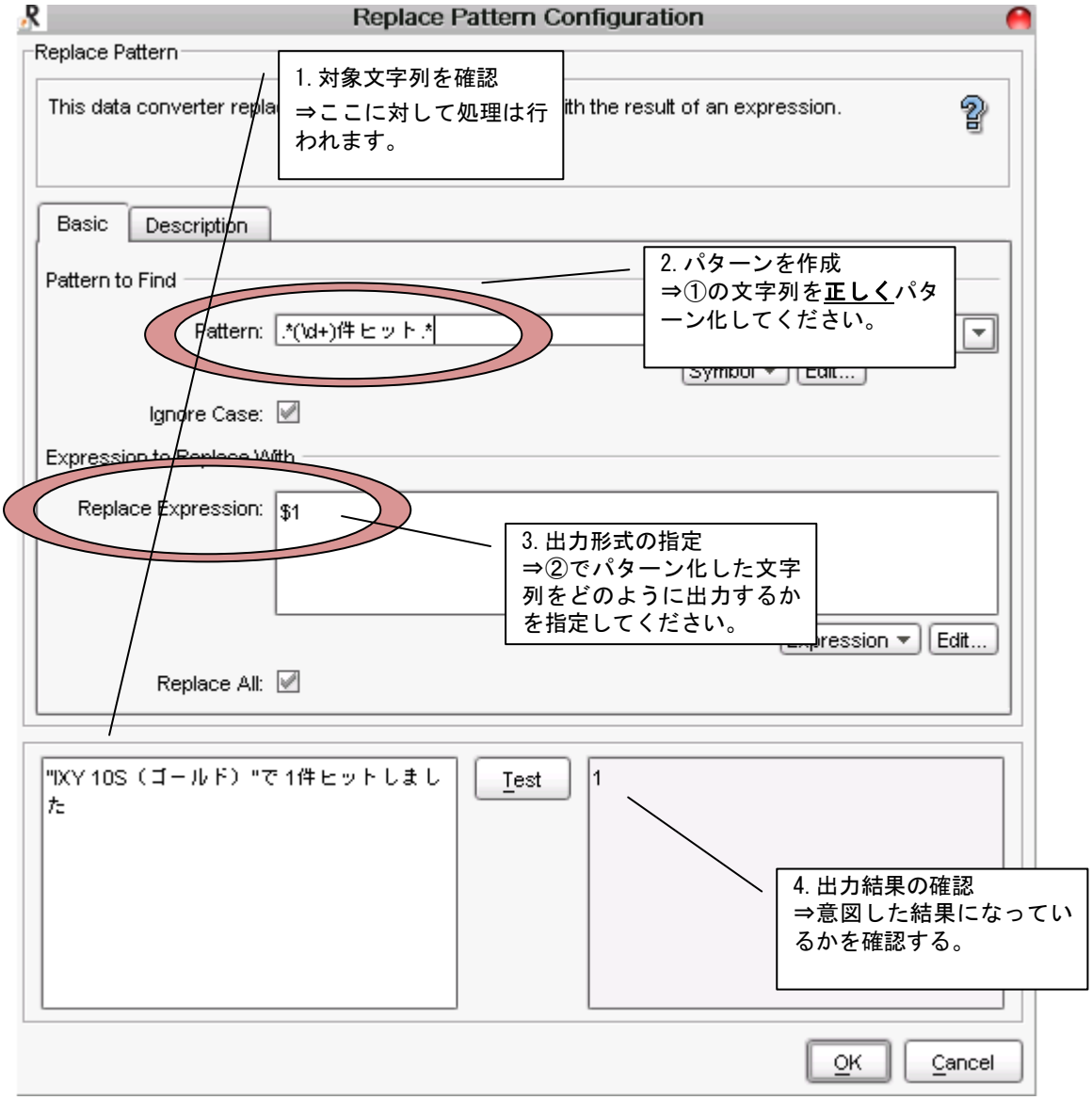
2.正規表現とは

正規表現とは、文字列の特徴(パターン)を記号化して表現するものです。製品番号、日付、プログラム言語などの文字列なら、おおよそ書式が決まっているので、正規表現で表すことが簡単にできます。
たとえば「F900i」,「SH900i」,「SO505i」の規則を日本語で表すと
「アルファベットが 1 文字以上 2 文字以内で、続いて 3 桁の数値がきて、続いて i という文字がくる」という規則になりますが、これを正規表現で表すと「[A-Z]{1,2}¥d{3}i」となります。

