本日のAcme


今月の標語
非互換は
革新の母
Perl 6.0の
正式名称が
Perl Vistaに
決定いたしました
Perl
Vista™
なら今回は
パスしても
いいですか？
Jo!

はじめに

本書はPerlモジュールの集積場、CPAN[1]に存するモジュールの中でも、最も~~痛い~~偉大な名前空間”Acme”を冠するモジュール全てを網羅しきった空前絶後の大著『Acme大全』（モライリー出版刊行予定）から、厳選[2]された30有余のモジュールを取り上げた冊子です。

Acmeを冠したモジュール（以下Acmeモジュール）は、様々な場面で役に立つものが多く、たくさんの人に愛されています（と思います、たぶん）。

本書では、四季折々の重要なイベントに相応しいAcmeモジュールの紹介と解説を行っています。ぜひご活用ください（何に？）。また人生の中で直面する様々な苦難に際して、指針となるようなAcmeモジュールのリストも掲載しています（指針？）。

本書が対象としている読者は主に、CPANを活用している人、Perl使いではないが他言語に興味がある人、そしてPerlが嫌いな人々です。
CPANを活用している方には今更感があるかもしれません。ですが、まだAcmeモジュールをよく知らない人にはぜひ知ってもらいたいと願っています。また、Perl使いではなくとも他言語に興味がある方は、ぜひこの機会にPerlの魅力を知ってもらえればと考えています。
そしてPerl嫌いの人は、本書を通じて一層「Perlアホだな、ダサ！」という思いを強めていただければと思います。最後は本書を薪にもできますし。

掲載しているサンプルコードは、たいていの場合、各モジュールのドキュメントからコピー・編集していますが、引用、内容の間違い等は、全て本書作者にその責があります。

最後に、あなたのPerl生活（もしくはPerlと全然関係ない生活）が、素晴らしいものになるよう、また本書が売れて次回『Acme大全』を本書作者が出せることを祈ってやみません。

---

[1] CPAN（Comprehensive Perl Archive Network）……「Perlのライブラリ・モジュールソフトウェアやその他のPerlで書かれたソフトウェアを集めた巨大なアーカイブで、世界中のサーバにその内容がミラー（コピー）されている。再利用性・汎用性の高いモジュールが登録されており、Perlプログラマができるだけ車輪の再発明をせずに済むための支援環境となっている。登録モジュールの検索システムも提供されているため、Perl プログラマは望む機能を持ったモジュールを容易に入手することができる。」（2008-12-22日時点のWikipedia - http://ja.wikipedia.org/wiki/CPAN より）
[2] 「時間と余力の都合で」の平安中期における表現。紫式部も使ったとか使わなかったとか。

元旦　Acme::Bleach

一年の計は元旦にあり。とか言いながら毎年無計画に始まるのが人情。

　やはり、年の初めはすっきりと。あるいは書き初め。とうことでこれ。Acme::Bleach。漂白。プログラムソースを本当に綺麗にしてくれる！　Acmeモジュールの真骨頂。

```
use Acme::Bleach;
print "Hello world";
```

上記のプログラムをbleach.plとでもして実行するとファイルが書き換わり、下のように。

```
use Acme::Bleach;
```

あれ、Hello worldはどこに……？
念のためbleach.plを実行すると、ちゃんとHello worldが表示される。
　第 1 日目の記念に少し詳しく解説。Acme::Bleachのソースをみてみると、

```
my $tie = " \t"x8;
sub whiten { local $_ = unpack "b*", pop; tr/01/ \t/; s/(.{9})/$1\n/g; $tie.$_ }
sub brighten { local $_ = pop; s/^$tie|[^ \t]//g; tr/ \t/01/; pack "b*", $_ }
sub dirty { $_[0] =~ /\S/ }
sub dress { $_[0] =~ /^$tie/ }
open 0 or print "Can't rebleach '$0'\n" and exit;
(my $shirt = join "", <0>) =~ s/.*^\s*use\s+Acme::Bleach\s*;\n//sm;
local $SIG{__WARN__} = \&dirty;
do {eval brighten $shirt; exit} unless dirty $shirt && not dress $shirt;
open 0, ">$0" or print "Cannot bleach '$0'\n" and exit;
print {0} "use Acme::Bleach;\n", whiten $shirt and exit;
```

