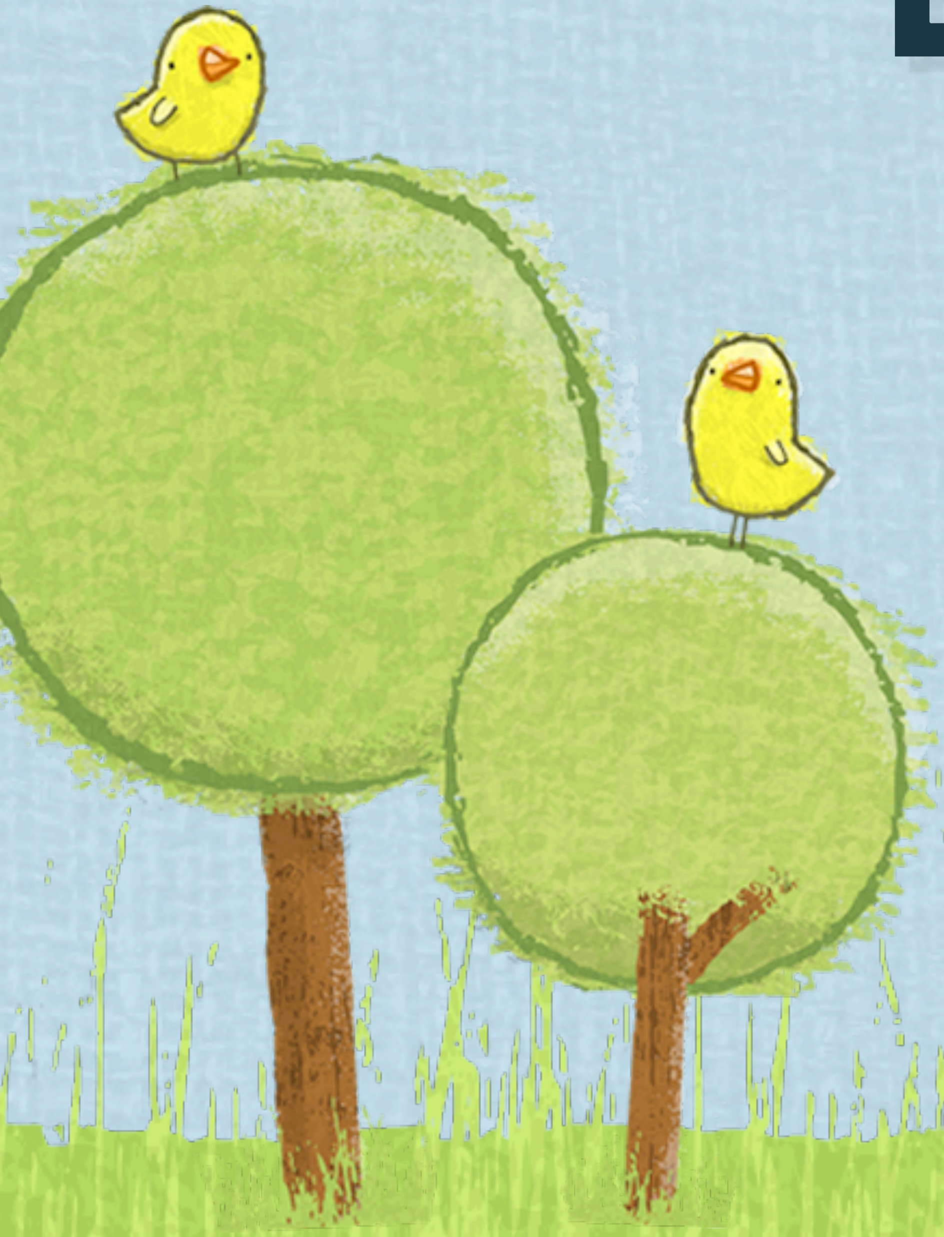
-HOW WE WROTE A- TEMPLATE ENGINE



Drew McLellan
Perch CMS
grabaperch.com

RDG Geek Night
10th September 2013

LIGHTWEIGHT PHP CMS FOR SMALL WEBSITES

WE NEEDED A REALLY GOOD TEMPLATE LANGUAGE

-SO-

WHAT WAS ALREADY OUT THERE?

SMARTY

(NOT SO SMART)

TWIG

(NOT SO MATURE)