後方参照（サブパターン）について

■Kapow におけるパターンマッチングについて

Kapow では下記 Replace Pattern Configuration を始め、正規表現での文字列の抽出や、加工を行うことが多く、慣れてしまえば EXCEL の関数のように便利に使えるものですので、是非覚えて頂きたく思います。



後方参照（サブパターン）とは、文字列の一部を( )でくくると、その文字列を置換などをする際に
$1,$2,$3...という文字で参照することを言います。
たとえば、以下のように使います。

```
1: 対象文字列= 0173123456
2: Pattern  (¥d¥d¥d¥d)(¥d¥d)(¥d¥d¥d¥d)
3: Replace Expression $1+ “-“ + $2 + “-“ + $3
4: 0173-12-3456
```

これは、0173123456 の文字列に、パターンを定義し、出力形式を指定し結果を編集し、電話番号の形に整形したパターンマッチングの一例です。

2 行目の置換を詳しく説明すると、まず**パターンに（ ）が 3 つ**登場しています。

これは、**後方参照（サブパターン）を 3 つ作ることを意味します**。そして置換後では、$1,$2,$3 という変数で（）に一致した文字列を参照しています。( ) と $1,$2,$3 は前から順番に対応するので

(¥d¥d¥d¥d) → **$1**,　　(¥d¥d) → **$2**,　　(¥d¥d¥d¥d) → **$3**

となります。

※演習問題

HTML 形式で書かれたテキストがあります。タグをすべて消去しなさい。
タグ以外の文字は消してはいけません。

```
<HTML>
<BODY>
<H1>めんずらすぃホームページ</H1>
<P ALIGN="CENTER">ヨグキテケシタ。</P>
</BODY>
</HTML>
```

こたえ

IP アドレスが書かれたテキストがあります。IP アドレスの 4 つ目の項目を*で隠蔽しなさい。(例：192.168.10.*)

```
210.250.103.186
202.33.91.161
```

こたえ:

特別な意味を持つ文字①

正規表現は特別な意味を持つ文字と、通常の文字列を組み合わせて記述します。
以下に特別な意味を持つ文字をいくつか紹介します。

正規表現	説明
*	直前の文字の 0 回以上に一致
+	直前の文字の 1 回以上に一致
.	1 つの任意文字(A, B, C, ...)（¥n を除く）
?	直前の文字の 0 回または 1 回に一致
{n}	直前の文字がちょうど n 回に一致
{n,}	直前の文字が n 回以上に一致
{n,m}	直前の文字が n 回以上,m 回以下に一致
[xyz]	x か y か z の何れか 1 文字に一致
¥w	英数文字かアンダーバーを表す(a～z,A～Z,0～9,_)
¥d	数値文字を表す(0～9)

これらを使った例を以下に書きます。

正規表現	説明	一致する文字列
A*	0 個以上連続した A に一致	, A, AA, AAA, ...
A+	1 個以上連続した A に一致	A, AA, AAA, ...
A.	A の次に何れかの 1 文字がある場合に一致, 改行文字は除く	AB, A1, A.
AB?C	A と C の間に B がないか,B がある場合に一致	AC, ABC
A{3}	3 個の A に一致	AAA
A{2,4}	2 個以上,4 個以内の A に一致	AA, AAA, AAAA
[a-z]+	a～z の何れか,つまりアルファベット小文字を表す	value, ascii
[A-Z]+	A-Z の何れか,つまりアルファベット大文字を表す	VALUE, ASCII
¥w+	1 個以上の英数文字に一致	abc, a001, 001