　まずopen 0で自分自身（use Acme::Bleachしてるプログラム）を開き、use Acme::Bleachの部分は消す。先頭が" \t"8回で始まっていて、かつ、" "と"\t"（スペースストタブと）だけで構成されていれば（漂白されている）、それぞれを0と1に変換して、pack "b*"で2進数から文字列に復元して実行。
　そうでなければ逆に、unpack "b*"でソースコードを2進数にして、0はスペース1はタブに変換（＋最初に" \t"を8回繰り返す）。そして先頭行にuse Acme::Bleachをくっつけて、上書き。これで漂白完了。

　これは様々な応用が効くので、同系モジュールが多数ある（他の月のイベントでも取り上げている）。

成人式　Acme::Playmate

ヤポネではこの通過儀礼において馬鹿騒ぎを行うことになっている。
式の中止やらタイーホなどを通じて大人の仲間入りをする。

成人ということで、Acme::Playmate。Playboyのカレンダーガールのデータ[1]をとってくるという、たいへん即物的な代物。

```
use Acme::Playmate;
my $playmate = Acme::Playmate->new("2008", "12");
print "Details for playmate " . $playmate->{ "Name" } . "\n";
print "Bust" . $playmate->{ "Bust" } . "\n";
print "Waist" . $playmate->{ "Waist" } . "\n";
print "Hips" . $playmate->{ "Hips" } . "\n";
```

LWP::UserAgentを使って直接webサイトからデータをとってくるという、これまた他のAcmeでも活用されている代表的な手法。

しかしよく考えたら、こういうのって成人になる前から見てるのか。

---

[1] http://www.playboy.com/girls/playmates/directory/ 下にある。

節分　Acme::Ook

歳の数だけ落花生を食べる日。
だと作者は思っていたが、どうやら大豆を食べる日らしい。

2月にして、もうネタ切れだ。節分に関連するAcmeモジュールなんて見つからない。困った。

節分……　→　豆まき　→　鬼は外……　鬼　→　オーク(Orc)　→　ウーク（Ook）！

はい、Acme::Ook。（……この企画無理がありすぎ。）Ookはオランウータン用プログラミング言語。「Ook.」「Ook!」「Ook?」（オランウータンの鳴き声）の3つのうち、2つの組み合わせで一つの命令を表現する。これをPerlで実行できるという、誠にオランウータンにも優しいPerlは偉大なりぃぃいいい！

```
use Acme::Ook;
my $Ook = Acme::Ook->new;
$Ook->Ook($Ook);
```

で、下がHello Worldを表示させるコード。

```
Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook. Ook. Ook? Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook. Ook. Ook!
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook!
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook. Ook.
Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook?
Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook? Ook.
Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook! Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook! Ook! Ook! Ook! Ook. Ook. Ook? Ook. Ook? Ook. Ook. Ook! Ook.
```

Perl成分の3%はオランウータンへの優しさで出来ています。

バレンタインデー　Acme::Roman

モテる者とモテざる者との闘争日。並びに、オフィスにおいて
ご婦人が年一度の義理立てをする日。

　バレンタインデー、それは男のロマン！　ということにしておく。
そこで、Acme::Roman！

```
use Acme::Roman;
print Ⅰ + Ⅱ;  # 結果はⅢ。
print Ⅹ * Ⅱ;  # 結果はXX。
```

そのロマンじゃなかった！

ひな祭り　Acme::Drunk

歳の数だけ雛あられを食べる日。もしくは歳の数だけ甘酒[1]を飲む日。
おいおい、何杯飲むきだ。

　白酒飲みすぎて泥酔、なんてことにならないように、このモジュールを携帯しましょう。

```
use Acme::Drunk;
my $bac = drunk(
  gender      => FEMALE,
  hours       => 2, # 飲み始めた時間
  body_weight  => 160, # ポンド換算
  alcohol_weight => 3, # オンス換算
);
$bac >= 0.08 ? call_cab() : walk_home();
```

　血中アルコール濃度を計算してくれる。計算方法はWidmark式。ドキュメントには下記のようにあった。

```
  ( ( (FlozEtOH * 0.0514 lb/flozEtOH) * 1.044 g/ml )
    / (Lbs of person * Widmark r g%/mlhr) )
  - (Hours since first drink * Widmark beta)
  = BAC g%/ml = BAC g/dL = BAC% w/v
```

酔っぱらった？

　作者はリットル、キログラム換算にできるようにラッパーモジュールを書いたのだが、CPANにアップはしていない。拙サイト http::/www.donzoko.net/ のどこかにあるはずだが……

