

Everything but the Secret Sauce

Modules, Magic and Monstrosities created for



Collaborative todo lists that work the way you do.

1. Create

Braindump
your tasks, tag
them, set due
dates, and
attach notes

2. Share

Set up
reminders for
yourself, create
groups, and
share tasks
with others

3. Review

Check out your
tasks, set
priorities, and
make decisions

4. Bee

productive!

(Get used to
the bee puns.)

[Take the full tour! >>](#)

Everything but the Secret Sauce



Contains
No Jifty!

Modules, Magic and Monstrosities created for

Hiveminder™

Collaborative todo lists that work the way you do.

1. Create

Braindump
your tasks, tag
them, set due
dates, and
attach notes

2. Share

Set up
reminders for
yourself, create
groups, and
share tasks
with others

3. Review

Check out your
tasks, set
priorities, and
make decisions

4. Bee

productive!

(Get used to
the bee puns.)

[Take the full tour! >>](#)

The background of the slide is a complex, abstract pattern of glowing lines. These lines are primarily yellow and white, forming a dense, web-like structure. Interspersed among these are several distinct, wavy lines in red and blue. The overall effect is reminiscent of a microscopic view of neural activity or a complex data visualization. The text is centered over this pattern.

Jesse Vincent

Best Practical



We make Hiveminder

私たちはHiveminderをつくっています



It's not opensource

オープンソースではありません



We built lots of cool
stuff for Hiveminder

Hiveminder用にクールなツールをたくさん作りました



We opensource
everything we can

オープンにできるものは全部公開しています

Today's Agenda

Ten tools and techniques to help you:

Find bugs faster バグの検出をもっと素早く

Build web apps ウェブアプリの構築

Ship software ソフトのリリース

Get input from users ユーザからの入力を受けつける

Own the Inbox 受信箱を用意する

今日の話



Find bugs faster

バグの検出をもっと素早く

TAP-Harness-Remote

Run your tests over there.

あちこちでテストを実行する

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from traffic visible. The sky is a deep blue with some wispy clouds. The title 'The Problem' is overlaid in large black text.

The Problem

問題

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from traffic visible. The sky is a deep blue with some wispy clouds.

Hiveminder has
lots of tests

Hiveminderにはテストがたくさん

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The overall scene is a classic representation of the Parisian skyline at night.

That's a problem?

それっていけないの？



Yes.

そう

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from traffic visible. The sky is a deep blue with some wispy clouds.

30 minutes to run tests on my Macbook

私のMacbook で30分もかかる

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, creating a warm, golden glow. The sky is a deep blue with some wispy clouds. The text "(It took the same on a Macbook Pro)" is overlaid in the center in a large, black, sans-serif font.

(It took the same on
a Macbook Pro)

(Macbook Proでもそう)

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The overall mood is serene yet vibrant.

What to do?

どうしよう

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The text 'TAP::Harness 3.0' is overlaid in the center in a large, black, sans-serif font.

TAP::Harness 3.0

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city's lights create a dense, glowing pattern across the landscape. The sky is a deep blue with some light clouds. The overall scene is a classic representation of the Parisian skyline at night.

I have two cores!

ポイントは2つ!

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city's lights create a dense, glowing pattern across the landscape. The sky is a deep blue with some light clouds. The text 'TAP::Harness::Parallel' is overlaid in the center of the image.

TAP::Harness::Parallel

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from traffic visible. The sky is a deep blue with some wispy clouds. The text "prove -j 5 --fork" is overlaid in the center of the image in a large, black, sans-serif font.

prove -j 5 --fork

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The text "22 minutes" is overlaid in large, bold, black letters.

22 minutes

prove -j 5で22分まで短くなった

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The word "Better!" is superimposed in the center of the image.

Better!

イイね!

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The overall mood is serene but carries a somber undertone due to the text overlay.

Not good.

まだただだけど

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, creating a warm, golden glow. The sky is a deep blue with some wispy clouds. The overall scene is a classic representation of the Parisian skyline at night.

Run tests on a server!

サーバでテストを実行できればいいのに!

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from traffic visible. The sky is a deep blue with some wispy clouds.

`rsync -rvp . server:`
`ssh server "prove -j 9 t"`



Better.

イイね

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The text "Annoying to run." is overlaid in the center in a large, black, sans-serif font.

Annoying to run.

実行するのが楽じゃないけど

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from traffic visible. The sky is a deep blue with some wispy clouds. The text 'TAP::Harness::Remote' is overlaid in the center in a large, black, sans-serif font.

TAP::Harness::Remote

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from moving vehicles. The sky is a deep blue with some wispy clouds.

Magic Remote Testing

魔法のリモートテスト



```
# ~/.remote_test
```

```
---
```

```
local: /path/to/local/testing/root/
```

```
host:
```

- remote1.testing.host.example.com
- remote2.testing.host.example.com

```
root: /where/to/place/local/root/on/remote/
```

```
user: username
```


```
master: 1
```

```
perl: /usr/bin/perl
```

```
ssh: /usr/bin/ssh
```

```
ssh_args:
```

- -X
- -S
- '~/.ssh/master-%r@%h:%p'

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some light trails from moving vehicles. The sky is a deep blue with some wispy clouds. The text 'prove --harness TAP::Harness::Remote t/*.*t' is overlaid in the center of the image.

prove --harness TAP::Harness::Remote t/*.*t

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The text "16.5 minutes" is overlaid in the center in a large, bold, black font.

16.5 minutes

16.5分!

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit with golden lights. The city's lights create a dense, glowing pattern across the landscape. Light trails from traffic are visible on major roads. The sky is a deep blue with some light clouds.

prove --harness TAP::Harness::Remote -j 5 t/*.*t

An aerial night photograph of Paris, France, featuring the Eiffel Tower prominently in the center, illuminated and standing out against the dark sky. The city's lights create a dense, glowing pattern across the landscape, with various buildings and streets visible. The sky is a deep blue, suggesting twilight or early night.

Magic Parallel Remote Testing

魔法の平行リモートテスト

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets. The sky is a deep blue with some light clouds. The text "12 minutes!" is overlaid in the center in a large, bold, black font.

12 minutes!

12分!

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, with some areas appearing blurred due to long exposure. The sky is a deep blue with some light clouds.

Not good enough
for TDD

まだまだテスト駆動開発する気にはなれないね

What I need is...

- Fast Servers (速いサーバ)
- Lots of Fast Servers (それもたくさん)
- Right Now (しかもいますぐ)
- Not Very Often (でも毎回要るわけじゃない)

ほしいのは

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city below is a dense grid of lights from buildings and streets, creating a warm, golden glow. The sky is a deep blue with some wispy clouds. The overall scene is a classic representation of a city at night.

A use for cloud computing!

クラウドコンピューティングをすればいい!



TAP::Harness::Remote::EC2



4 hosts. -j 5

An aerial night photograph of Paris, France. The Eiffel Tower is the central focus, brightly lit and standing out against the dark sky. The city's lights create a dense, glowing pattern across the landscape. Light trails from traffic are visible on major roads, and the overall atmosphere is one of a bustling metropolis at night.

89 seconds

89秒

An aerial night photograph of Paris, France, featuring the Eiffel Tower as the central landmark, illuminated and standing out against the dark sky. The city lights create a dense, glowing pattern across the landscape. Overlaid on the image is the large, bold Japanese text 'やった' (Yatta), which translates to 'Yay!' or 'We did it!', written in a black, stylized font. The text is positioned horizontally across the middle of the frame, with the Eiffel Tower visible through the central character.

やった

Yatta!

An aerial night photograph of Paris, France, featuring the Eiffel Tower illuminated in the center. Large, bold Japanese characters 'やった' (Yatta!) are superimposed over the image. The city lights and the dome of St. Stephen's Basilica are visible on the right.

やった

Yatta!



EC2 Caveats

Default Perl builds are really bad
デフォルトのPerlはほんとにひどい

Machines are mostly underpowered
マシンはたいてい非力

(“XL” instances are great, though)
「XL」のマシンはすごいけどね

Machine Setup/Tearardown is ~manual
マシンのセットアップとかは手動

EC2を使うときの注意

What's next for ::Harness::Remote::* ?

More polish on EC2 tools
EC2のツールを磨きあげる

Better EC2 instance control
EC2のインスタンス管理を手直しする

More widespread usage
インスタンスをもっと増やす

この先どうしよう？

Carp::REPL

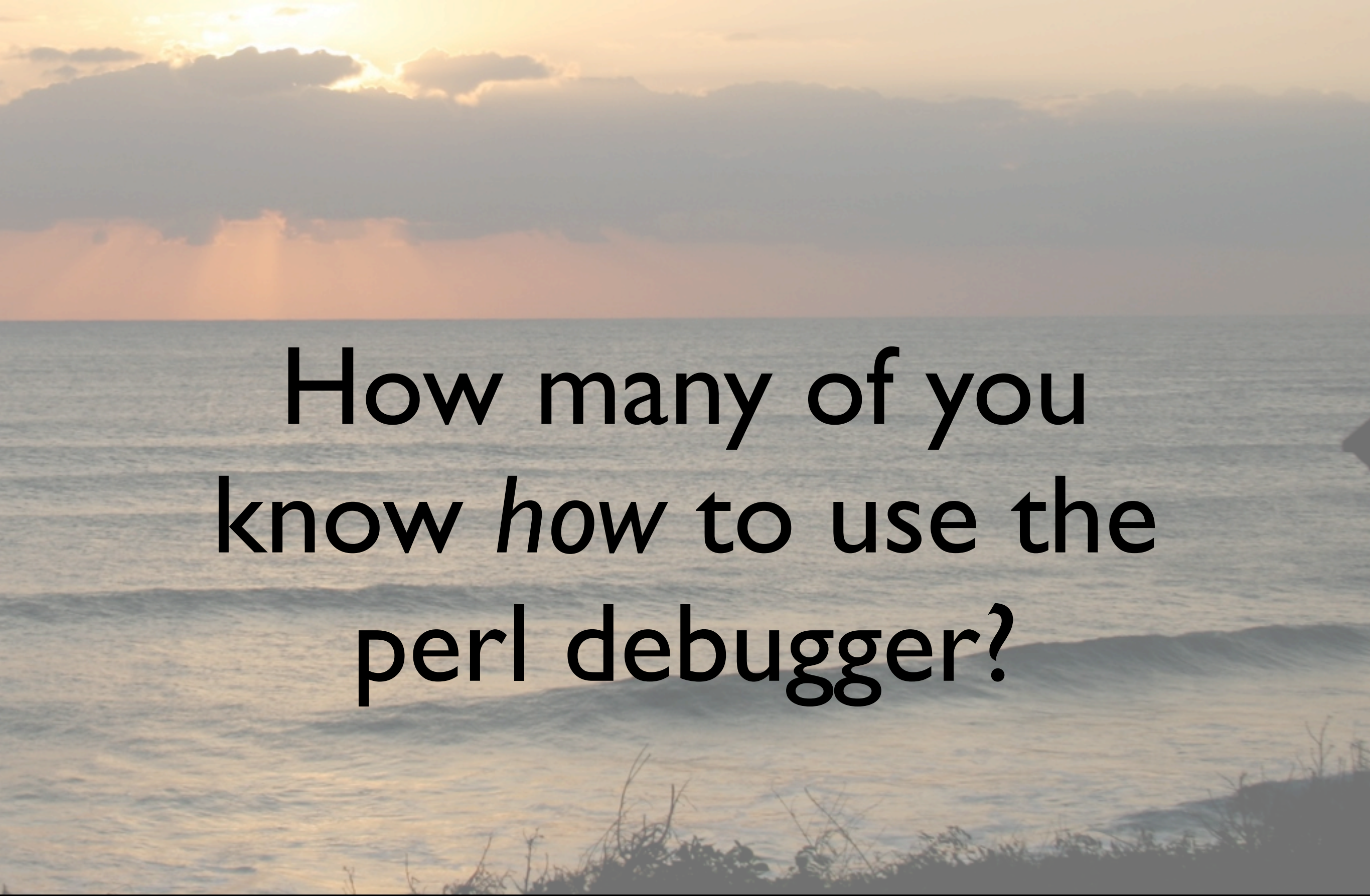
It's not quite a debugger

ただのデバッガじゃないよ




How many of you use
the Perl debugger?

Perlのデバッガを使ったことのある人は？



How many of you
know *how* to use the
perl debugger?

というか、使い方くらいはわかるという人は？

A photograph of a sunset over the ocean. The sun is low on the horizon, partially obscured by clouds, casting a warm orange glow. The ocean has gentle waves, and the foreground shows dark, silhouetted vegetation.

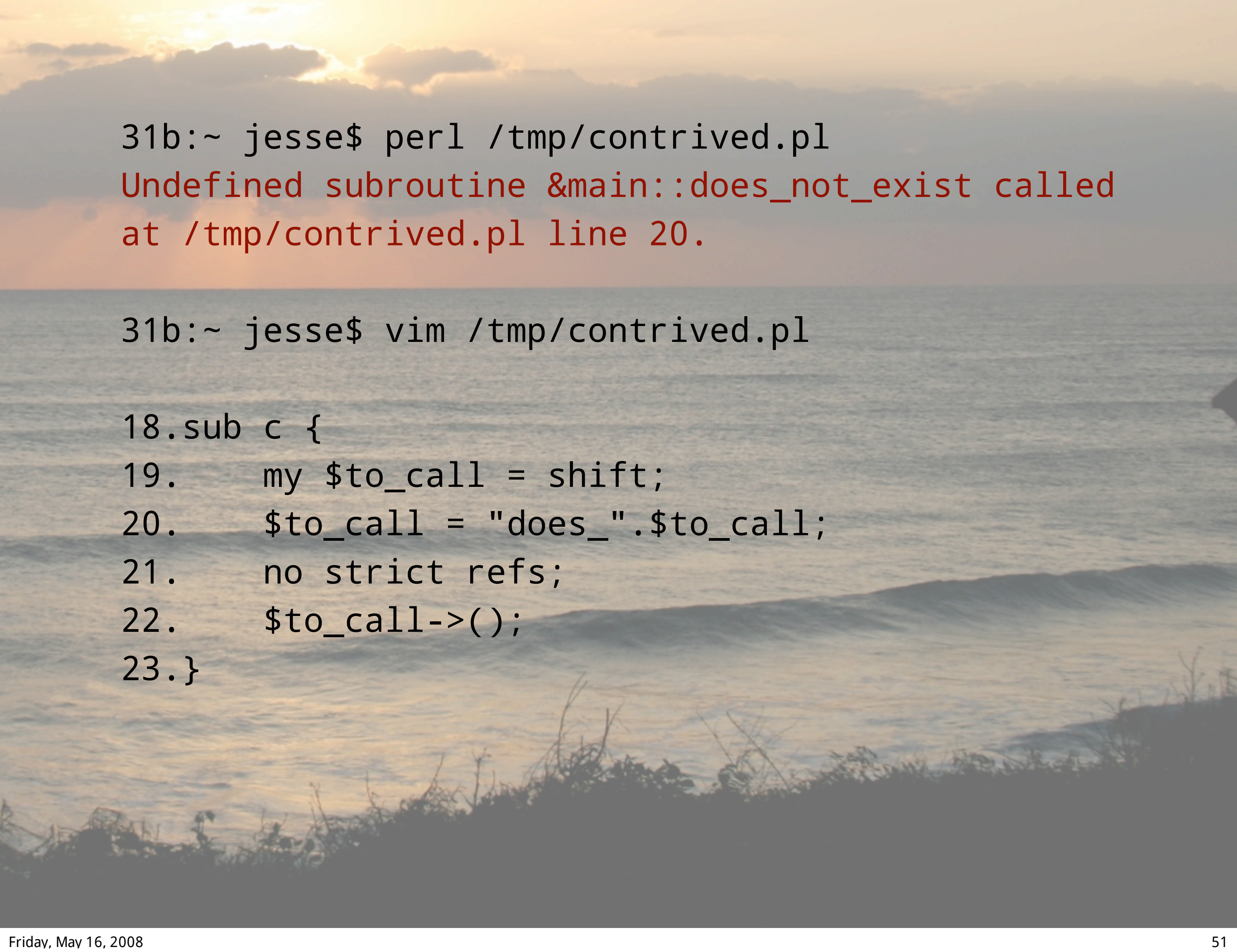
I don't, either.

私はどっちもノー

A photograph of a sunset over the ocean. The sun is low on the horizon, partially obscured by clouds, casting a warm orange glow. The ocean has gentle waves, and the foreground shows some dark, silhouetted vegetation.

So, what do I use?