正規表現	説明	一致する文字列
¥d+	1個以上の数字に一致	1, 12, 123, 001

※演習問題

以下の文字列があるとして、設問に答えなさい。

```
ニンジン, 03-1234-5678, yama@nakasato.pref.jp, F900i
ニンニク, 017-712-1234, abc@nakasato.pref.jp, SH900i
ニラ, 090-9635-2759, x-y-z555@anpan-man.com, SO505i
ニモ, 0172-53-1234, pocket@monsters.com, D505i
```

●電話番号のみに一致する正規表現を書きなさい。

●メールアドレスのみに一致する正規表現を書きなさい。

特別な意味を持つ文字②

前回の特殊文字に加えて、下記の特殊文字を使用します。

正規表現	説明	使用例	例の説明
|	選択	AA|BB|CC	AA または BB または CC に一致
()	グループ化	A(01|02|03)	A01 または A02 または A03 に一致
[^A]	A 以外の文字	[^ABC]+	DEF など 1 つ以上の ABC 以外の文字に一致
¥s	空白,タブ	¥s+	1 つ以上の空白に一致
¥S	空白文字以外	¥S+	1 つ以上の空白以外の文字に一致
¥d	数値文字	¥d+	123 など 1 つ以上の数字に一致
¥D	¥d 以外	¥D+	ABC など 1 つ以上の数字以外の文字に一致
¥w	英数文字かアンダーバー	¥w+	ABC123 など 1 つ以上の英数文字かアンダーバーに一致
¥W	¥w 以外の文字	¥W+	-+!?など 1 つ以上の英数文字かアンダーバー以外(記号)に一致
¥b	単語の境界(¥w と¥W の境界)	ABC¥bDEF	ABC と DEF の間に英数文字かアンダーバー以外の文字がある文字列に一致

選択とグループ化を使ってみる

選択とグループ化について詳しく説明します。

```
すいか|メロン
```

と書くと、スイカまたはメロンのどちらかに一致します。これを選択といい、選択肢を|で分けて記述します。

```
(あまい){2}すいか
```

と書くと、「あまいあまいすいか」に一致します。この例では、「すいか」をグループとし、それが 2 回出現することを表します。グループにしたい文字列を()で囲んで記述します。

上記で説明した、選択とグループを使って、

```
(すいか|メロン)(アイス|シェーク)
```

と記述すると、「すいかアイス」、「スイカシェーク」、「メロンアイス」、「メロンシェーク」のどれかに一致します。

※演習問題

以下はことわざとオヤジギャグを行単位で記述した文字列である。「犬」と「猫」で始まる文字列に一致する正規表現を書きなさい。

```
犬も歩けば棒にあたる
ニューヨークで入浴
猫に小判
梅は旨めぇ
猿も木から落ちる
飼い犬に手をかまれる
借りてきた猫
```

こたえ:

以下は麻雀牌を上がれる状態にし、行単位で記述した文字列である。各牌はスペース区切りで並べられている。九連宝燈のみに一致する正規表現を書きなさい。(九連宝燈は、１と９を３牌づつ、２～８を１牌づつ、そして１～９を１牌集めたもの。)

```
1萬 1萬 3萬 4萬 5萬 7筒 8筒 9筒 2索 3索 4索 5索 6索 7索
1萬 1萬 1萬 2萬 3萬 4萬 5萬 6萬 7萬 8萬 9萬 9萬 9萬 9萬
1萬 1萬 3萬 3萬 2筒 2筒 7筒 7筒 9筒 9筒 南 南 北 北
```

こたえ:

最短一致について

「.*」や「.+」を使うと、限りなく連続した文字を表します。
これは非常に便利なのですが、予想以上に長くマッチして、思ったように動かないことがあります。
これを解決するためには「?」を付加して**最短一致**するように仕向ける方法があります。