---

[1]正しくは白酒。でもこっちの方だと思っている人多いでしょ？　作者もそうでした。

ホワイトデー　Acme::Echo

徴税人が、バレンタインデーの勝者と義理立てを受けた者の戸口に立つ日。

　ホワイトデー、それは女性のロマン！　ということもなさそうだ……　お返しということだから、とりあえずAcme::Echoでいってみる。

```
use Acme::Echo qw/lines/;
print "hello world\n";
```

すると、実行しているコード自体がコンソール画面に表示される（この場合‘print "hello world\n";’）。

卒業式　Acme::EyeDrops

何も得ることが無かったであろうことを忘れて、感動する日。

　とにもかくにも、卒業式といえば涙がつきもの。涙、ということでAcme::EyeDrops。プログラムソースを文字も数字も使わないものに変換する。　卒業式とは何の関係ない。

hoge.pl

```
#!/usr/bin/perl -w
use strict;
print "Hello World!\n";
```

別プログラム

```
use Acme::EyeDrops qw(sightly);
print sightly( { Shape  => 'camel', SourceFile  => 'hoge.pl' } );
```

　すると、素敵コードになる。実行するとちゃんと”Hello World!”が表示される。Acme::Bleachとはまた違った趣があるね。もちろんcamel以外にもcoffeeとかlarryとか色々な種類がある。

エイプリルフール　Acme::Meow

この本のことだ。または作者の結婚記念日。

　Acme::Meowは猫を生成する。餌をやったり、かまったり……　そうすると、喜んでゴロゴロと喉を鳴らしたり。だたし、まだほとんど完成していないモジュールなので、出来ることはほとんどない。寝ているかどうかをチェックするis_sleep() も、常に偽を返すだけだし。

```
use Acme::Meow;
my $kitty = Acme::Meow->new();
$kitty->pet();
$kitty->feed();
```

　なんでこのモジュールを取り上げたかというと、作者のIDがFOOLISHだったから。

入学式　Acme::ButFirst

何も得ることがなかったことを後で確認するために
通らなければならない最初の儀式。

　入学式といえば新しい生活の最初のイベント。ということで、これ。but firstブロックが導入できる。　名前の通り、先に処理されるブロック。
　このモジュールが利用しているテクニックはFilter::Simple。　Perlがソースをコンパイルする前に、好き勝手に書き換えてしまうFilterモジュールがあるが、それをさらに使いやすくしたもの。Filter::Simpleを使用したAcmeモジュールはたくさんあるので、他のところでも取り上げている。

```
use Acme::ButFirst;
{
    print "Good morning!\n";
} but first {
    print "I need a coffee\n";
}
```

メーデー　Acme::Morse

メーデー。こちらは作者。締め切り三日前にて原稿執筆の遭難中、代筆を求む。

　聞け万国の労働者！　轟きわたるメーデーの　示威者に起こる足どりと　未来をつぐる鬨の声！[1]　しかし、そんなイベントに関連するAcmeモジュールは見当たらないので、代わりにAcme::Bleachに同梱のAcme::Morse。　"　ど"\　の代わりに". ど"-　で　変。換

```
use Acme::Morse;
print "Hello world\n";
  ↓
use Acme::Morse;
.--.-..--..---.-.--..--.-..--..---.-.--.
....---..-..---.-..-.--..---.--...-.---.
.....-...-...-.....-..-.-.-..--...--.--.
..--.--.----.--......-..---.---.----.--.
.-..---...--.--...-..--...---.-..---.--.
.-...-..--.---...-.-.....-.-....
```

　他にも聞こえるようにMIDIファイルを生成するAcme::Morse::Audibleとか、コメント部分だけ本当にモールス信号にするAcme::MorseCommentsとか。まあ、とにかくメーデー、メーデー[2]……

```
use Acme::MorseComments;
# This is comment.
print "hoge\n";
  ↓
# - .... .. ...  .. ...  -.-. --- -- -- . -. - .-.-.-
print "hoge\n";
```

---

[1] http://www.mahoroba.ne.jp/~gonbe007/hog/shouka/mayday.html参照

[2] モールス信号なら遭難通信はSOSだろう。ただし現在は使われない。

端午の節句　Acme::Tango

タンゴ、タンゴ、クロネコのタンゴ。つまり、猫を可愛がる日か何かだと思われる。

　素晴らしい！　端午の節句用モジュールがあったよ！

　えーと、'#00bbff'のような16進数表記の色を渡すと、彩度と明度は同じで、オレンジがかった色に変換するんだって。なんじゃこりゃ？[1]

```
use Acme::Tango;
my $hex_string = Acme::Tango::drink('#00bbff');
my $hex_string = Acme::Tango::drink('#00bbcc', 'apple');
```