じゃあ、どうすればいい？



```
31b:~ jesse$ perl /tmp/contrived.pl
```

```
Undefined subroutine &main::does_not_exist called  
at /tmp/contrived.pl line 20.
```

```
31b:~ jesse$ vim /tmp/contrived.pl
```

```
18.sub c {  
19.     my $to_call = shift;  
20.     $to_call = "does_".$to_call;  
21.     no strict refs;  
22.     $to_call->();  
23.}
```



```
31b:~ jesse$ perl /tmp/contrived.pl  
Undefined subroutine &main::does_not_exist called  
at /tmp/contrived.pl line 22.
```


```
31b:~ jesse$ vim /tmp/contrived.pl
```

```
18.sub c {  
19.     my $to_call = shift;  
20.     $to_call = "does_".$to_call;  
21.     no strict refs;  
22.     use Carp; Carp::cluck("$to_call");  
23.     $to_call->();  
24. }
```


A photograph of a sunset over the ocean. The sun is low on the horizon, casting a warm orange glow across the sky and water. There are some clouds in the sky. The ocean has gentle waves. In the foreground, there is a dark, silhouetted area that looks like grass or a cliff edge.

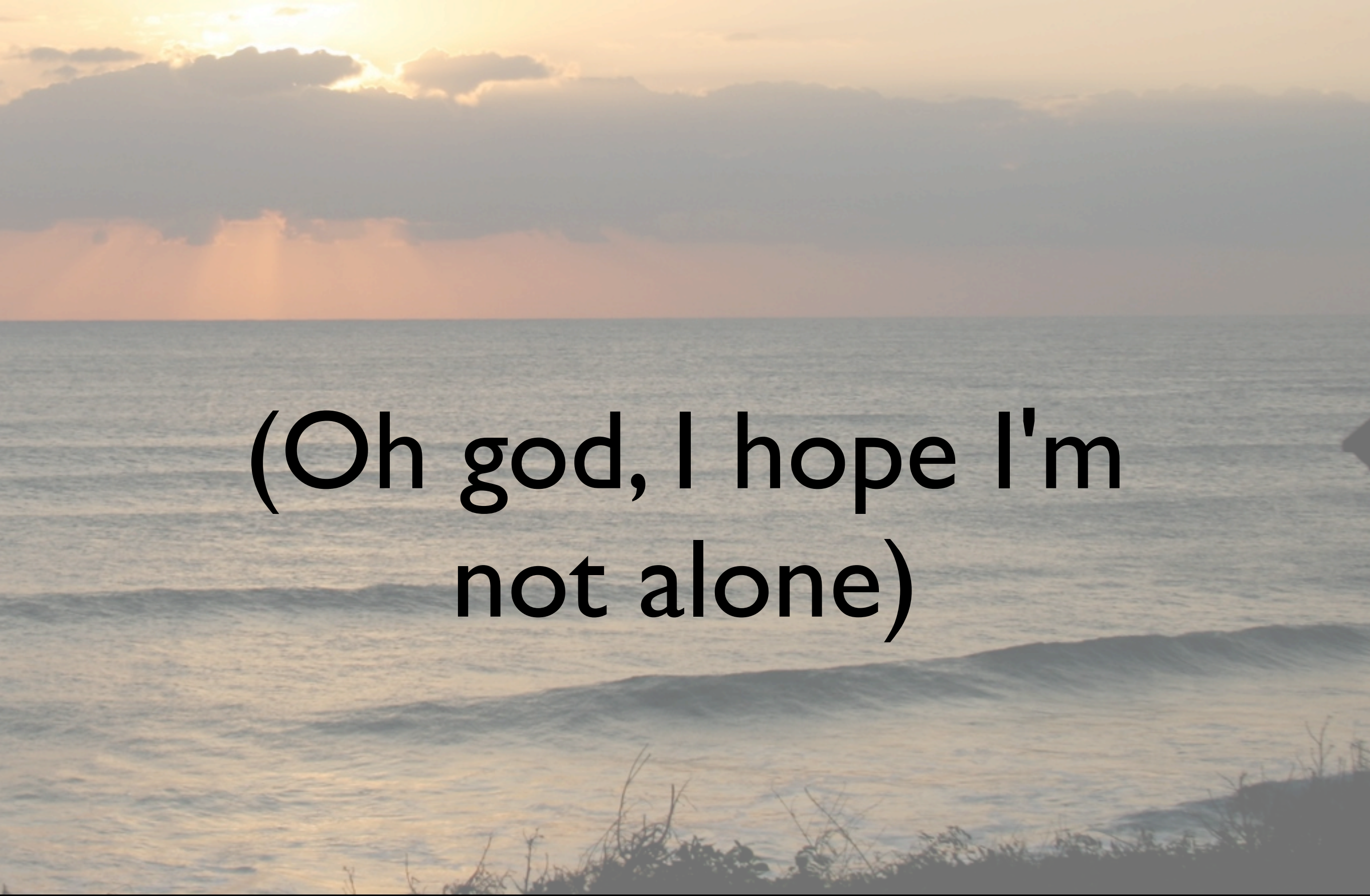
Ick. A Print Statement.

げ、Print文かよ




Anyone else here use
print statement
debugging?

printデバッギングしている人は？

A photograph of a sunset over the ocean. The sun is low on the horizon, casting a warm orange glow across the sky and the water. The sky is filled with soft, white clouds. The ocean has gentle waves rolling in. In the foreground, there is a dark, silhouetted area that appears to be a grassy or rocky shore.


(Oh god, I hope I'm
not alone)

私だけじゃないといいんだけど

A photograph of a sunset over the ocean. The sun is low on the horizon, casting a warm orange glow across the sky and water. The sky is filled with soft, white clouds. The ocean has gentle waves rolling in. In the foreground, there is a dark, silhouetted shoreline with some sparse vegetation.


Acme, you wrote
Devel::ebug...

Leon BrocardがDevel::ebugというのを書いてくれたけど...

A photograph of a sunset over the ocean. The sun is low on the horizon, partially obscured by clouds, casting a warm orange glow. The ocean has gentle waves, and the foreground shows some dark, silhouetted vegetation.

Do you use it?

使ってる？

A background image showing a sunset over the ocean. The sun is low on the horizon, casting a warm, orange glow across the sky and water. The foreground is dark and silhouetted, showing some vegetation or rocks. The text is overlaid in the center of the image.

I want a simple
debugger that's *easier*
than print statements

ほしいのはprint文より簡単なデバッガなんです



sartak created Carp::REPL

Shawn MooreはCarp::REPLを書ってくれた

A sunset over the ocean with waves and a dark foreground.

A lightweight, modern, interactive debugger

軽くてモダンで対話的なデバッガ

A sunset over the ocean with waves and a dark foreground.

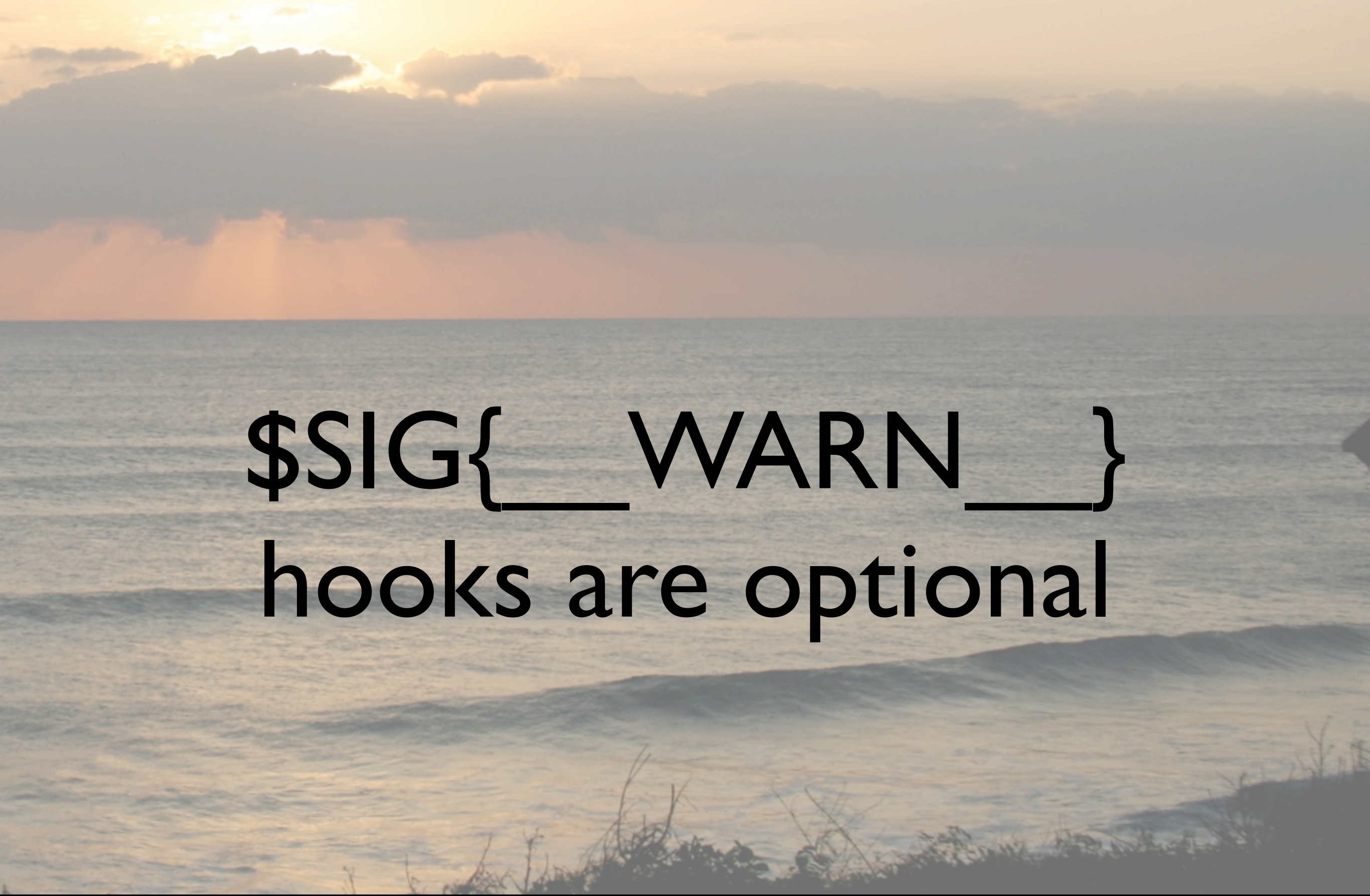
Built on Devel::REPL

Devel::REPLの上につくられている

A background image of a sunset over the ocean. The sun is low on the horizon, casting a warm orange glow across the sky and water. The sky is filled with soft, white clouds. The ocean is a deep blue-grey color, with gentle waves visible. In the foreground, there are some dark, silhouetted plants or grasses.

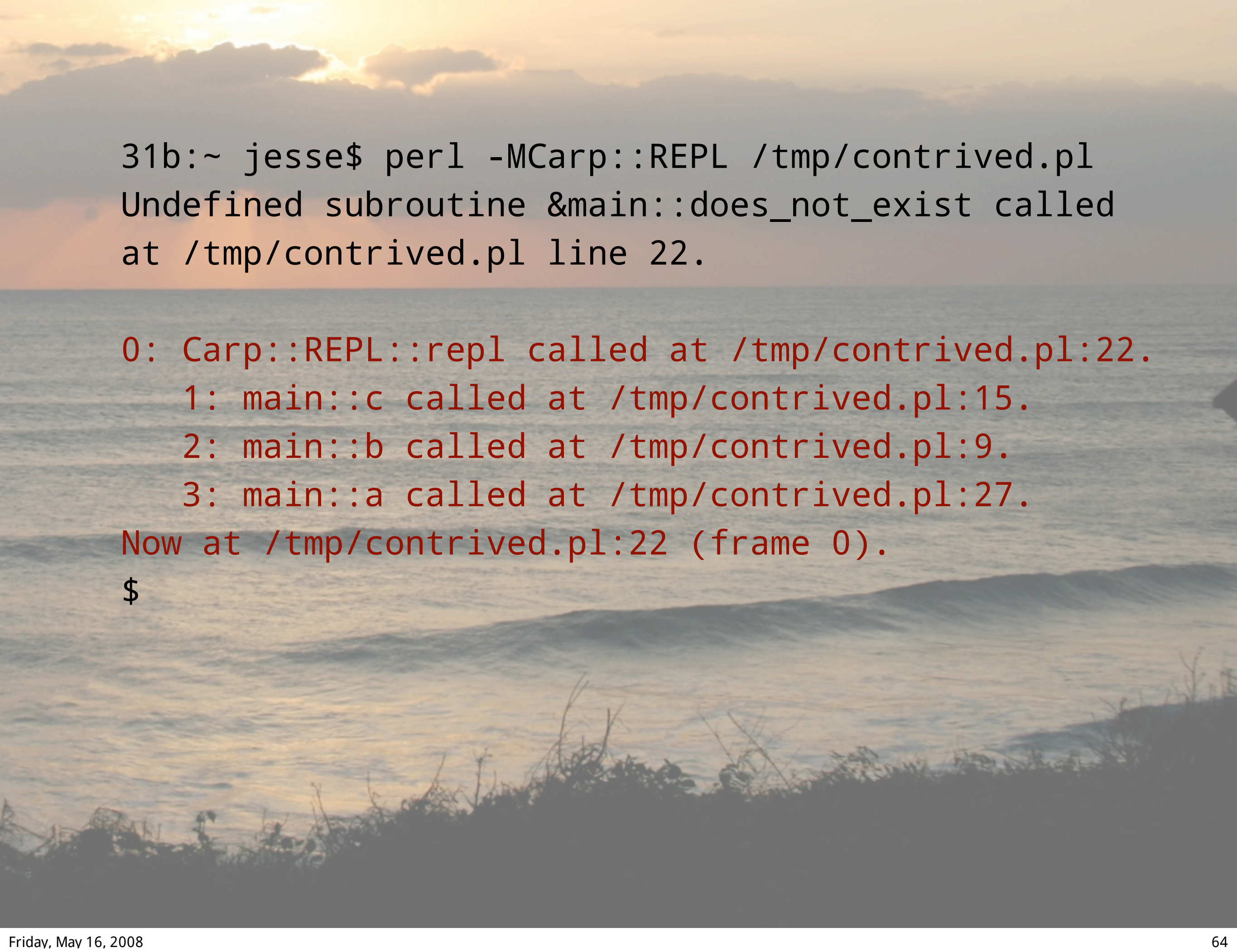
Carp::REPL Hooks into \$SIG{__DIE__}

Carp::REPLは\$SIG{__DIE__}をフックしている

A background image of a sunset over the ocean. The sun is low on the horizon, casting a warm orange glow across the sky and water. The sky is filled with soft, white clouds. The ocean is calm with gentle waves. In the foreground, there are some dark, silhouetted plants or grasses.

\$SIG{__WARN__}
hooks are optional

\$SIG{__WARN__}の方はお好みで



```
31b:~ jesse$ perl -MCarp::REPL /tmp/contrived.pl  
Undefined subroutine &main::does_not_exist called  
at /tmp/contrived.pl line 22.
```

```
0: Carp::REPL::repl called at /tmp/contrived.pl:22.  
1: main::c called at /tmp/contrived.pl:15.  
2: main::b called at /tmp/contrived.pl:9.  
3: main::a called at /tmp/contrived.pl:27.  
Now at /tmp/contrived.pl:22 (frame 0).  
$
```


A photograph of a sunset over the ocean. The sun is low on the horizon, casting a warm orange glow across the sky and water. The sky is filled with soft, white clouds. The ocean has gentle waves rolling in. In the foreground, there is a dark, silhouetted area that appears to be a grassy or rocky shore.