また、ここで説明する「?」は「直前の文字の０回または１回に一致」とは違います。
文字は同じですが、まったくの別物です。（記述する位置で見分けてください。）

たとえば、こういった<P>abc</P> HTML のＰタグをマッチングさせる場合、このように書いたとします。

```
< . * >
```

これでタグ内の文字も指定できるのですが、タグだけではなく、タグに挟まれた間の文字まで指定してしまいます。
（下線部分）

```
<P>abc</P>
```

理由は**最長一致モード**で動作しているからです。
動作をみると、途中に「>」が出てきているのですが、そこでは止まらず、最後の「>」までマッチしています。
つまり**最長一致**とは、できるだけ長くマッチングさせるという意味になります。
これを防ぐためには、以下のように、「>」の前に「?」（最短一致記号）をつけます。

```
<. *?>
```

こうすることにより「<」の後に出てくる、最初の「>」までとなり、タグ１つ分しか指定しなくなります。
下線部分のようにタグだけにマッチングします。

```
<P>abc</P>
```

エスケープが必要な文字

正規表現では特別な意味を持つ文字がいくつかあり、そのまま記述すると意味のある指定と解釈されます。
これらの文字を、通常の文字列として認識させる場合は、エスケープしてください。
（エスケープとは、¥ マークをつけて特殊動作を無効にさせること。）

エスケープ前	エスケープ後	注意点
¥	¥¥	エスケープを行う文字**そのもの**なので、¥ だけの記述はできません。 ¥ にマッチングさせたい場合は ¥¥ と記述してください。
*	¥*	
+	¥+	
.	¥.	
?	¥?	
{ }	¥{ ¥}	出現回数指定文字なのでエスケープが必要
( )	¥(¥)	エスケープしないと後方参照が作成される。またはグループ化される。
[ ]	¥[¥]	直前文字の出現回数指定文字なのでエスケープが必要
^	¥^	行頭を指定することになる。
$	¥$	行末を指定することになる。Perl の場合は、変数の先頭文字である。
-	¥-	[ ]の中に書く場合のみエスケープが必要
|	¥|	
/	¥/	Perl では / が正規表現の指定になるのでエスケープが必要。 言語によっては、″ がエスケープ必要となる。

■パターンマッチング・エクササイズ

- ある文字列の一部だけを変更したい

  置換前
  aaaa@te1.jp, bbbb@te1.net, cccc@te1.com
  置換後
  aaaa@te2.jp, bbbb@te2.net, cccc@te2.com

- 日付を検索する

  2005/09/30　（よくある日付の書式）

- 電話番号を検索する

  0123-12-1234　（加入電話、携帯電話も OK）

- メールアドレスを検索する

  aaa@bbb.com　（メールアドレス）

- かっこで囲まれた文字を検索する

  aaa@bbb.com　（メールアドレス）

文字列の一部だけを変更したい

置換したい文字

置換前

```
aaaa@te1.jp, bbbb@te1.net, cccc@te1.com
```

置換後

```
aaaa@te2.jp, bbbb@te2.net, cccc@te2.com
```

上記のようにあるフォーマットに基づいた文字列の一部だけを変更したい場合にこの正規表現を使用できます。
この例では、メールアドレスの書式のドメイン部分を te1 から te2 に変更しています。
そのほかの部分は変えません。

正規表現の書き方

検索文字列

```
(.*?)@te1¥.(.*?)
```

置換文字列

```
$1@te2¥.$2
```

上記の文字をそのまま検索文字列は”Pattern”欄に、置換文字列は”Replace Expression”欄入力してください。

正規表現の説明

検索文字列の「()」は、置換後にそのまま残す部分を表します。($1、$2 で
参照できます。)
検索文字列の「¥.」は「.」(ドット)そのものを表します。正規表現では「.」
は意味を持つのでエスケープします。
検索文字列の「.*」は空白、または 1 文字以上の文字を表します。そのすぐ後ろに「?」がついて
いるのは、最短一致を表します。