オレンジ以外にも、林檎やレモンにもできるらしい。

---

[1] http://www.tango.com/を見よと書いてあるが、既にネット上には存在していない。

母の日　Acme::Prereq::A

父の日を見よ！

　Acme::Prereq::A。とくに何するわけでもなくて、Acme::Prereq::Bを必要としている。手抜きをしたいからこのモジュールを選んだのではなく、深い考えがあってのこと。ということではない。手抜きがしたかったのだ。　また、同じ作者によるAcme::Prereq::Noneとういのもある。

父の日　Acme::Prereq::B

母の日を見よ！

　Acme::Prereq::B。とくに何するわけでもなくて、Acme::Prereq::Aを必要としている。手抜きをしたいからこのモジュールを選んだのではなく、深い考えがあってのこと。ということではない。手抜きがしたかったのだ。　また、同じ作者によるAcme::Prereq::Noneとういのもある。

七夕　Acme::Stardate

歳の数だけ笹の葉を……

　七夕にちなんで星という単語が入っているモジュールを。ということで、Acme::Stardate。あなた用の「宇宙時間」を返してくれる優れもの！　なのだが、うーん……

```
use Acme::Stardate;
my $t = stardate();
```

現在時刻をyyyymmdd.fffffで返してくれる。ソースでは

```
use POSIX 'strftime';

sub stardate {
    strftime("%Y%m%d.", gmtime). int(time%86400/86400 * 100000)
}
```

　Acme::MetaSyntactic::starsの方がよかったかな？　こちらは
　　http://en.wikipedia.org/wiki/List_of_traditional_star_names
に載っている伝統的な星の名前が入っている（Acme::MetaSyntacticについてはクリスマスの項にて）。

夏休み　Acme::Nothing

素晴らしい、休みだ。だからこのページも休みだ。

夏休み……

　だけだと何なので、Acme::Nothing。このモジュール自体が何もしないわけではない。ソースをみると結構色々頑張っている。useやrequireしても、モジュールをインストールしなくなる。

ソース：

```
 @INC = sub {
    local *__ANON__ = $_[1];
    $INC{ $_[1] } = $_[1];
    open my $fh, '<', \!$[ or die;
    return $fh;
 };
 Internals::SvREADONLY( $_, 1 ) for @INC;
 Internals::SvREADONLY( @INC, 1 );
```

　@INCはPerlがモジュールを探しに行くディレクトリパスを入れる配列。ここに無名サブルーチンなんかを入れると、パスを探しにいく代わりに、このルーチンが実行される（第一引数はサブルーチン自身、第二引数にモジュール名）。返すのはファイルハンドル（通常はモジュールファイルということに）。

$[は配列のインデックスの最初の値を示し、通常は0。なので!$[は真値になる。だから

```
    open my $fh, '<', \!$[ or die;
    return $fh;
```

は、1という文字だけのファイルのハンドルを返すのと一緒[1]。結局何もしないモジュールがロードされることになる。

　あとInternals::SvREADONLY（ $var, 1 ）で変数のREADONLYフラグが立つ（つまり読み込みのみで書き込み不可になる）。もう何をやっても新規にモジュールをロードすることはできないのだろう……[2]。

---

[1] openの第三引数がスカラーリファレンスのとき、ファイルファンドルは、リファレンス元の文字列が入ったファイルを参照するように振舞う。
[2] 似たようなモジュールに、これまた似たような名前のAcme::Anythingがある。

敬老の日　Acme::Yoda

敬われるような歳の取り方をしたいじゃないかね。ええ？

　年寄りを敬う、ということで、とりあえずAcme::Yodaを紹介してみる。

　映画「スターウォーズ」に出てくるジェダイ・マスター、ヨーダ。このモジュールは与えたテキストを彼の話し方風に変換する。はず。

```
use Acme::Yoda;
my $y = Acme::Yoda->new();
my $translated = $y->yoda('I am your father');
```

ソースをみたところ、たいして変わらないみたい。発展途上のまま放置されてしまっているモジュールといえよう……

　なお、「ヨーダ語概論」という論文があるようなのだが、元サイトは見当たらない[1]。しかし、日本語に訳しているサイト[2]はあるので、参照されてはいかがだろうか。

---

[1] http://www.yoda-speak.org/yodish.html
[2] http://eigodestarwars.web.fc2.com/yodish.html