That's not a shell
prompt

シェルのプロンプトじゃないよ

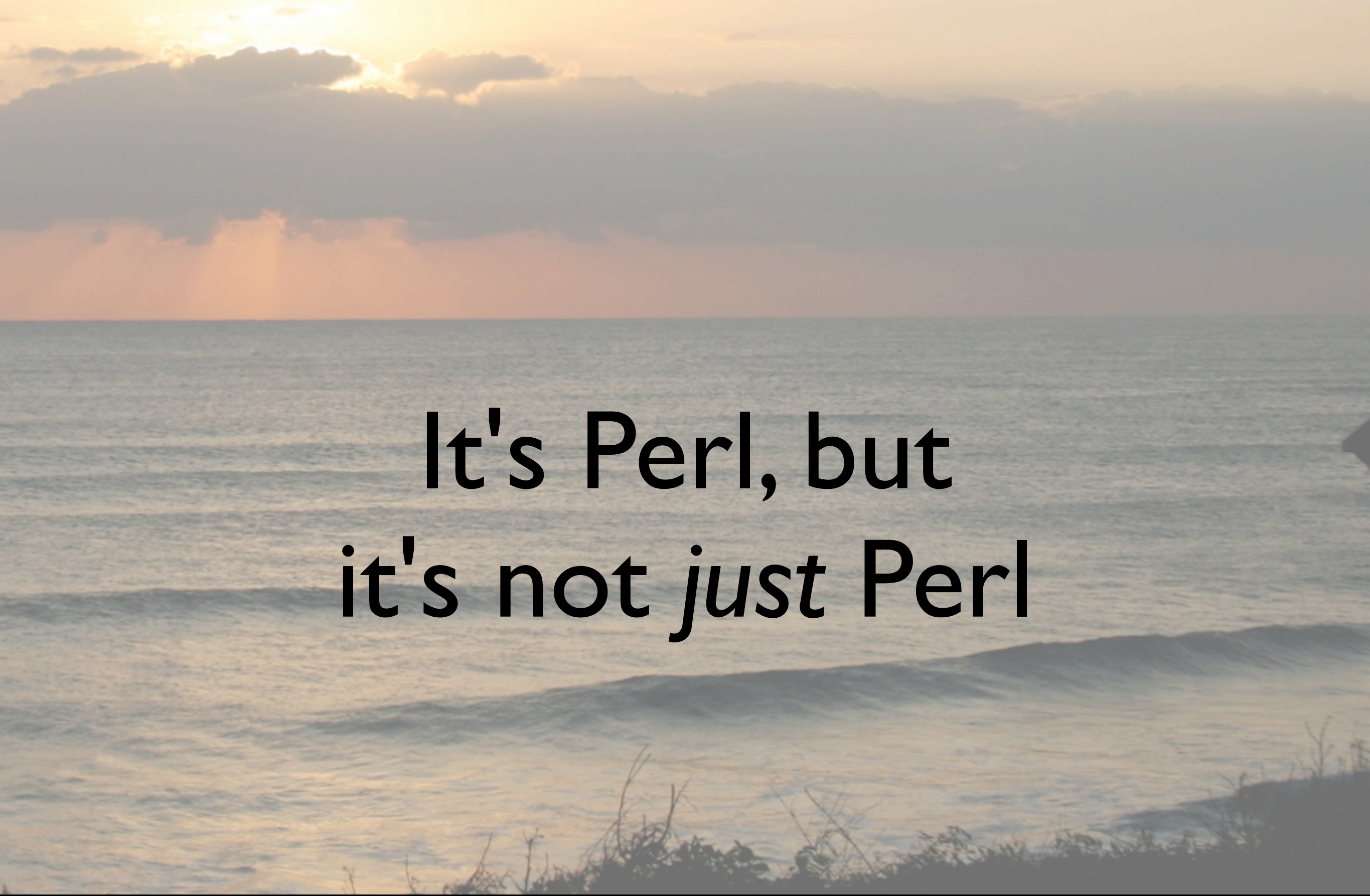

```
31b:~ jesse$ perl -MCarp::REPL /tmp/contrived.pl  
Undefined subroutine &main::does_not_exist called  
at /tmp/contrived.pl line 22.
```

```
0: Carp::REPL::repl called at /tmp/contrived.pl:22.  
  1: main::c called at /tmp/contrived.pl:15.  
  2: main::b called at /tmp/contrived.pl:9.  
  3: main::a called at /tmp/contrived.pl:27.
```

```
Now at /tmp/contrived.pl:22 (frame 0).
```


```
$ warn $0
```

```
/tmp/contrived.pl at (eval 135) line 15, <FIN> line 1.
```


It's Perl, but
it's not *just* Perl

Perlではあるんだけど、ただのPerlじゃあない



`:t` - Stack trace
`:u` - Up one stack frame
`:d` - Down one stack frame
`:e` - Print this frame's lexical environment
`:q` - Quit this REPL

`$_e` - ARRAYREF of this frame's `@_`



\$:t

0: Carp::REPL::repl called at /tmp/contrived.pl:22.

1: main::c called at /tmp/contrived.pl:15.

2: main::b called at /tmp/contrived.pl:9.

3: main::a called at /tmp/contrived.pl:27.

\$


```
$ :e
$Env = { "\$to_call" => \do { my $v = 'does_not_exist' } };
$ :u
$ :u
Now at /tmp/contrived.pl:9 (frame 2).
$ :e
$Env = { "\$to_call" => \do { my $v = 'exist' } };
$ :u
Now at /tmp/contrived.pl:27 (frame 3).
$ :e
$Env = { "\$to_call" => \do { my $v = '' } };
$ :d
Now at /tmp/contrived.pl:9 (frame 2).
$ :e
$Env = { "\$to_call" => \do { my $v = 'exist' } };
$ warn join(", ", @$_);
exist at (eval 138) line 18, <FIN> line 4.
```


Why is my code 'warn'ing?

Maybe it's in an eval.
evalの中だからかな？

Or in a CPAN library.
CPANモジュールのせいかも

No problem!
まあいいや！

```
perl -MCarp::REPL=warn
```

どうして警告が出るんだろう？


```
31b:~ jesse$ perl -MCarp::REPL=warn /tmp/contrived.pl
Use of uninitialized value in concatenation (.) or string at /tmp/contrived.pl
line 8.
```

```
0: Carp::REPL::repl called at /tmp/contrived.pl:8.
```

```
1: main::a called at /tmp/contrived.pl:27.
```

```
Now at /tmp/contrived.pl:8 (frame 0).
```

```
$ :e
```

```
$Env = { "\$to_call" => \do { my $v = undef } };
```

```
$ :q
```

```
Undefined subroutine &main::does_not_exist called at /tmp/contrived.pl line 22,
<FIN> line 2.
```

```
0: Carp::REPL::repl called at /tmp/contrived.pl:22.
```

```
1: main::c called at /tmp/contrived.pl:15.
```

```
2: main::b called at /tmp/contrived.pl:9.
```

```
3: main::a called at /tmp/contrived.pl:27.
```

```
Now at /tmp/contrived.pl:22 (frame 0).
```

```
$
```


Inserting breakpoints

You know your 'print' statements?

「print」文をどこに入れたか覚えてる？

Turn them into breakpoints.

こういうのはブレークポイントにしてしまおう


Two ways:

2通りのやり方があるよ:

```
s/print/warn/ and -MCarp::REPL=warn
```

```
Carp::REPL::repl
```

ブレークポイントを挿入する



```
18.sub c {  
19.    my $to_call = shift;  
20.    no strict 'refs';  
21.    Carp::REPL::repl(); #Gimme a breakpoint here!  
22.    $to_call = "does_".$to_call;  
23.    $to_call->();  
24.}
```


What's next for Carp::REPL?

I have no idea.
思いつかない

It's simple.
シンプルだし

It works.
動いてるしね

この先どうしよう？

The background of the slide is a complex, abstract pattern of glowing lines. These lines are primarily yellow and white, with some segments in red and blue. They appear to be interconnected, forming a dense, web-like structure that fills the entire frame. The lines have a soft, ethereal glow, giving the impression of light trails or a complex network of connections.

Build web apps



Template::Declare

A Pure Perl Templating Engine

Template::Declare

A Pure Perl Templating Engine

2007 Talk
Encore
Presentation!



Perl has two big templating camps

Perlのテンプレートエンジンには2つの流派がある

Template::Toolkit

A DSL for templates
テンプレート用のDSL

It's NOT Perl
ではない

Why should I learn a new language for
templates?
なんでテンプレートのために新しい言語

HTML::Mason

There's Perl inside.
こちらはPerlを埋め込んでる

HTML::Mason

<mason><is><still>

<full><of>

<angle><brackets>

<でも><まだ><ブラケット><が><たくさん><ある>

HTML::Mason

</brackets></angle>

</of></full>

</still></is></mason>

</ある></たくさん></が></ブラケット></まだ></でも>



I'm Just A Simple Perl Hacker

私はただのPerlハッカーだから




I needed something
more Perlsh

もっとPerlっぽいのがほしかった



```
template '/pages/mypage.html' => sub {  
  html {  
    head {};  
    body {  
      h1 {'Hey, this is text'};  
    }  
  }  
};
```

```
template choices => page {  
  h1 { 'My Choices' }  
  dl {  
    my $choices = Doxory::Model::ChoiceCollection->new;  
    $choices->limit(  
      column => 'asked_by',  
      value   => Jifty->web->current_user->id,  
    );  
    while (my $c = $choices->next) {  
      dt { $c->name, ' (' , $c->asked_by->name, ' )' }  
      dd { b { $c->a } em { 'vs' } b { $c->b } }  
    }  
  }  
};
```




But!

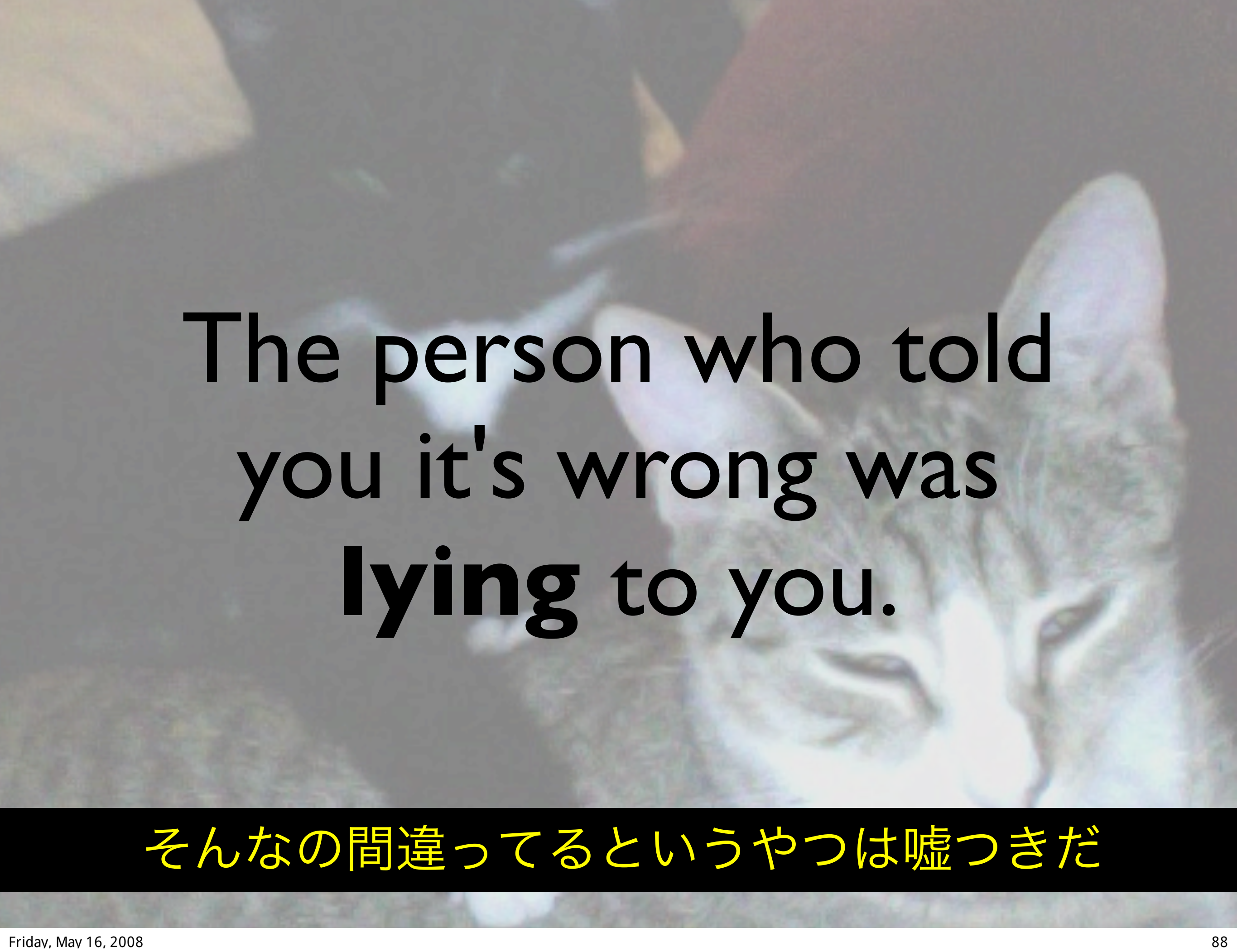
Content! Templates!
コンテンツ! テンプレート!

Design! Code!
デザイン! コード!

OMGWTFBBQ!?
やってられるか!?

THAT'S WRONG!
こんなの間違ってる!

でも




The person who told
you it's wrong was
lying to you.

そんなの間違ってるというやつは嘘つきだ



We're Perl hackers

私たちはPerlハッカーなんだから



Why are we writing
our templates in
another language?

なんで別の言語でテンプレートを書いているのさ



This is not 1997

1997年じゃないんだよ



It's 2008

2008年なんだから



People use CSS
for design now

デザインにはCSSを使うものでしょ



Programmers still have
to make templates

テンプレートはプログラマがつくらないといけないけどね



Templates run like CODE

テンプレートはコードのように実行できる



Because they ARE code

だってコードなんだから



Let's use our CODING
tools to work with
them

テンプレートの作業にもコーディング用のツールを使おうよ



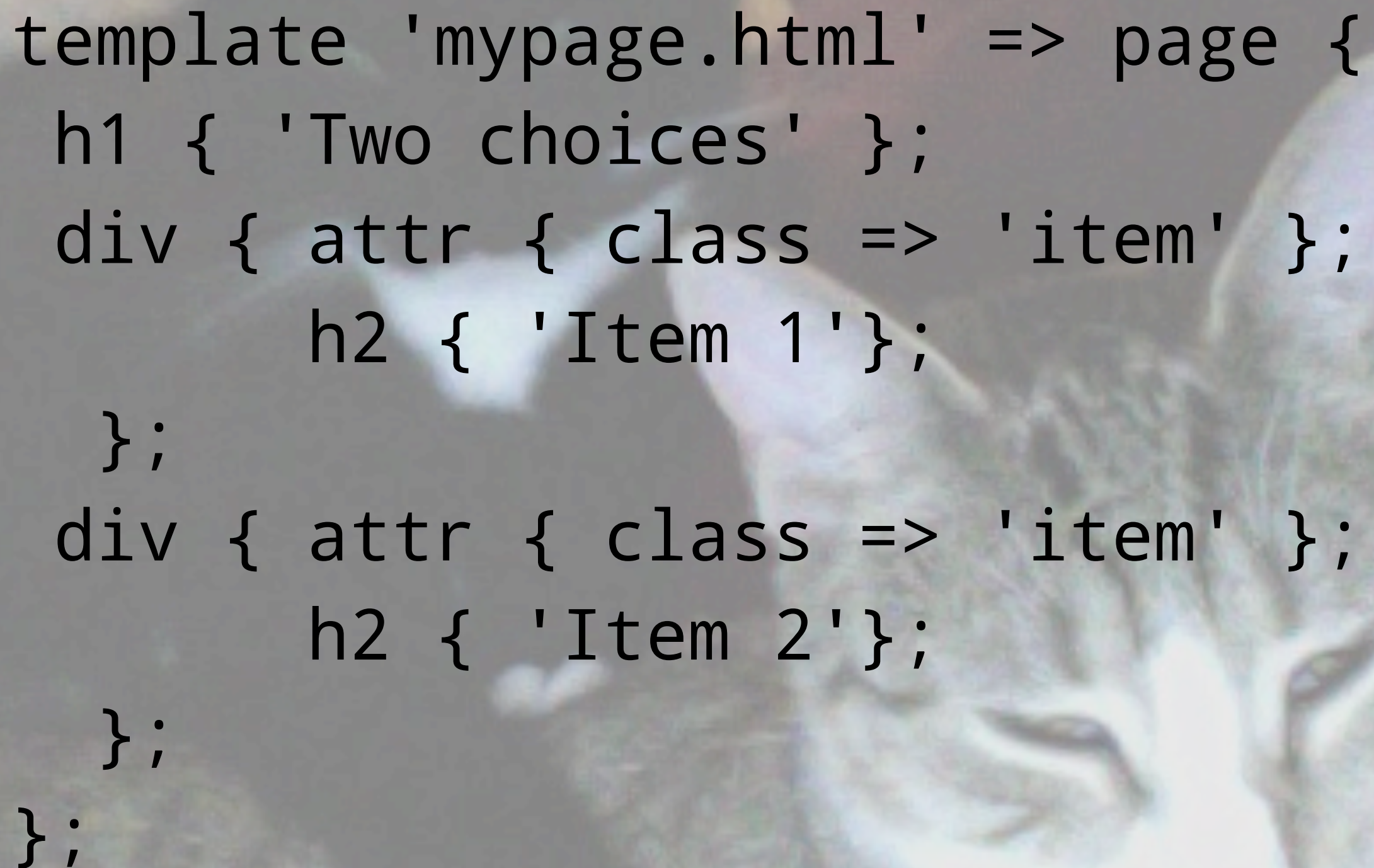
How about Refactoring?

