

# P e r 1 でサブリ

~ There is More Than One Way To Do It ! ~

2006/03/19  
in Kansai.pm 6th Meeting

なみかわみさき@Media Kiss Lab./MW

## TMTOWTDI とは？

There is More Than One Way To Do It の頭文字  
(他にもいろんな方法があるよ)

たとえば.....

「Hello,Perl!と print する」には？ いろんな答え方がある

\* `print "Hello,Perl!";`

\* `print "Hello,","Perl!";`

\* `$str = "Hello,Perl!";  
print "$str";`

などなど

## Perl でサブリのポイント

### パターン A

それでもいいんだけど、、、地道な(たぶん長い)ソース

関数や特殊変数のテクニックで「すっきり」

### パターン B

たぶんこんな答えだろうなあ、、、ふつうのソース

あっこんな書き方もあったんだ！ へえ～で「さっぱり」

# 頭をやわらかくして、良い刺激を受けて元気になろう(^ ^)

## 第1問

1 から 10 を  
1 ずつ増やして掛けた答えは？

答えを考えてみよう

## 解答例 1

```
$kotae = (1*2*3*4*5*6*7*8*9*10);
```

```
print "1 から 10 まで 1 ずつ増やして掛けた";  
print "こたえは $kotae です.¥n";
```

## 解答例 2 (もやっと)

```
$kakerukazu = 1;
```

```
$moyatto_kotae = ($kakerukazu * ($kakerukazu + 1) * ($kakerukazu + 2) *  
($kakerukazu + 3) * ($kakerukazu + 4) * ($kakerukazu + 5) * ($kakerukazu + 6) * ($kakerukazu + 7) * ($kakerukazu + 8) * ($kakerukazu + 9)  
);
```

```
print "こたえは $moyatto_kotae です.¥n";
```

### 解答例 3 ( すっきり )

```
#!/usr/bin/perl
```

```
$sukkiri_kotae = 1;
```

```
foreach $i (1..10) { $sukkiri_kotae *= $i; }
```

```
print "1 から 10 まで 1 ずつ増やして掛けた";
```

```
print "こたえは $sukkiri_kotae です.\n";
```

## 第2問

「1」「3」「2」「4」を、  
「1」「2」「3」「4」に並び替える。

答えを考えてみよう

## 解答例 1

```
#!/usr/bin/perl
```

```
@hairetu = ( 1,3,2,4 );
```

```
@kaitou = sort (@hairetu); #sort で文字コード順にソート
```

```
print @kaitou;
```

#数値のソートは、<=>を用いたほうがよい

```
@hairetu_kazude = ( 1,3,2,4 );
```

```
@narabikaeta = sort { $a <=> $b } @hairetu_kazude;
```

```
print @narabikaeta;
```



## 配列の基礎

@hairetu = ( 1,3,2,4 ); とすると、、

1	3	2	4
[0]	[1]	[2]	[3]

となる（先頭は[1]でなくて[0]です）

#いきなり \$hairetu[100] = 5; とすると  
[0] ~ [99]までの値は空っぽ（undef）

undef	undef	undef	5
[0]	[1]	[2]....	[100]

## はしもとさんのこたえ

```
use strict;  
my @ary = ( 1, 3, 2, 4 );  
my @result;  
$result[$_] = $_ for @ary;  
@result = grep { defined } @result;  
print join ', ', @result;
```

# 配列の要素にそれぞれの値を利用  
# \$result[0]が undef になる

そのほかの解答例（もやっど？）

push, pop, shift, unshift を駆使するとか、、

push  
pop  
shift  
unshift

### 第3問

Kansaihogehoge.pm から hogehoge を抜き取って  
Kansai.pm にしてください。

答えを考えてみよう  
ヒント：正規表現

みしまさんのこたえ (参考)

echo Kansaihogehoge.pm | perl -pe s/hoge//g

#Perl を オプション付きで起動する perl -h , perldoc perlrun

perl -cw : 文法チェック

perl -e : 一行プログラム

perl -p : while ( <> ) { . . . } みたいになる

## 置き換えをする - 置換演算子

s/これとマッチするやつを/これに置き換える/;

マッチする = 文字列が同一、じゃないよ  
うしろに g があるとマッチしたすべてを置き換え  
( Kansaihogehogehogehoge.pm でもできる )

### 解答例

```
my $okikaeru;  
$okikaeru = "Kansaihogehoge.pm";  
$okikaeru = ~ s/hoge//g;  
print $okikaeru;
```

hogehoge にマッチさせてみよう

hoge (うしろに g) または hogehoge にマッチできるような正規表現を考える

\* `s/hogehoge//;`

\* `s/hoge//g;`

\* `s/h(.*)h¥1//` : h+ 何か + h+ 何か

いろいろ考えてみよう

## 特殊変数を使ってみる

```
$_ = "Kansaihogehoge.pm";  
/hogehoge/;
```

```
# $& は hogehoge ( マッチしている部分 )  
# $` は Kansai ( マッチしている部分より前 )  
# $' は .pm ( マッチしている部分より後 )
```

```
print "$`"."$'";
```



## 参考資料

- \* 初めての Perl (第2版) ISBN:4900900818  
Randal L. Schwartz, Tom Christiansen, 近藤嘉雪 訳,  
<http://www.oreilly.co.jp/books/4873111269/> (第3版)
- \* Perl プログラミングレッスン入門編 ISBN:4797312211  
結城浩,<http://www.hyuki.com/pb/index.html>
- \* 結城浩の Perl クイズ,ISBN:4797321261  
<http://www.hyuki.com/pq/pqbook.html>
- \* perldocjp,<http://perldoc.jp/>
- \* そして perldoc

Special Thanks : Kansai.pm のみなさん