置換文字列の「$1」は検索文字列の「()」で囲んだ部分の 1 番目を表します。
置換文字列の「$2」は検索文字列の「()」で囲んだ部分の 2 番目を表します。

日付を検索する

検索したい文字

2005/09/30
2005/9/1

上記のような日付を検索したい場合にこの正規表現を使用できます。
数字がスラッシュで区切られていて、3 つ並んでいる場合です。

正規表現の書き方

¥d{4}/¥d{1,2}/¥d{1,2}
また、他に文字がない場合は **.＊{4}/.＊{1,2}/.＊{1,2}** という書き方も可

上記の文字をそのまま”Pattern”欄に入力してください。

正規表現の説明

「¥d」は半角数字の 0～9 を表します。
「{4}」は「¥d」が 4 個続く事を表します。
「{1,2}」は「¥d」が 1～2 個続く事を表します。
「/」は間に「/」が含まれている事を表します。

間の区切り文字が「-」である場合は、「¥d{2,4}-¥d{1,2}-¥d{1,2}」と書いてください。

桁数を気にしなければ「¥d+/¥d+/¥d+」と簡単に書くことも出来ます。

■実践演習

毎年のクリスマスにマッチするパターンを書いてください。

毎月 20 日にマッチするパターンを書いてください。

電話番号を検索する

検索したい文字

```
0123-12-1234
03-12-1234
090-1234-1234
```

上記の文字を検索したい場合にこの正規表現を使用できます。
上記の用に 1 行空いている部分です。

正規表現の書き方

```
¥d{2,4}-{2,4}-¥d{4}
```

上記の文字をそのまま”Pattern”欄に入力してください。

正規表現の説明

「¥d」は半角数字の 0～9 を表します。
「¥d{2,4}」は「¥d」が 2～4 個続く事を表します。
「¥d{4}」は「¥d」が 4 個続く事を表します。

電話番号の桁数にこだわらず検索したい場合は「¥d+-¥d+-¥d+」で検索可能です。

■実践サンプル

携帯電話にマッチするパターンを書いてください。

フリーダイヤルにマッチするパターンを書いてください。

メールアドレスを検索する

検索したい文字

```
hodade@gmail.com
store-news@amazon.co.jp
ranking@emagazine.rakuten.co.jp
noreply@postmaster.twitter.com
News_Japan@insideapple.apple.com
microsoft@e-mail.microsoft.com
info@dle.jp
```

上記のようなメールアドレスを検索したい場合にこの正規表現を使用できます。

正規表現の書き方

```
[¥w¥d_-]+@[¥w¥d_-]+¥.[¥w¥d._-]+
```

上記の文字をそのまま”Pattern”欄に入力してください。

正規表現の説明

「¥w」はA～Zを表します。
「¥d」は0～9を表します。
「[¥w¥d_-]」は[]内のいずれかの文字を表します。
「+」がつくと、前述の文字が1つ以上続くことを表します。
「@」はそのまま「@」の出現を表します。
「¥.」は「.」(ドット)の出現を表します。

実践サンプル

メールアドレスを厳しくチェックするにはこのようになります。

```
^[a-zA-Z0-9!$&*.=^`|~#%'+¥/?_{}-]+@([a-zA-Z0-9_-]+¥.)+[a-zA-Z]{2,4}$
```

かっこで囲まれた文字を検索する

検索したい文字

(111)（abc)（あいうえお）

上記のように、半角かっこで囲まれた文字を検索したい場合にこの正規表現を使用できます。

正規表現の書き方

¥(.＊?¥)

上記の文字をそのまま”Pattern”欄に入力してください。

正規表現の説明

「¥(」と「¥)」は半角かっこを表します。
（ ）は正規表現で意味のある文字なので、¥ でエスケープする必要があります。
「.+」は１文字以上のどんな文字でも一致します。
「?」は最短一致記号です。必要以上に大きくマッチングしないようにします。

かっこの種類が異なる場合は、適宜変えてください。
ただし、[ ] もエスケープが必要です。
¥[.+¥]

（ ）と [ ] の両方ににマッチさせたい場合は、以下になります。
[(¥[].+?[)¥]]