リファクタリングはどうするのって？

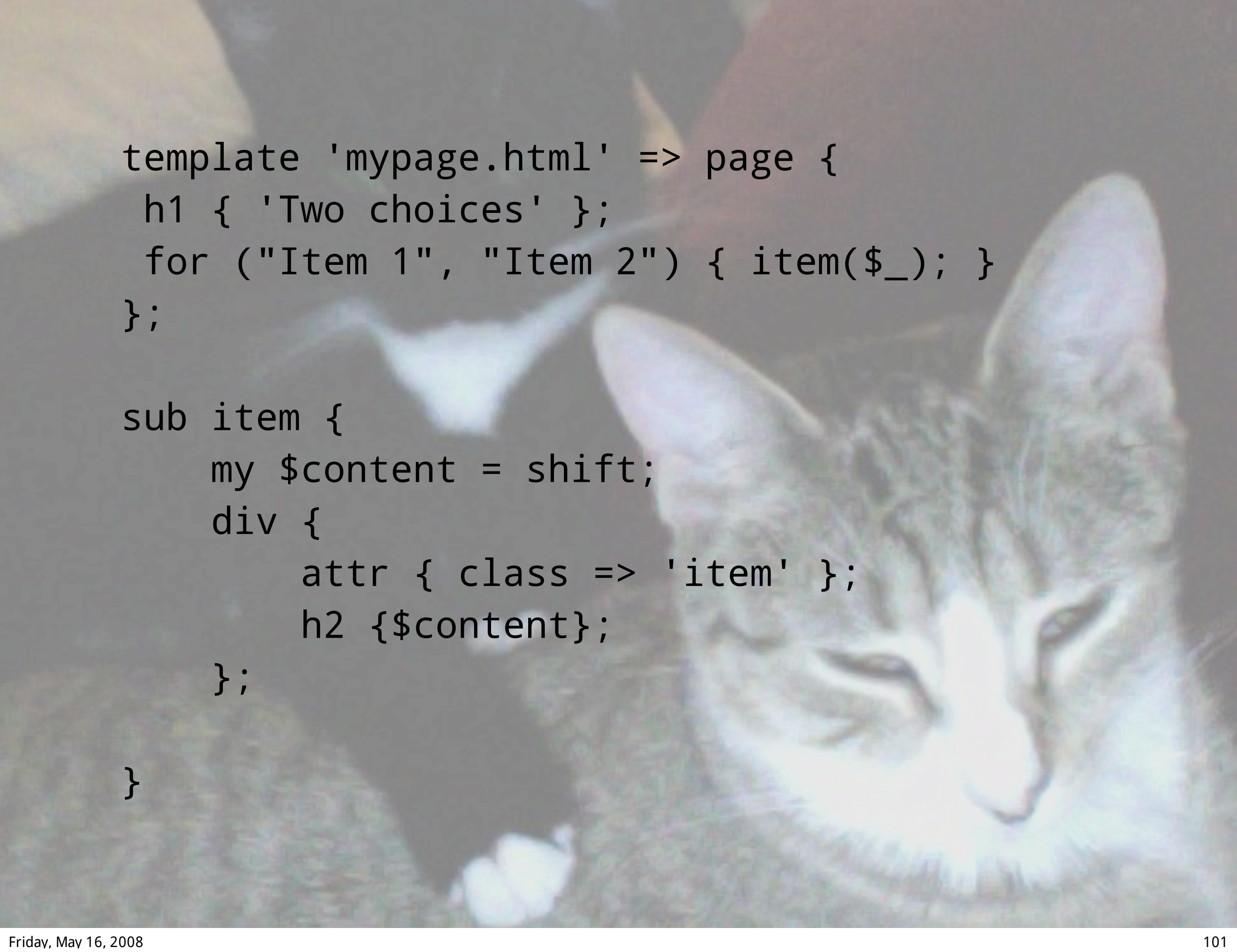


Have you ever tried to
refactor HTML?

HTMLのリファクタなんてしたことあります？



```
template 'mypage.html' => page {  
  h1 { 'Two choices' };  
  div { attr { class => 'item' };  
    h2 { 'Item 1' };  
  };  
  div { attr { class => 'item' };  
    h2 { 'Item 2' };  
  };  
};
```

```
template 'mypage.html' => page {  
  h1 { 'Two choices' };  
  for ("Item 1", "Item 2") { item($_); }  
};  
  
sub item {  
  my $content = shift;  
  div {  
    attr { class => 'item' };  
    h2 {$content};  
  };  
}
```




We can refactor
templates!

私たちはできるんですよ!



Our HTML is
magically valid

私たちのHTMLには魔法がかかっていますから



(Syntax errors are...
Syntax Errors)

文法に間違いがあったらシンタックスエラーになります...



Object Oriented Templates

オブジェクト指向テンプレート

Subclassing

It just works
まったく問題なし

We're using Perl's symbol table
Perlのシンボルテーブルを使っていますから

(Template::Declare's show() passes the class for you)
Template::Declareのshow()がクラスを渡してくれます

サブクラス化

Mixins

`alias Some::Template::Library under '/path';`

That's it.
それだけ



Closures

クロージャ

Tags as Closures

HTML tags take blocks of content

HTMLのタグはコンテンツのブロックをとる

Our tag methods take blocks of Perl

私たちのタグメソッドはPerlのブロックをとって

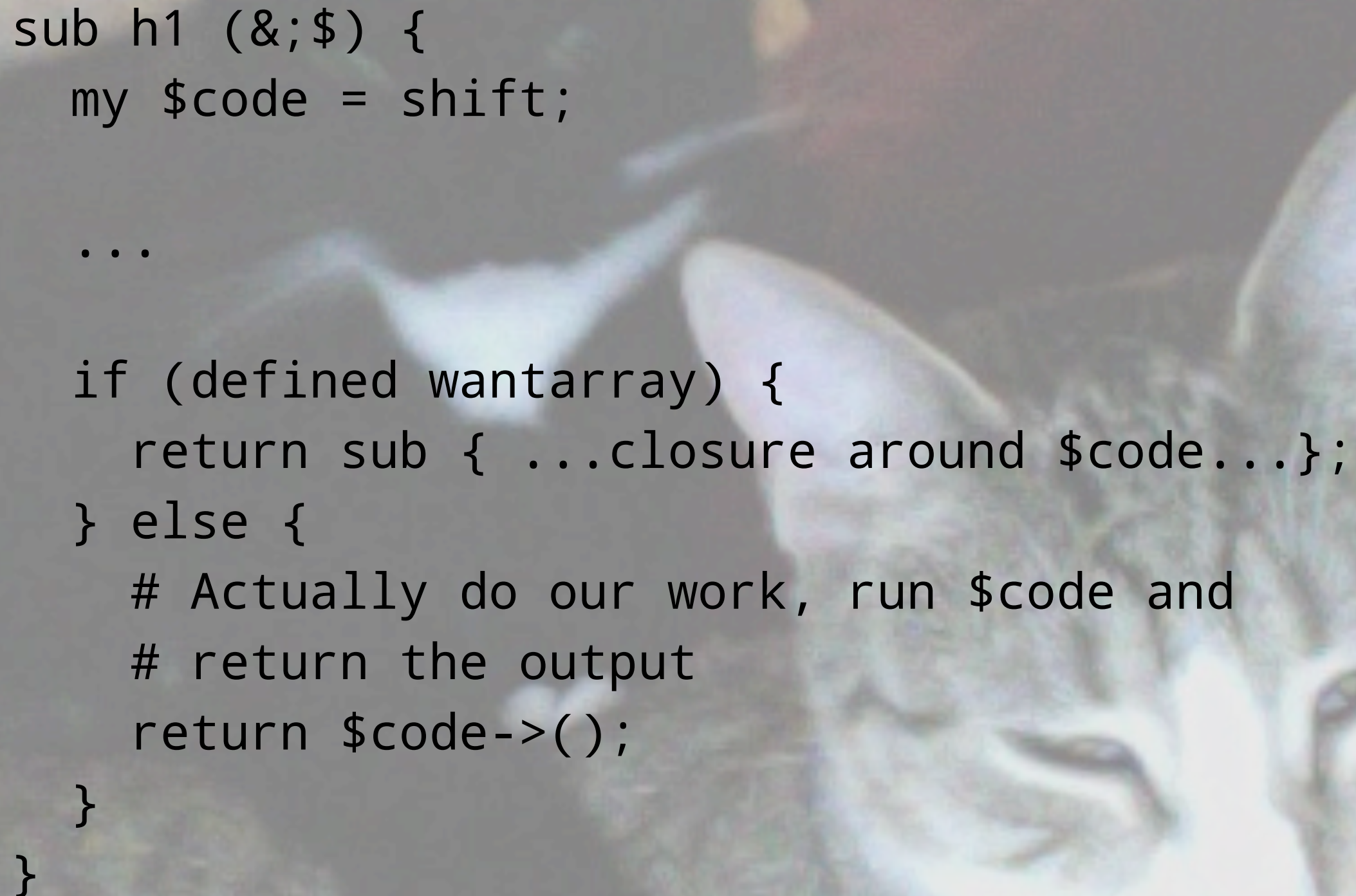
They return closures when you want them

必要なときにクロージャを返して

They run and output their content when you want them to

必要なときに実行して中身を出力

タグはクロージャで



```
sub h1 (&,$) {  
  my $code = shift;  
  
  ...  
  
  if (defined wantarray) {  
    return sub { ...closure around $code...};  
  } else {  
    # Actually do our work, run $code and  
    # return the output  
    return $code->();  
  }  
}
```


What's next for Template::Declare?

HTML Attribute Validation

HTML属性のバリデーション

Compile to .js

jsへのコンパイル

この先どうしよう？

CSS::Squish

A compiler for Cascading Style Sheets

CSSのコンパイラ

The Problem

Many small CSS Files

CSSのファイルは小さいものがたくさん

Easy to work with

作業するのは簡単だけど

Painful to serve

サーブするのは大変

問題

The Problem

One big CSS file

ひとまとめにした大きなCSSファイル

Painful to work with
作業は大変だけど

Easy to serve
サーブするのは簡単

問題

How CSS works

main.css:

```
@import "yui-fonts.css";
```

```
@import "base.css";
```

```
@import "layout.css";
```

```
@import "nav.css";
```

```
@import "forms.css";
```

```
@import "boxes.css";
```

```
@import "login.css";
```

...

CSSの動作

How CSS works

- Browser downloads CSS stylesheets listed in HTML
- Browser processes each stylesheet for @include directives
- Browser fetches each stylesheet found
- This is *slow*.

CSS::Squishの使い方

How CSS works

To make CSS faster, you can tell browsers to cache stylesheets
ブラウザにスタイルシートをキャッシュさせればCSSの処理は速くなる

...then you have trouble if you want to update your styles
そうするとスタイルを更新したくなくなったときに困ることになる

Some people add timestamps to the stylesheet names:
スタイルシートの名前に日時を加える人もいる

/css/mypagestyle.css?1210362379

CSS::Squishの使い方

How CSS::Squish works

It pretends to be a browser.
ブラウザのふりをする

You give it the path to a stylesheet.
スタイルシートのパスを渡すと

It returns the fully computed stylesheet the browser would get eventually.

ブラウザが最終的に受け取るような形に整形したスタイルシートを返すので

You serve that file out to the user.
そのファイルをユーザにサーブする

CSS::Squishの動作

How we use CSS::Squish

Instead of: こうするかわりに

```
<link rel="stylesheet" type="text/css"
href="main.css"/>
```

We have this: 私たちはこうしている

```
<link rel="stylesheet" type="text/css" href="/
css/squished/
4a50a85044a6ba727f7aa0683ac21f7e.css"/>
```

CSS::Squishの使い方

How we use CSS::Squish

If we have a cached copy of the squished CSS, name the 'file' as an MD5 of the content

圧縮したCSSのコピーがキャッシュにあったら内容のMD5を
ファイル名にして

Tell browsers to cache it forever

ブラウザにはずっとキャッシュしておくようにすると

When users load a page, they will always fetch the squished CSS if it's changed. By magic

ユーザがページをロードするときに内容が変わっていても、
あら不思議、いつでも圧縮したCSSをとってこられる

CSS::Squishの使い方

What's next for CSS::Squish

Refactor the MD5 hash bits from Jifty to
CSS::Squish

MD5ハッシュの部分はJiftyではなく
CSS::Squishできるようにリファクタ

What else do you want?
あとはどうしよう？

この先どうしよう？



Ship software

ソフトのリリース



App::ChangeLogger

Gorgeous changelogs

華麗な履歴



We create a lot of
software

私たちはソフトをたくさんつくっている



(145 Projects in
bps-public svn)

公開リポジトリにあるプロジェクトは145個



We're not so good at
doing *releases*

リリースはあまり得意じゃない



Lots of manual work

手作業が多い




Shiplt can help with
most of it

Shipltを使えばだいぶ楽になるけど



What about Changelog?

更新履歴はどうしよう？

A background image showing two light-colored monkeys, possibly Japanese macaques, behind heavy metal chains. The monkey on the left is partially visible, with its head and red face in focus. The monkey on the right is more prominent, sitting and looking towards the left. The chains are thick and dark, creating a sense of confinement. The background is a blurred outdoor setting with some greenery and a body of water.

```
svn log svn://svn.foo.com/MyProj  
> Changelog
```


A photograph of two Japanese macaques (snow monkeys) behind a metal chain. The monkey on the left is partially visible, with its head and hands near the chain. The monkey on the right is more prominent, looking towards the left. The background is a blurred natural setting.

Ship it!

これをShipltすると



Ew. No. Ick.
Wrong. Bad.

うーん。だめ。いや。間違ってる



App::ChangeLogger

Two commands

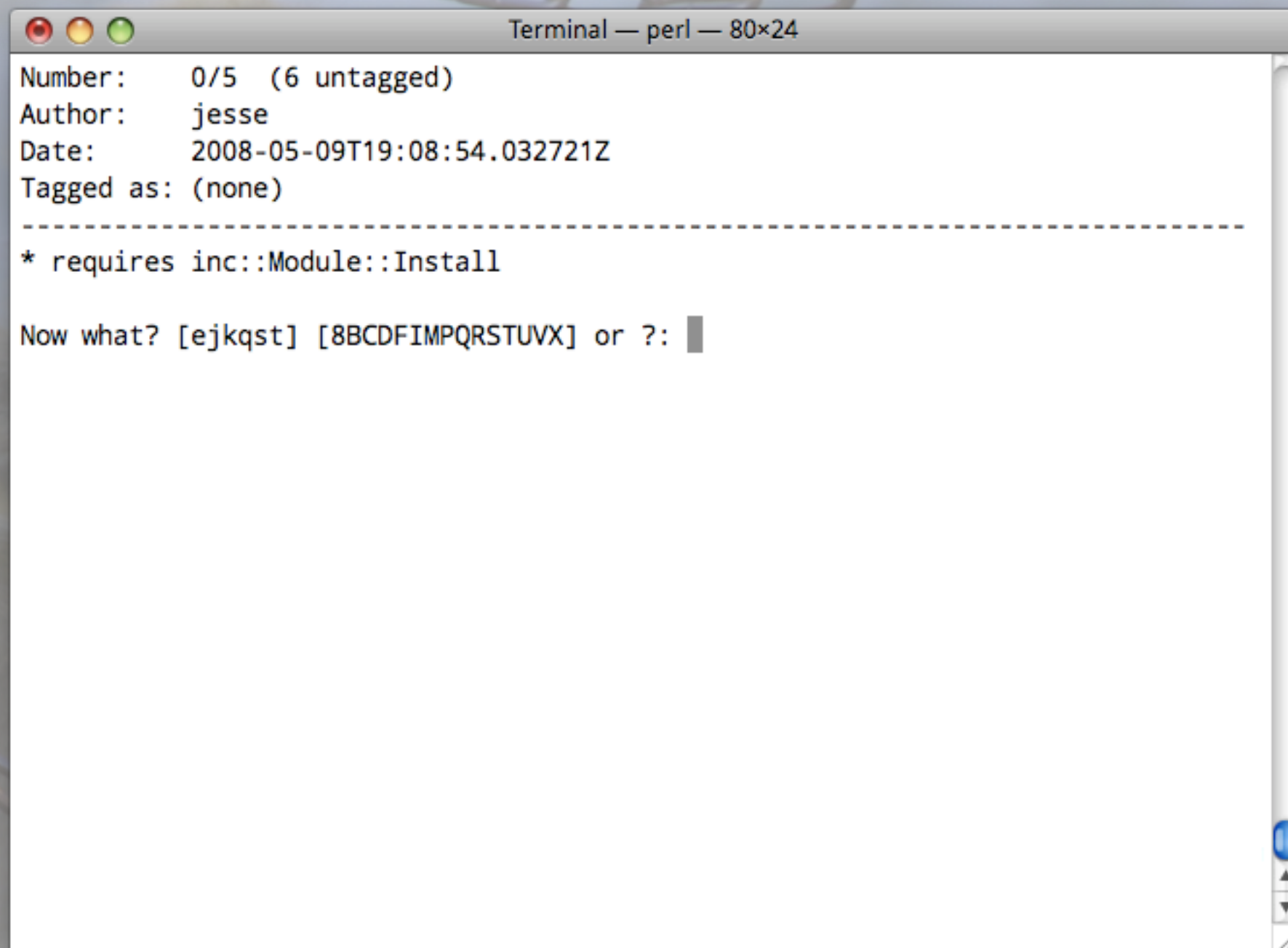
- sort-changelog
- generate-changelog

コマンドは2つ

Ok, three commands:

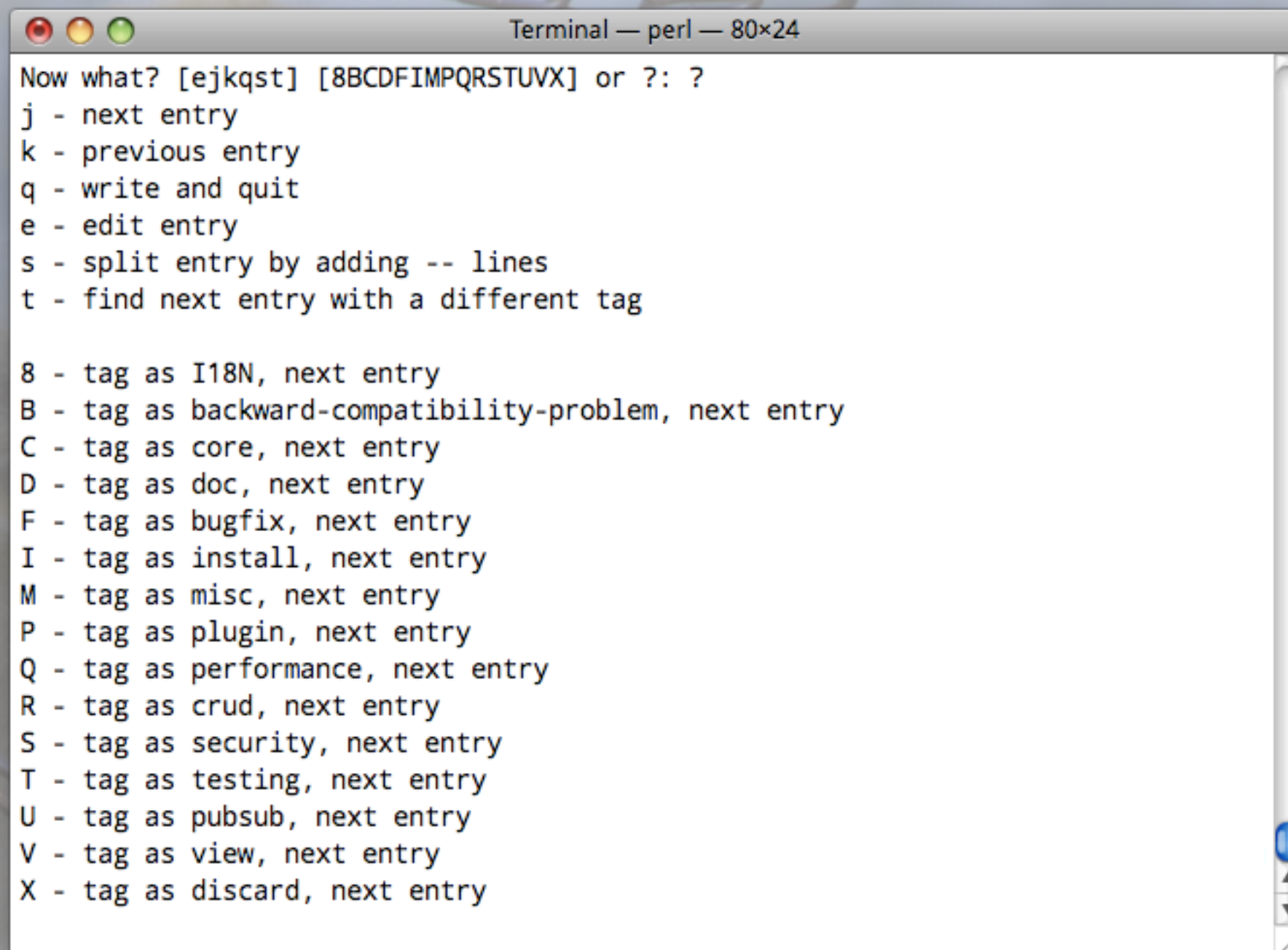
- `svn log --xml svn://svn.bestpractical.com/App-Changelogger > changelog.raw.xml`
- `./bin/sort-changelog changelog.xml
changelog.out.xml`
- `bin/generate-changelog --generate
changelog.out.xml Changelog`

いや、3つ



```
Terminal — perl — 80x24
Number:    0/5  (6 untagged)
Author:    jesse
Date:      2008-05-09T19:08:54.032721Z
Tagged as: (none)
-----
* requires inc::Module::Install

Now what? [ejkqst] [8BCDFIMPQRSTUVX] or ?:
```

```
Terminal — perl — 80x24
Now what? [ejkqst] [8BCDFIMPQRSTUVX] or ?: ?
j - next entry
k - previous entry
q - write and quit
e - edit entry
s - split entry by adding -- lines
t - find next entry with a different tag

8 - tag as I18N, next entry
B - tag as backward-compatibility-problem, next entry
C - tag as core, next entry
D - tag as doc, next entry
F - tag as bugfix, next entry
I - tag as install, next entry
M - tag as misc, next entry
P - tag as plugin, next entry
Q - tag as performance, next entry
R - tag as crud, next entry
S - tag as security, next entry
T - tag as testing, next entry
U - tag as pubsub, next entry
V - tag as view, next entry
X - tag as discard, next entry
```



```
Terminal — bash — 80x24
31b:App-Changelogger jesse$ cat Changelog
BUGFIX
=====

* Add a way to split commits for those pesky mega-commits - sartak

CORE
=====

* Factor out invoking the user's editor, usage line fix - sartak
* Initial import. Will work on this more after Jifty comes out - sartak

DISCARD
=====

* requires inc::Module::Install - jesse
* App::Changelogger, moved out of Jifty - sartak

DOC
===

* Sketch out the module hierarchy - sartak
```


What's next for App::Changelogger?

Use it on itself. (Release it)

隗より始めよ（まずは自分をリリースしないと）

Store changelog metadata in repository

更新履歴のメタデータをリポジトリに

Support for Git, Hg

GitやHgのサポート

Easier customization

カスタマイズをもっと簡単に

この先どうしよう？



Shipwright

Build. Ship.

ビルド、リリース

The Problem

CAPAN

問題

The Problem

Everything that
is not CPAN

問題: CPAN以外のすべて



We make RT

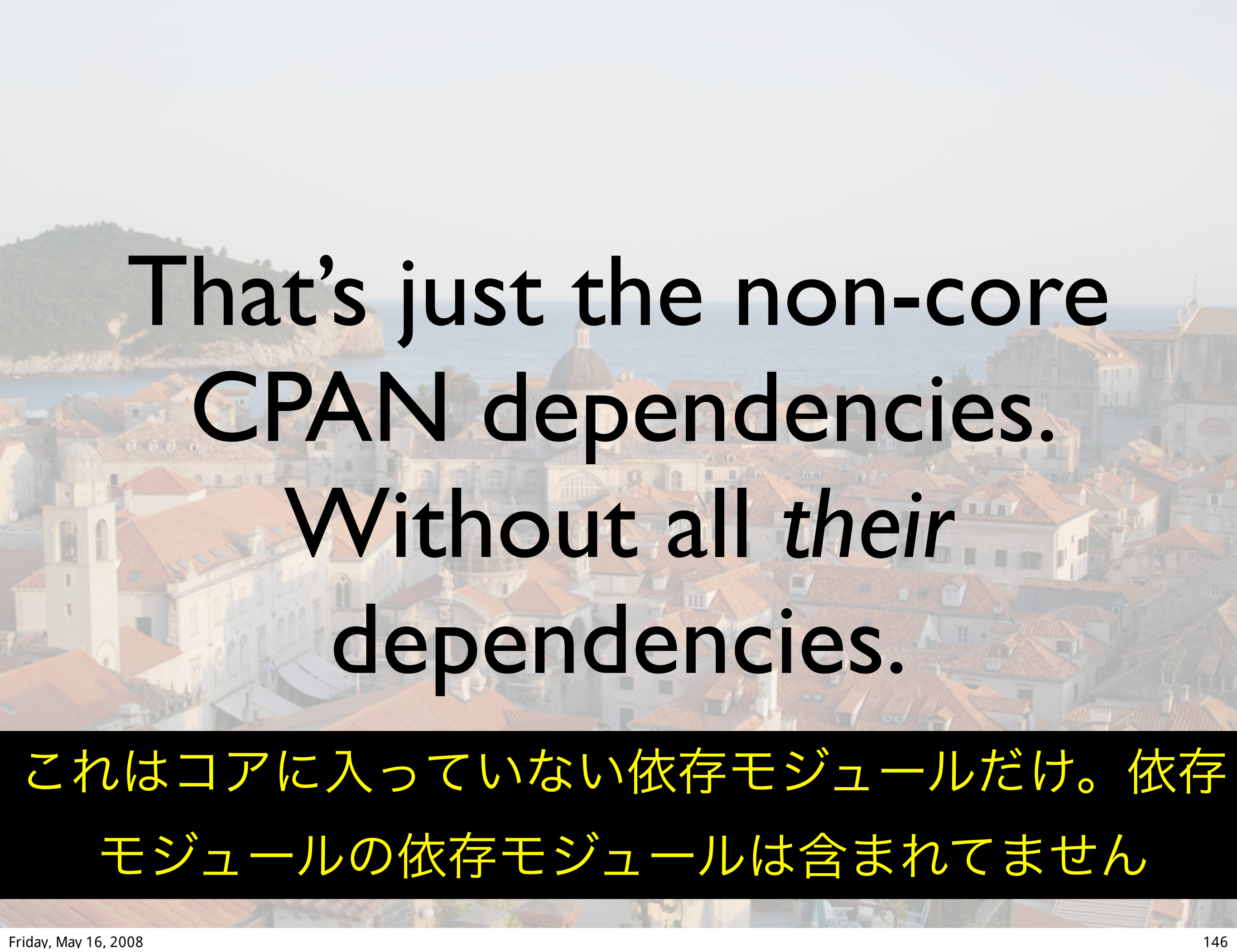
私たちはRTをつくっている



RT has Many Dependencies

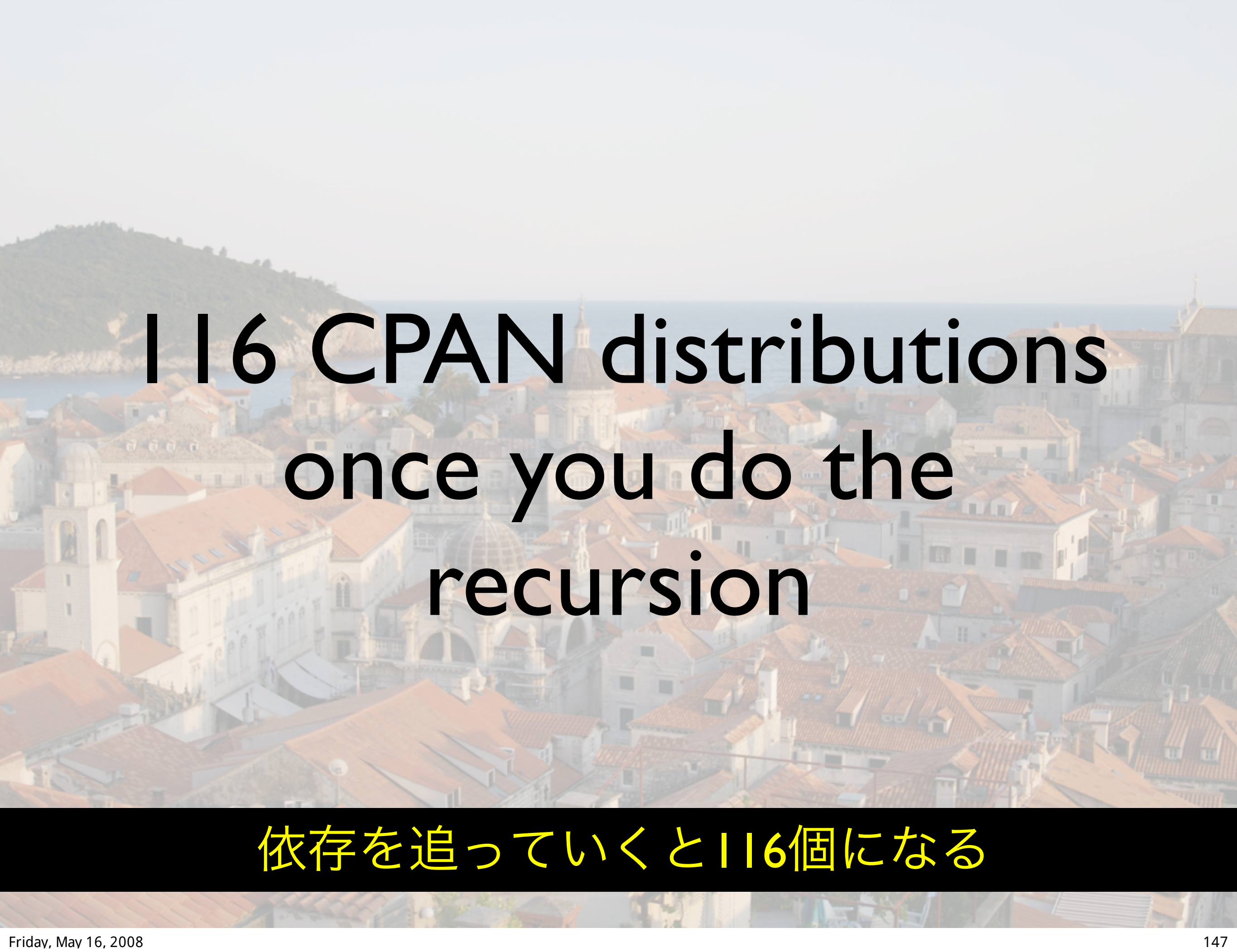
RTは依存が多い

Apache::DBI	Digest::MD5	HTML::Scrubber	Mail::Mailer	Test::Warn
Apache::Request	Digest::base	HTML::TokenParser	Module::Refresh	Text::ParseWords
Apache::Session	File::Find	HTML::TreeBuilder	Module::Versions::Report	Text::Quoted
CGI::Cookie	File::Glob	HTTP::Request::Common	Net::SMTP	Text::Template
CGI::Fast	File::ShareDir	HTTP::Server::Simple	PerlIO::eol	Text::WikiFormat
CGI::SpeedyCGI	File::Spec	HTTP::Server::Simple::Mason	Pod::Usage	Text::Wrapper
CSS::Squish	File::Temp	IPC::Run	Regexp::Common	Time::HiRes
Cache::Simple::TimedExpiry	GD::Graph	IPC::Run::SafeHandles	Scalar::Util	Time::ParseDate
Calendar::Simple	GD::Text	LWP::UserAgent	String::ShellQuote	Tree::Simple
Class::ReturnValue	Getopt::Long	Locale::Maketext	Term::ReadKey	UNIVERSAL::require
DBD::Oracle	GnuPG::Interface	Locale::Maketext::Fuzzy	Term::ReadLine	WWW::Mechanize
DBD::Pg	HTML::Entities	Locale::Maketext::Lexicon	Test::Builder	XML::RSS
DBD::SQLite	HTML::Form	Log::Dispatch	Test::Deep	XML::Simple
DBD::mysql	HTML::FormatText	Log::Dispatch::Perl	Test::Expect	
DBIx::SearchBuilder	HTML::Mason	MIME::Entity	Test::HTTP::Server::Simple	
Data::ICal	HTML::RewriteAttributes	MIME::Types	Test::WWW::Mechanize	



That's just the non-core
CPAN dependencies.
Without all *their*
dependencies.