中秋の名月　Acme::Apache::Werewolf

歳の数だけ月のウサギの数を数える日。

　これはmod_perlで使用するApacheモジュール。満月になるとサイトにアクセスしてもForbiddenにしてくれる。これで狼人間が来ても安心
　Apacheが狼人間になるわけではない。

Astro::MoonPhase(http://search.cpan.org/dist/Astro-MoonPhase/)を利用して月齢を取得。
MoonLengthで満月とみなす期間を設定できる。

```
<Directory /myhome>
    PerlAccessHandler Acme::Apache::Werewolf
    PerlSetVar MoonLength 2
</Directory>
```

体育の日　Acme::Kensiro

ヤポネのTokioにおいて良く晴れる日。だった。

　これまた困った。どんなモジュール紹介しろというのか？

　　体育　→　スポーツ　→　肉体……　肉体派　→　ケンシロウ！

　すまん、本当にすまん。というわけでAcme::Kensiro。

前提：ケンシロウ進数。
　　2進数「0」「1」をそれぞれ「た」と「あ」で表す。

```
use Acme::Kensiro;
kensiro(16); # => あたたたた
```

ケンシロウ進数はhttp://www.asahi-net.or.jp/~rc4t-ishr/kensiro.htmlに説明されている。
これによると10進0→あ　1→た　4→あたた　512→あたたたたたたたたたた

文化の日　Acme::Don’t

文化放送とか文化包丁とか文化シャッターとか文化女中器を促進する日。

　ううん、もうネタがない。本当にない。どうしても関連モジュールが思いつかない！　この「無力感」をAcme::Don’tに託す。

```
use Acme::Don't;
don't { print "Hello World.\n" };
```

doの反対なんだ。何もしないんだ[1]。ところでdon’tの部分は、これはかつて（Perl4とか）パッケージ区切りを「’」で表して、それが今でも互換性があるために実現できる小ネタ。Perl5風に書けば don::tということに。

---

[1] 2008年9月、入れ子になったdon’tが否定の否定でdoにならないのはおかしい、という疑問が出され、改良版が考え出された（http://d.hatena.ne.jp/gfx/20080929/1222669383）。
今後機会があったら取り上げたいと思う。

勤労感謝の日　Acme::Everything

私にも感謝してみてください。お願いします。

　大変勤勉なAcme::Everythingを紹介しよう。どんなモジュールのどんなメソッドでも、実行環境に存在しなければ、その場でCPANからインストールしてきて使えるようにしてしまう優れものだ。内部的にはClass::Autouseを使っている。なので、そっちの説明をするべきだが、余白が中途半端なので、ここまで。

　勤労なことはわかったが、感謝が抜けていた！

クリスマス　Acme::MetaSyntactic

歳の数だけ赤いべべ着たヒゲもじゃの爺さんになる日。

　サンプル用のコードを書くときに変数の名前を捻り出すのに苦労した経験はないだろうか？適当なのを思いつかなくて結局 $foo とかしたり。そんなあなたにサンタクロースから素敵なプレゼント、Acme::MetaSyntactic。

```
use Acme::MetaSyntactic;
my $meta = Acme::MetaSyntactic->new('batman');
print $meta->name;        # ランダムに1つ取り出す
my @array = $meta->name(3); # 3個いっぺんに
```

　これでバットマンで使われる「音」[1]—'awkkkkkk' だとか'zzzzzwap'が手に入る！
　……あんまり嬉しくないね。他にも様々なテーマがあり、007の映画タイトルを集めた'jamesbond'、元素記号を集めた'elements'、ゼラズニィのアンバー 'amber'などなど、その数100個近く。現在も増殖中。

　また、Acme::MetaSyntactic::hogehoge（小文字で始める）なモジュールをこさえれば新しいテーマも自作可能。

　ちなみにAcme::MetaSyntactic->new()にテーマ名を渡さないとデフォルトのテーマはAcme::MetaSyntactic::fooに[2]。

---

[1] http://www.usfamily.net/web/wpattinson/otr/batman/batfight.htm にある。
[2] この項目は拙日誌http://www.donzoko.net/cgi-bin/tdiary/20050531.htmlを再利用。

大晦日　Acme

やった！　この不毛な文章もこれで終わりだ！

　最後に、そのものずばりAcme。

　完全無欠なモジュールのためのベースクラス。UNIVERSAL::is_acmeが導入され、ありとあらゆるクラスがis_acmeメソッドを持つことに。　そしてAcmeを継承しているクラスは真を返す。このモジュールをベースクラス（base）として使えば、頂点(summit)になれる！[1]とこのモジュールのドキュメントには書いてある。

```
use Acme;
print "Acme!" if acme->is_acme and acme->is_perfect;
```