これはコアに入っていない依存モジュールだけ。依存
モジュールの依存モジュールは含まれてません



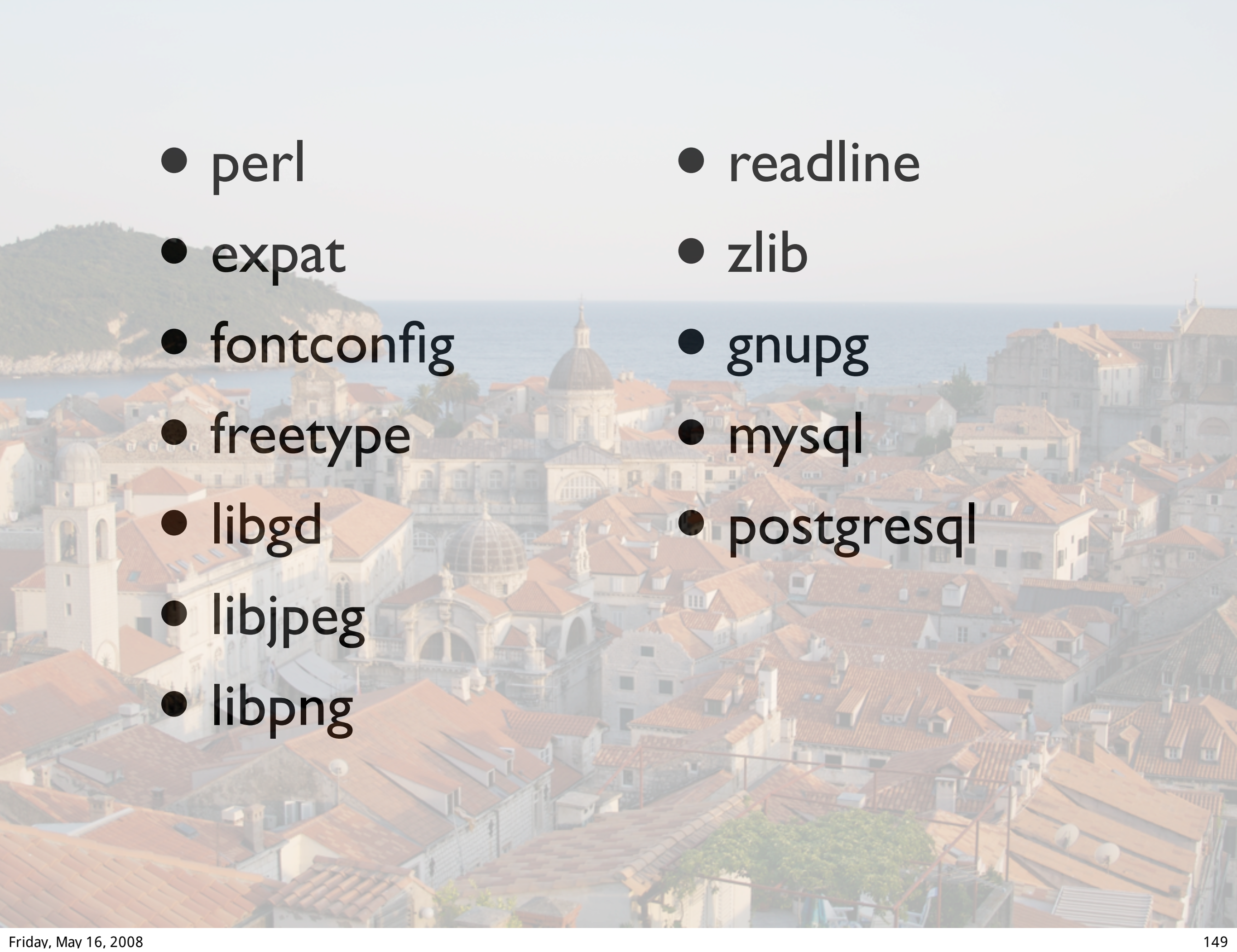
116 CPAN distributions
once you do the
recursion


依存を追っていくと116個になる



And then there are the
external libraries and
programs

そのほかに外部ライブラリとかプログラムもある

- 
- perl
 - expat
 - fontconfig
 - freetype
 - libgd
 - libjpeg
 - libpng
 - readline
 - zlib
 - gnupg
 - mysql
 - postgresql



Plagger beats RT.
(It has 135 CPAN deps)

Plaggerには負けるけど (Plaggerの依存は135個)

An aerial photograph of Dubrovnik, Croatia, showing a dense cluster of buildings with red-tiled roofs. A prominent large dome is visible in the center. The city is situated along the coast, with the sea and a green hill in the background.


RT is hard to install

RTのインストールは大変



RT should be easy to
install

もっと簡単にインストールできないと



All software should be
easy to install

というか、どんなソフトでも簡単にインストールできないとね

What Shipwright does

Tracks all dependencies in version control
すべての依存モジュールをバージョン管理化に
(As far down as you want. I skip libc.)
お好きなだけ。私はlibcまでは追いかけません

Shipwrightがしてくれること

What Shipwright does

Computes a full dependency order
依存の順番を全部計算したら

No more recursive CPAN installs
それ以上CPANからはインストールしません

Shipwrightがしてくれること

What Shipwright does

Keeps all packages in version control

すべてのパッケージをバージョン管理化に

No more “that site is down”

これで「サイトが落ちる」ことはなくなります

Shipwrightがしてくれること

What Shipwright does

Automates multi-package builds
複数パッケージのビルドも自動化

(No manual intervention)
人間の介入する余地はなし

Shipwrightがしてくれること

What Shipwright does

Makes installed packages relocatable

インストールしたパッケージをリロケータブルに

(Wraps all scripts and binaries)

すべてのスクリプトとバイナリをくるんで

Shipwrightがしてくれること

What Shipwright does

Makes historical builds reproducible
過去のビルドを再現可能にする

(No more “Oh. that API changed in 2.0”)
もう「あのAPIは2.0で変更されたよ」というこ
とはなくなります

Shipwrightがしてくれること

Shipwright and CPAN

CPAN-friendly CPANにやさしい

(Not CPAN-only) CPANのみではなく

CPAN dist import CPANモジュールのインポート

Automatic CPAN dependency resolution
依存の自動解決

Full build ordering 完全なビルド順

No more CPAN dependency hell.
もう依存地獄はなし

ShipwrightとCPAN

Shipwright and C

Native support for autoconfiscated packages
非互換パッケージもネイティブサポート

No support for magic C->C dependency
resolutions

CからCへの魔法のような依存解決はサポー
トしていません

ShipwrightとC

Shipwright and Ruby

Not Ruby friendly :(
まだRubyには未対応

No Gem support yet
Gemにも未対応

Hackathon this weekend?
今週のハッカソンで？

ShipwrightとRuby

What you can ship

A platform-neutral source distribution

プラットフォームに依存していないソースディストリビューション

Platform-specific binary distributions

プラットフォーム依存のバイナリディストリビューション

リリースできるもの

Making a Shipwright Vessel

```
svnadmin create file:///Users/jesse/shiprepo
```

```
shipwright create --repository
```

```
svn:file:///Users/jesse/shiprepo/myproject
```

```
shipwright import --repository
```

```
svn:file:///Users/jesse/shiprepo/myproject
```

```
cpan:Plagger
```

Go get a cup of coffee

(Shipwright will chase all 135 dependencies)

コーヒーを飲みに行こう (Shipwrightは135の依存すべてを追跡します)

Shipwrightの器をつくる

Building a Shipwright Vessel

```
svn co file:///Users/jesse/shiprepo/myproject
```

```
cd myproject
```

```
./bin/shipwright-builder
```

(Possibly more coffee)

(たぶんもう一度コーヒーかな)

```
tar czvf myproject.tgz /path/to/build
```

Shipwrightの器をつくる

Using a Shipwright Vessel

- `tar xzvf myproject.tgz c /usr/local/myproject`
- `ln -s /usr/local/myproject/bin/myprogram /usr/local/bin`

Shipwrightの器をつくる

What's next for Shipwright?

Build tool usability improvements
ビルドツールの使い勝手を改善

Ruby support?
Rubyサポート

この先どうしよう？



Get input from users

ユーザからの入力を受けつける

A background image featuring a butterfly with orange and black wings resting on a green plant. The plant has large green leaves and clusters of small, red, tubular flowers. The entire image is semi-transparent, allowing the text to be clearly visible.

Date::Extract

Get a date

日付を取得

A photograph of a butterfly with orange and black wings resting on a green plant with red, tubular flowers. The image is slightly blurred and has a soft, ethereal quality.

The Problem

問題

A close-up photograph of a butterfly with orange and black wings resting on a green leaf. Several red, teardrop-shaped flower buds are visible in the background.

Pick up the drycleaning
on tuesday after school

火曜日の学校帰りにドライクリーニングを取ってくること

A background image of a butterfly with orange and black wings perched on a pink flower with green leaves.

There are many Date / Time parsers on CPAN

CPANには日時のパーサがたくさんあります

A butterfly with orange and black wings is the central focus, resting on a large green leaf. The background is filled with more green foliage and clusters of small red flowers. The text "Date::Manip" is overlaid in the center in a large, black, sans-serif font.

Date::Manip

A close-up photograph of a butterfly with orange and black wings, perched on a red flower. The background is filled with green leaves and other flowers. The text 'Time::ParseDate' is overlaid in the center in a large, black, sans-serif font.

Time::ParseDate

A butterfly with orange and black wings is the central focus, resting on a large green leaf. The background is filled with more green foliage and clusters of small red flowers. The text "Date::Parse" is overlaid in the center in a large, black, sans-serif font.

Date::Parse

A close-up photograph of a butterfly with orange and black wings, perched on a red flower. The background is filled with green leaves and other flowers, creating a natural setting. The text "DateTime::Format::Natural" is overlaid in the center of the image.

DateTime::Format::Natural

A close-up photograph of a butterfly with orange and black wings, perched on a red flower. The background is filled with green leaves and other red flowers, creating a natural, garden-like setting. The text is overlaid on the center of the image.

Each can parse dates
and times

どれも日時のパースはできますが



None of them are any
good at finding dates in
a block of text

テキストの塊から日付を見つけられるものではありません

A butterfly with orange and black wings is perched on a green plant with large leaves and clusters of red, tubular flowers. The background is a soft-focus green.

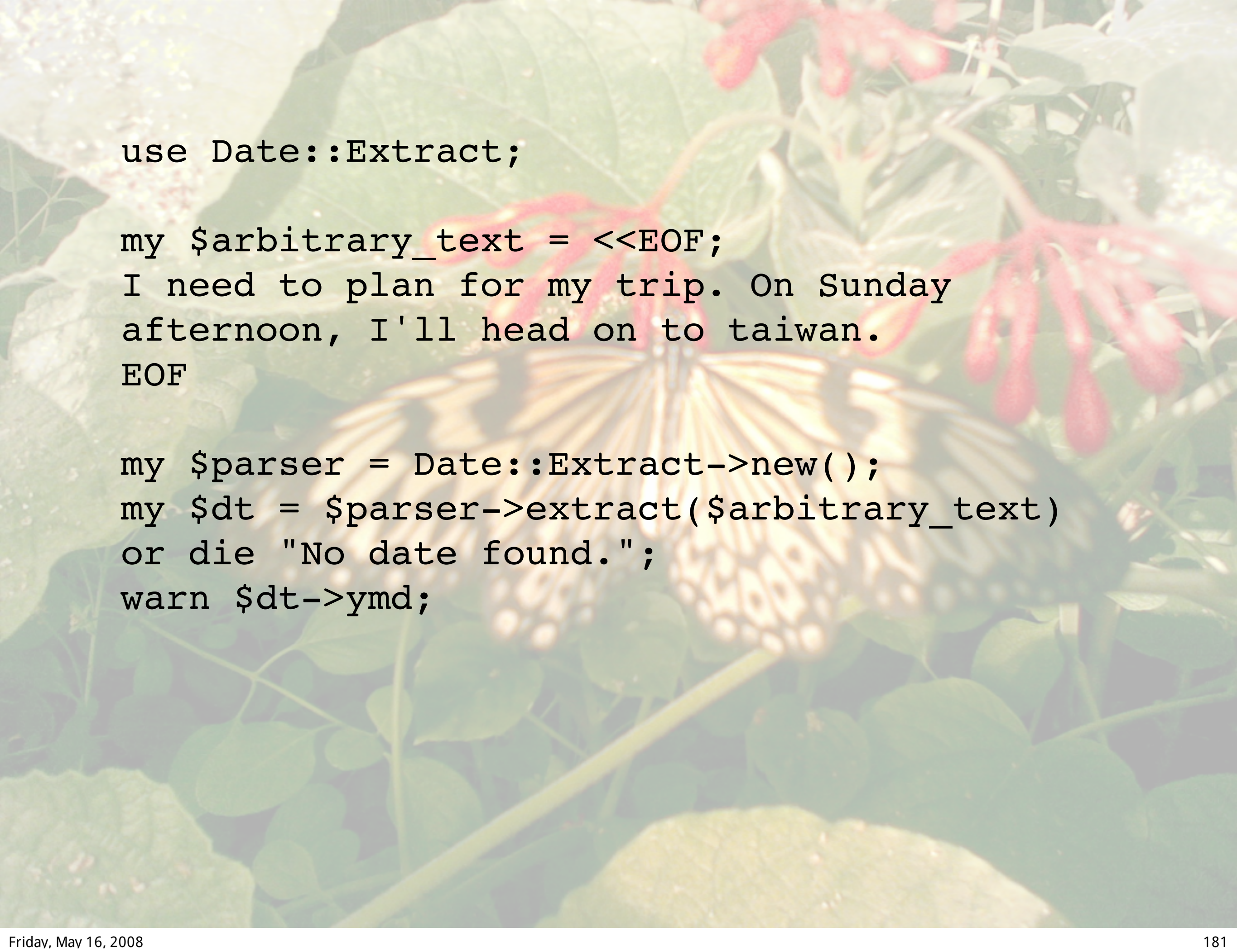
Timezones are hard

タイムゾーンもむずかしい

A monarch butterfly with orange and black wings is perched on a milkweed plant. The plant has large green leaves and clusters of red, tubular flowers. The background is a soft-focus view of the same plant.

**Most parsers generate
many false positives**

たいていのものは違うものがたくさん混じってしまう

A background image featuring a large, vibrant orange and black butterfly perched on a pink flower. The scene is set against a backdrop of lush green foliage and other pink flowers, creating a natural and serene environment.

```
use Date::Extract;
```


```
my $arbitrary_text = <<EOF;  
I need to plan for my trip. On Sunday  
afternoon, I'll head on to taiwan.  
EOF
```

```
my $parser = Date::Extract->new();  
my $dt = $parser->extract($arbitrary_text)  
or die "No date found.";  
warn $dt->ymd;
```




It's simple.

简单

A photograph of a butterfly with orange and black wings, perched on a plant with large green leaves and clusters of red, tubular flowers. The image is slightly faded, serving as a background for the text.

It's unsexy.

美しくはない

A photograph of a butterfly with orange and black wings, perched on a green plant with red, tubular flowers. The image is slightly faded and serves as a background for the text.

It's reliable.

信頼できる

What's next for Date::Extract?

Parsing more obscure human-written dates? もっと
あいまいな日付もパースできるようにする？

Date::Extract::Lingua::* ?

この先どうしよう？

The Feedback Box 意見箱

Let users make suggestions everywhere

ユーザがどこでも提案できるようにする



mailto: links?



getsatisfaction.com?

People-powered customer service & support for I want Sandy

http://getsatisfaction.com/iwantsandy RSS Google

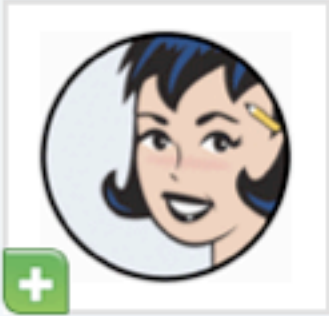
How to buy a...ulSteves.com My DebugBar ... / HomePage Workflow soft...ng the bluff. Code rant: Vi... 's Nightmare CSS Gradient ...esigner Wall

Home Companies Products You Sign In Or Sign up to get Satisfaction

Find a: Topic in I want Sandy Go

GET SATISFACTION

Home / Companies / I want Sandy





I want Sandy
I want Sandy is here!
3 employees are listening and participating
[Put a topic widget on your website](#)
[About I want Sandy](#)

What do you want to ask?

I'd like some help with sandy. [Go](#)

OR [share an idea](#), [report a problem](#), or [start a discussion](#)

They make **3 products** and have **2 products** related to them.




Explore topics by popular tags


[email](#), [reminder](#), [sandy](#), [sms](#), [tags](#), [ical](#), [twitter](#), [jott](#), [daily digest](#), [mobile](#)

[Explore all popular tags in I want Sandy »](#)

Recently active topics in I want Sandy


All Topics(1051) Questions(439) Ideas(199) Problems(348) Talk(65) 

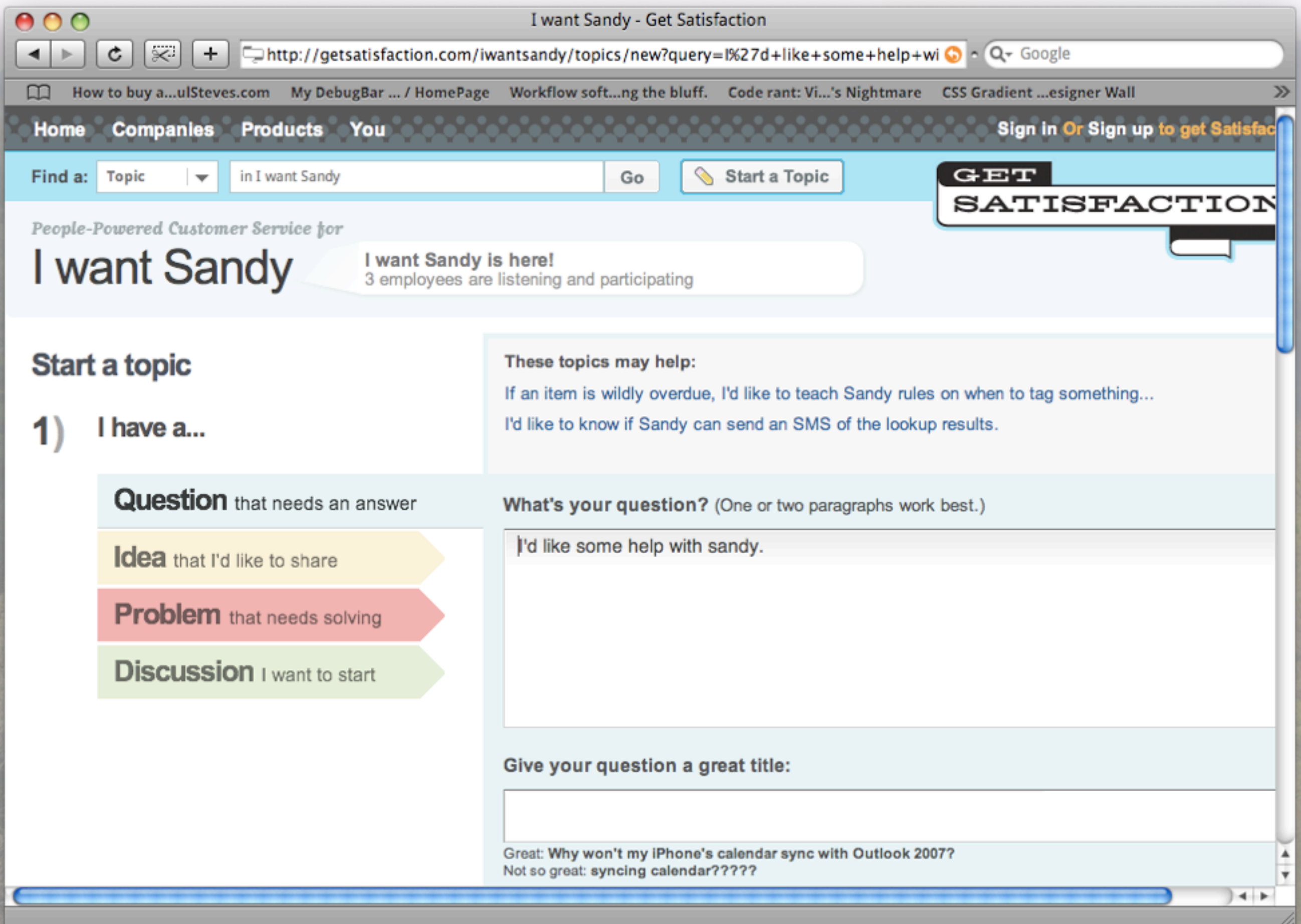
— [Recently Active](#) [Latest Topics](#) [Unanswered](#) 1-6 of 1051 topics

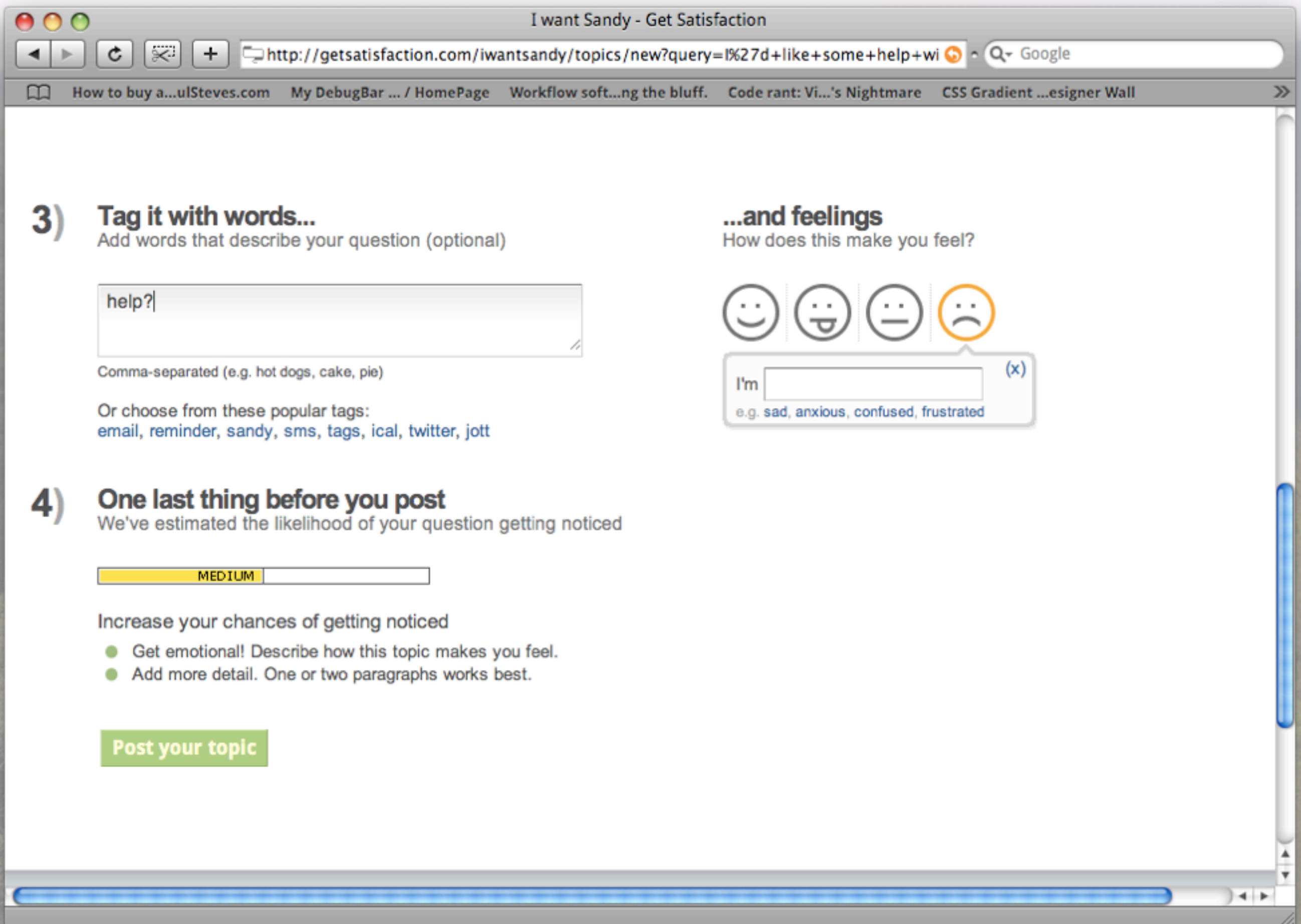
 **UPDATE for May 5th, 2008**
Last reply 6 minutes ago

13 Official Reps

I want Sandy has something to say to you.
Yes, you!

 **UPDATE for May 12th, 2008**





I want Sandy - Get Satisfaction

http://getsatisfaction.com/iwantsandy/topics/new?query=I%27d+like+some+help+wi

Google

How to buy a...ulSteves.com

My DebugBar ... / HomePage

Workflow soft...ng the bluff.

Code rant: Vi... 's Nightmare

CSS Gradient ...esigner Wall

MEDIUM

Increase your chances of getting noticed

- Get emotional! Describe how this topic makes you feel.
- Add more detail. One or two paragraphs works best.

Post your topic

Create a new account to post your topic

Email Address:

Your email address will always remain private

Nickname:

Publicly displayed nickname

Tip: Real people usually get more respect in online communities. [Add your real name and password »](#)

We need to confirm you're human:

YorkFontenort

Can't read it? [Try a different one.](#)

Type the two words:

Sign up

Log in if you already have an account

Email:

Password:

[Forgot your password?](#)

☐ Remember me

Sign in

Start a topic

Ask a question

Share an idea

Report a problem

Start a discussion

© Copy

Friday, May 16, 2008

192



I give up.

私ならあきらめます



Don't let your users
give up.

でも、あきらめてほしくないですね

A scenic background image of a mountain valley. In the foreground, a river flows through a lush green forest. A small bridge crosses the river in the middle ground. The background features rolling mountains covered in dense green foliage under a clear sky.

Feedback is *important!*

フィードバックは重要です



Feedback can be *easy*!

フィードバックは簡単にできます！

Send us feedback!

For customer support,
please visit the [support](#)
[page](#).

Tell us what's good,
what's bad, and what
else you want
Hiveminder to do!



Send this feedback!

Send us feedback!

For customer support,
please visit the [support](#)
[page](#).

I'd really like Hiveminder
to do all my work for me!

Send this feedback!

3797 @task



Thanks for the
feedback. We
appreciate it!


Send us feedback!

For customer support,
please visit the [support](#)
[page](#).

Tell us what's good,
what's bad, and what
else you want
Hiveminder to do!



Send this feedback!



If you make it easy for
users to give you
feedback...

ユーザがフィードバックしやすくしておく

A scenic view of a river flowing through a lush green valley with mountains in the background. The river is in the foreground, surrounded by dense green trees and vegetation. In the middle ground, a small bridge crosses the river. The background features rolling green hills and mountains under a clear sky.

You can capture lots of
good information!

ためになる情報をたくさんもらえます！

From: Jesse Vincent / Feedback <comment-373709-destygrypuju@tasks.hiveminder.com>
Date: Mon, 12 May 2008 12:31:49 -0700
Subject: Up for grabs: I'd really like Hiveminder to do all my work for me! (#EFYG)

Jesse Vincent <jesse@bestpractical.com> created a task and put it up for grabs:

<http://task.hm/EFYG>

I'd really like Hiveminder to do all my work for me!

--

Private debugging information:

Is a Pro user!

HTTP_ACCEPT: application/xml, text/xml, */*
HTTP_ACCEPT_ENCODING: gzip, deflate
HTTP_ACCEPT_LANGUAGE: en-us
HTTP_CONNECTION: keep-alive
HTTP_COOKIE: JIFTY_SID_HIVEMINDER=b58e123456719fd411bb9f3a8123458f;
HTTP_HOST: hiveminder.com
HTTP_REFERER: http://hiveminder.com/todo/
HTTP_USER_AGENT: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_2; en-us)
 AppleWebKit/526.5+ (KHTML, like Gecko) Version/3.1.1 Safari/525.18
HTTP_X_REQUESTED_WITH: XMLHttpRequest
REMOTE_ADDR: 75.147.59.54
REMOTE_PORT: 35412
REQUEST_METHOD: POST
REQUEST_URI: /todo/

A scenic view of a mountain valley. In the foreground, a river flows through a lush green forest. A small bridge crosses the river in the middle ground. The background features rolling mountains covered in dense green foliage under a clear sky.


Collect lots of feedback

フィードバックをたくさん集めましょう

A scenic view of a mountain valley. In the foreground, a river flows through a lush green forest. A small bridge crosses the river in the middle ground. The background features rolling mountains covered in dense green foliage under a clear sky.


Reply to lots of feedback

フィードバックにはどんどん返信しましょう



Conversations make
users happy

対話することでユーザはうれしい

A scenic view of a river flowing through a lush green valley with mountains in the background. The river is in the foreground, surrounded by dense green trees and foliage. In the middle ground, a small bridge crosses the river. The background features rolling green hills and mountains under a clear sky.

Happy = Loyal

うれしい = 忠実



Own the inbox

受信箱を用意します



Net::IMAP::Server


An IMAP Server in Pure Perl

Pure PerlのIMAP Server



Why?

なぜ



Offline Access

オフラインアクセス



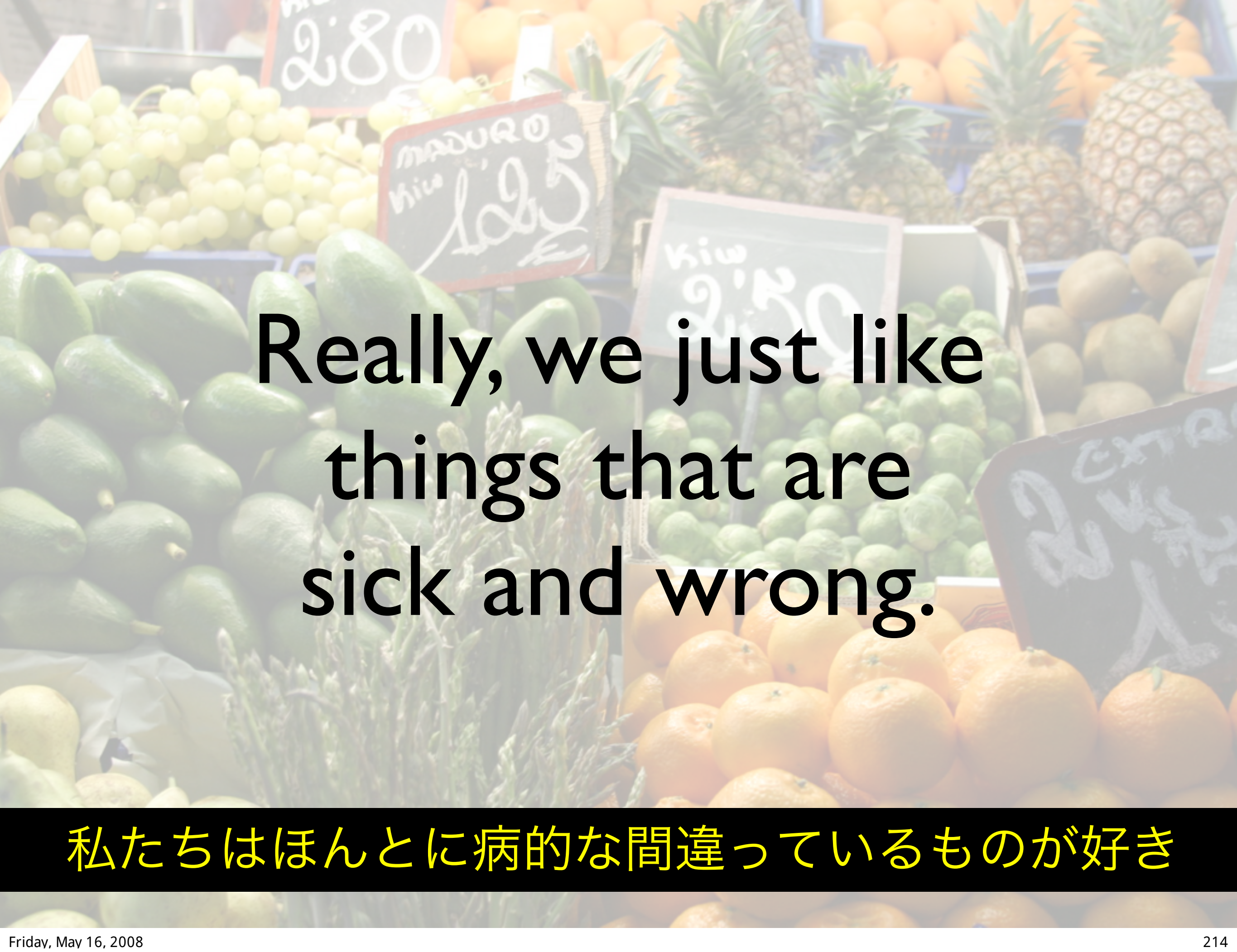
iPhone Access

iPhoneアクセス




Desktop Access

デスクトップアクセス



Really, we just like
things that are
sick and wrong.