なおis_acmeのエイリアスとして、もれなく

```
is_perfect
is_the_highest_point
is_the_highest_stage
is_the_highest_point_or_stage
is_one_that_represents_perfection_of_the_thing_expressed
is_the_bizzity_bomb
is_teh_shiznit
```

が導入される。

---

[1] Acmeは絶頂を意味するので、それにかけているのだろう。

苦難に際して

人は人生において、様々な困難に直面する。挫けそうな時、決断を迫られる時。そんな局面において、Acmeモジュールがあなたに何らかの指針を与えてくれるかもしれない。

納期が近い時

```
use Acme::MetaSyntactic::magic
```

会社を離れて遠くへ行った時

```
use Acme::Magic8Ball
```

滅多に発生しないバグに気づいた時

```
use Acme::Magic8Ball
```

コピペで済ませたい時

```
use Acme::Magic8Ball
```

納期がもっと近い時

```
use Acme::Magic8Ball
```

納期が前倒しになりそうな時

```
use Acme::Magic8Ball
```

納期のことを忘れたい時

```
use Acme::Magic8Ball
```

上司になじられた時

```
use Acme::Damn
```

隣の同僚が居眠りをしていた時

```
use Acme::Curse
```

Perlに失望した時

```
use Acme::Pythonic
```

Rubyの誘惑に負けそうな時

```
use Acme::Brainfuck
```

明日娘が嫁ぐ時

u s e　A c m e : : N o

明日妹が嫁ぐ時

u s e　A c m e : : N o

明日姉が嫁ぐ時

u s e　A c m e : : N o

明日幼馴染が嫁ぐ時

u s e　A c m e : : N o

この本がつまらなかった時

薪にする

おわりに

本書は、2008年8月の夏コミで出した『Pythons & PerlMongers』に続く、単体Perl同人誌第２弾です。単体というのは、Perl同人自体は2006年12月の冬コミから書いているのですが、それらは別サークル「スタジオな労働」の機関誌もとい、同人誌『マルチ中毒』の中の一記事だからです。

今回は前回に比べて一層時間がなかったよ！　本当しんどかった。だいたい、何を書くかが決まるのが遅かった。11月末になってやっと決まったのです。それから仕事が忙しくてほとんど作業は進まず、もうコミケ当日まで残すところ数日です。でもまあ、終わって良かった良かった。本書が10冊売れたら『Acme大全』をつくります。これはCPANにあがっているAcmeという文字を含んだ全ディストリビューション277個（2008年11月28日時点）を紹介するという、まことに意欲的（というより猟奇的）な本になるはずです！

というか、もともとは『Acme大全』の方をつくる予定のものだったのですが、まあ、無理ですよね。時間的にも体力的にも。なので、今回は20〜30個ぐらいの紹介ですむように、イベント行事に即したものになりました。でもこれはこれで、難しい。だいたい各イベントに即したAcmeモジュールなんてそうそうないですよ。当たり前ですが。
結局、全般的に非常に苦しいネタだけになってしまったのですが（特に「苦難に際して」なんて、本当に何の説明もないし）、どうか、ご容赦くださいませ。

あと、本当はもっと色々紹介したいモジュールもあったのですが、それは次の機会に頑張ります。

それでは皆様、またお会いしましょう。

2008年12月

Pubooに移植するにあたって

どんぞこ楽屋が2008年12月に冬コミに出した同人誌です。コラムと一部のページが省かれています。それと一部のページを訂正。あと脚注のリンクは、Wordのデータからコピペしてるので、機能していませんね。

これは『Acme大全』の前身ともいえるコピー誌でした。「10冊売れたら『Acme大全』出す」と、あとがきに書いてしまったために、翌年本当に『Acme大全2009』を出す羽目に。今見返すと結構Acme大全に流用している部分があります。

まかまか@どんぞこ楽屋

奥付

まかまか般若波羅蜜 ＠どんぞこ楽屋
http://www.donzoko.net/gakuya/

表紙イラスト：鬼才、三鷹畳
※表示タイトルに使用したフォント、あんずもじは京風子（Kyoko）氏が提供されているものを利用しました。
（あんずいろapricot×color　URL：http://www8.plala.or.jp/p_dolce/）