私たちはほんとに病的な間違っているものが好き

A vibrant display of fresh produce including green grapes, avocados, pineapples, and oranges, with handwritten price tags.

Now Hiveminder users
can take their tasks
with them

これでHiveminderユーザはタスクを持ち歩けるように



Magic mailboxes

魔法のメールボックス



Update your tasks
while offline

オフラインでタスクを更新



What sort of mailbox
do you use with that?

どんな種類のメールボックスを使う



mbox?



no



maildir?



no



How about Perl?

Perlならどう？

A photograph of a fruit stand with various produce including grapes, avocados, pineapples, kiwis, and oranges, with handwritten price tags.

Net::IMAP::Server



Complete RFC 3501 Implementation

RFC 3501を完全に実装



Write your own
backends

バックエンドは自分で



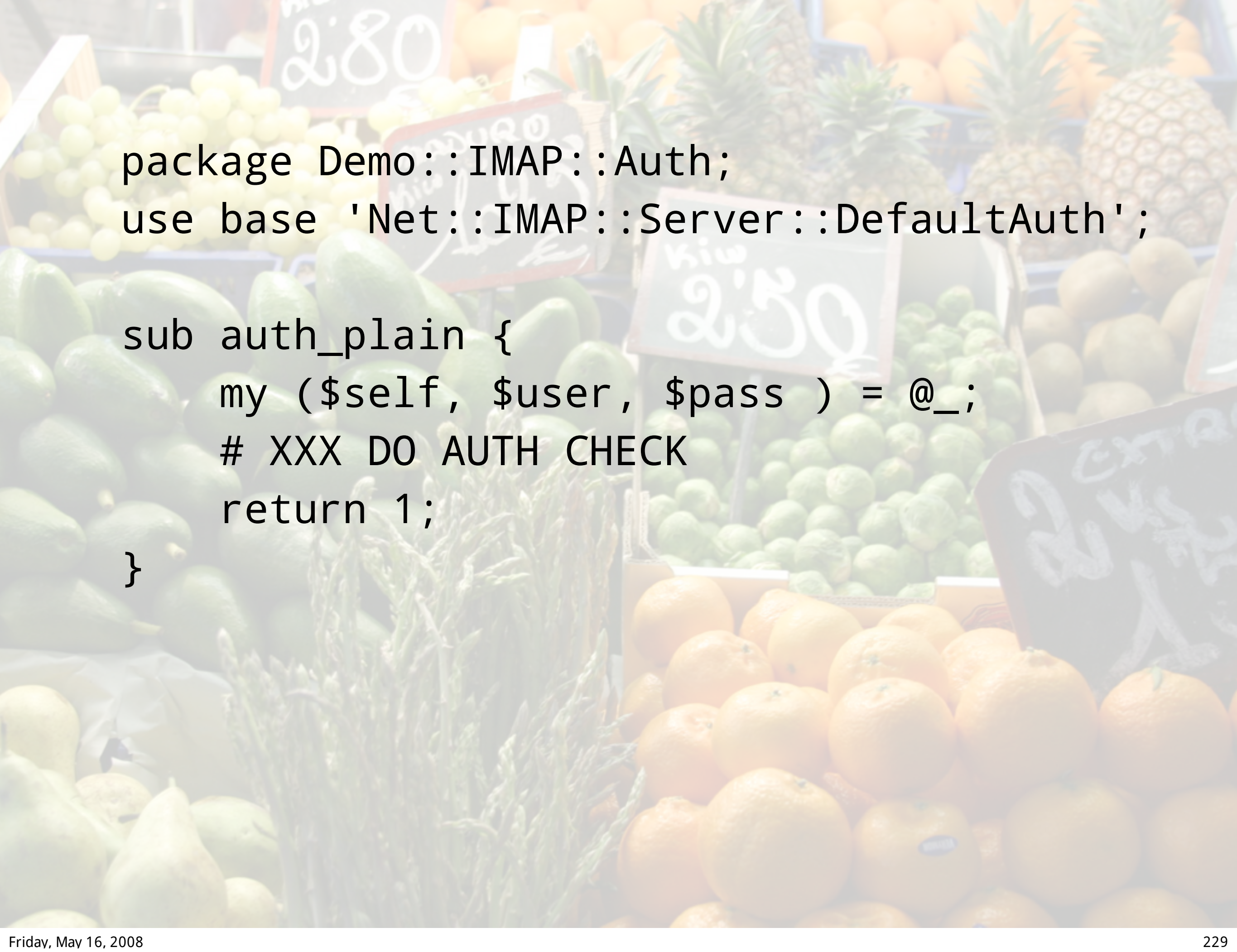
Let's write one now

書きましょう

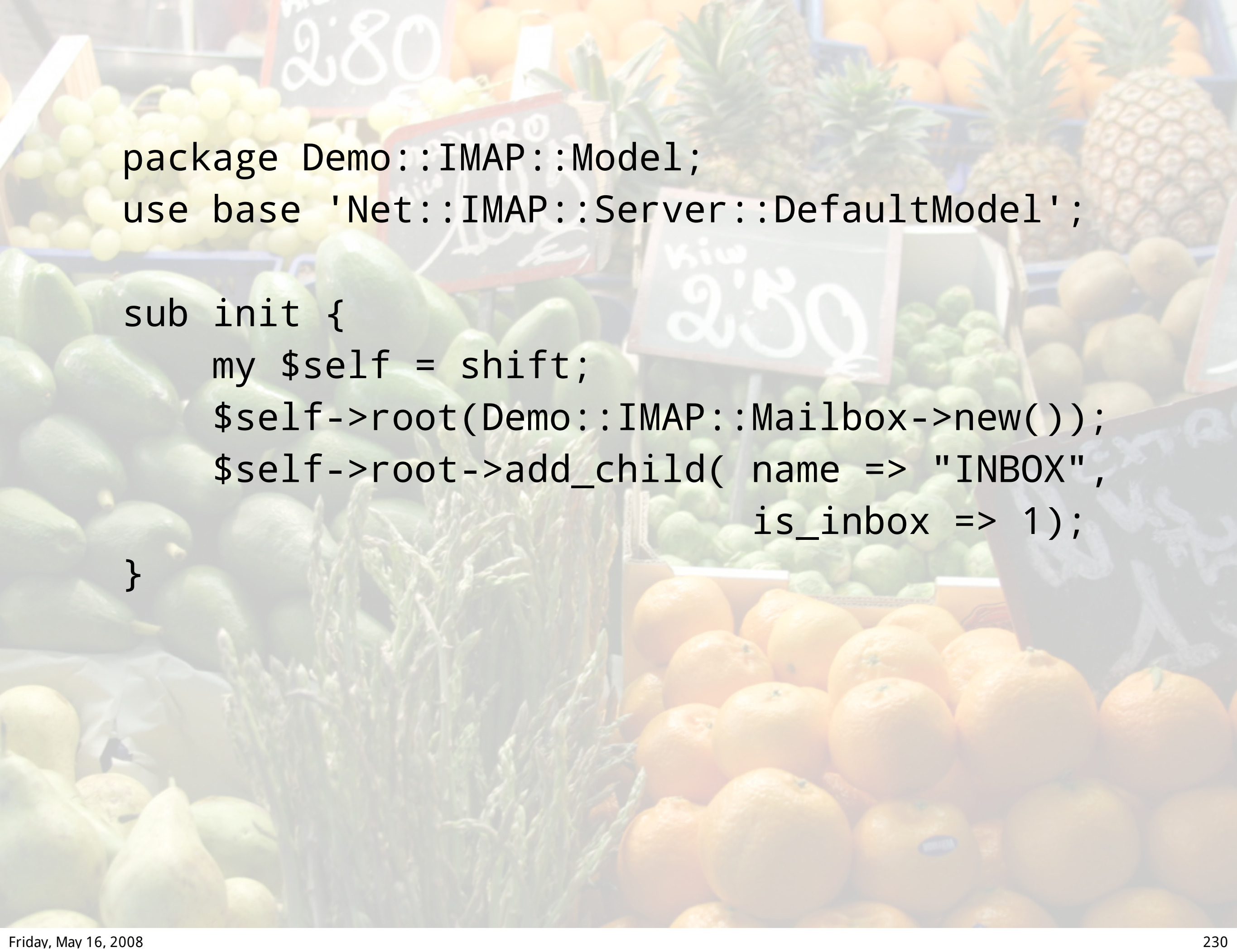


How about a
Spam server?

Spamサーバはどうする？



```
package Demo::IMAP::Auth;  
use base 'Net::IMAP::Server::DefaultAuth';  
  
sub auth_plain {  
    my ( $self, $user, $pass ) = @_;  
    # XXX DO AUTH CHECK  
    return 1;  
}
```

```
package Demo::IMAP::Model;
use base 'Net::IMAP::Server::DefaultModel';


sub init {
    my $self = shift;
    $self->root(Demo::IMAP::Mailbox->new());
    $self->root->add_child( name => "INBOX",
                           is_inbox => 1);
}
```



```
package Demo::IMAP::Mailbox;  
use base qw/Net::IMAP::Server::Mailbox/;  
  
my $msg = Net::IMAP::Server::Message->new(<<'EOM');  
From: jesse@example.com  
To: user@example.com  
Subject: This is a test message!
```


```
Hello. I am executive assistant to the director of  
Bear Stearns, a failed investment Bank. I have  
access to USD5,000,000. ...  
EOM
```

```
sub load_data {  
    my $self = shift;  
    $self->add_message($msg);  
}
```

```
#!/usr/bin/perl

use Net::IMAP::Server;
Net::IMAP::Server->new(
    auth_class    => "Demo::IMAP::Auth",
    model_class   => "Demo::IMAP::Model",
    user          => 'nobody',
    port          => 143,
    ssl_port      => 993
)->run();
```

It runs! Ship it!
(Just add SSL certificates)


できた！ リリースしましょう！（SSL証明書も忘れずに）



with.hm

Assign tasks by email

メールでタスクを割り当て



We wanted to make it
easy to work with
Hiveminder by email.

Hiveminderをメールでなんとかしたかった



We had a few ideas...

アイディアはいくつかあったけど...

Cc:



Too clumsy

ダサすぎ



Bcc:



Too likely to break

すぐに壊れそう

Commands in the body

本分にコマンド




Too ugly

みっともないよ



hm...

ふむ



What could we do?

どうしよう

DNS Hacks!

Everybody loves DNS hacks

みんなDNSハックが大好き



.hm



Hiveminder!

<http://hiveminder.com/>

Hiveminder™

Collaborative todo lists that work the way you do.

1. Create

Braindump
your tasks, tag
them, set due
dates, and
attach notes

2. Share

Set up
reminders for
yourself, create
groups, and
share tasks
with others

3. Review

Check out your
tasks, set
priorities, and
make decisions

**4. Bee
productive!**

(Get used to
the bee puns.)

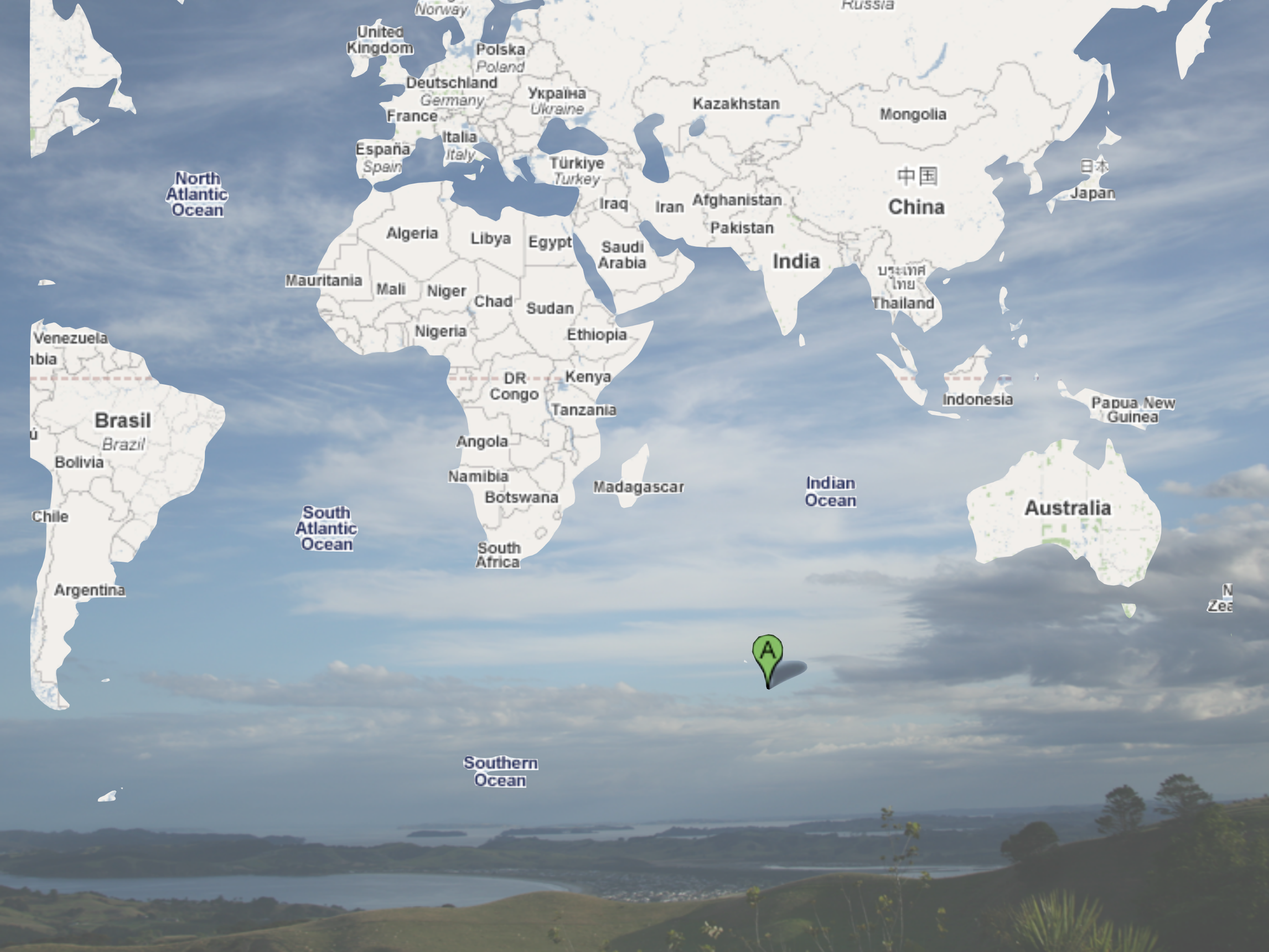
[Take the full tour! >>](#)



.hm

Heard and McDonald Islands!

ハード島とマクドナルド諸島



North
Atlantic
Ocean

Mauritania

South
Atlantic
Ocean

Southern
Ocean

Indian
Ocean

N
Zea

United
Kingdom

Norway

Polska
Poland

Deutschland
Germany

France

España
Spain

Italia
Italy

Україна
Ukraine

Türkiye
Turkey

Kazakhstan

Mongolia

中国

China

日本

Japan

Iraq

Iran

Afghanistan

Pakistan

India

ประเทศไทย
ไทย

Thailand

Indonesia

Papua New
Guinea

Australia

Venezuela

Colombia

Brasil

Brazil

Bolivia

Chile

Argentina

Algeria

Libya

Egypt

Saudi
Arabia

Mali

Niger

Chad

Sudan

Nigeria

Ethiopia

DR
Congo

Kenya

Tanzania

Angola

Namibia

Botswana

South
Africa

Madagascar

A



with.hm

Send your email with Hiveminder

メールをHiveminderで送る

Wildcard MX Record!

ワイルドカードMXレコード!



*.with.hm ➡
mail.hiveminder.com

An Example:

From: jesse@bestpractical.com

To: miyagawa@bulknews.net.with.hm

Subject: Could you release a new version of plagger?




My mail server looks up
bulknews.net.with.hm

メールサーバがbulknews.net.with.hmを探して


“Oh. That's handled by
mail.hiveminder.com”

「ああ、これはmail.hiveminder.comで処理ね」




mail.hiveminder.com
unwraps the address.

mail.hiveminder.comがアドレスを取り出して



Hiveminder creates a task
for miyagawa@bulknews.net

miyagawa@bulknews.netにタスクを作成して



Hiveminder emails
miyagawa@bulknews.net

miyagawa@bulknews.netにメール

Preventing Spam

スパムを防ぐために

miyagawa@bulknews.net.my-
secret.with.hm

•

I bet you can come up
with even scarier uses
of wildcard DNS.

ワイルドカードDNSを使うともっとすごいこともできるよ



Thank you!

ありがとう