JUST PHP
-WITH A-
DIFFERENT SYNTAX

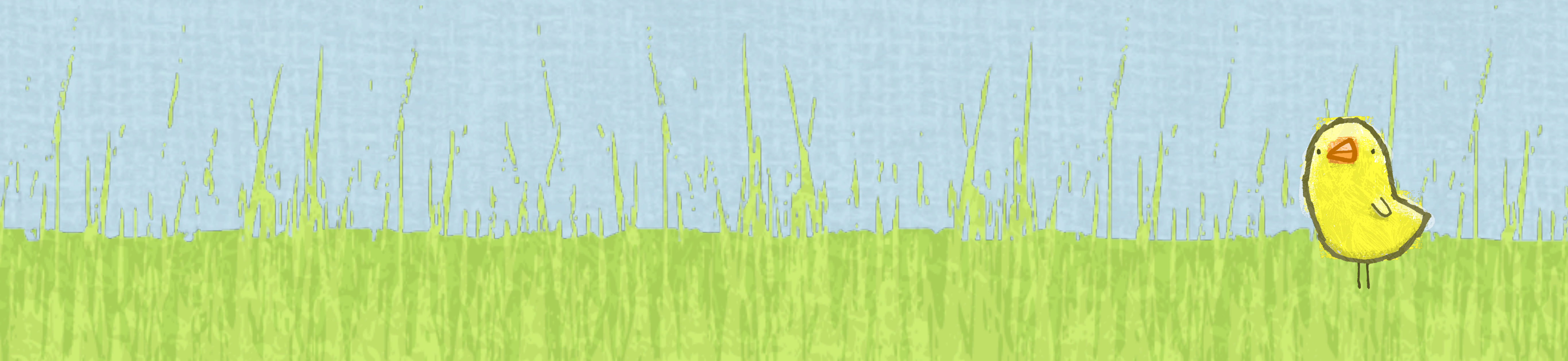
```
<perch:content
    type=”text”
    id=”heading”
    label=”Main heading”
    required=”true” />
```

```
<h2><perch:content id="heading"
type="text" label="Heading"
required="true" title="true" /></h2>

<p class="date"><perch:content id="date"
type="date" label="Date" format="%d %B
%Y" /></p>

<perch:content id="body" type="textarea"
label="Body" textile="true"
editor="markitup" required="true" />
```

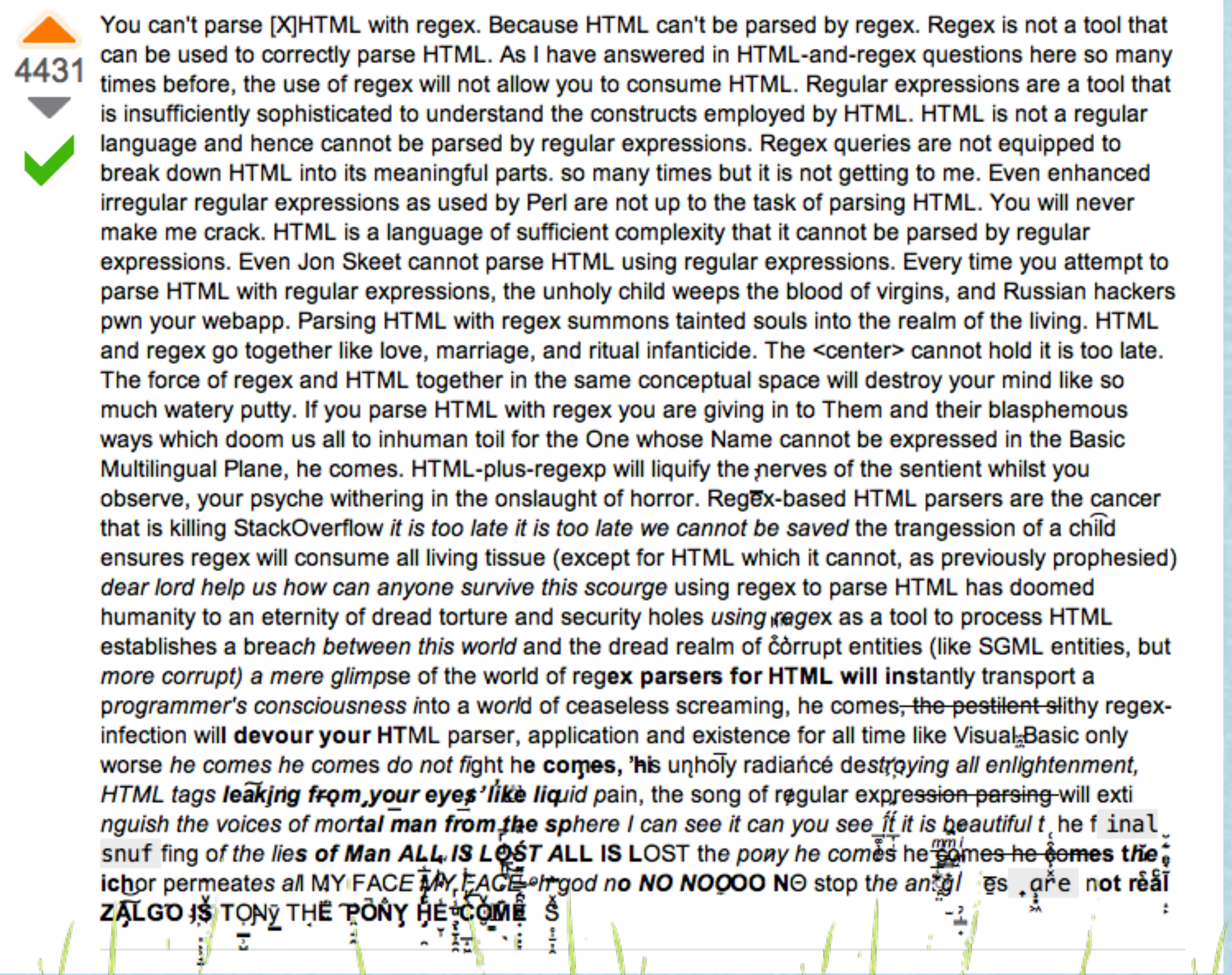
HTML IS LIKE XML RIGHT?



```
<img src="<perch:content
id="photo" type="image"
label="Photo" />" />
```

EVERYBODY STAND BACK.
I KNOW REGULAR EXPRESSIONS.

xkcd.com/208



4431

You can't parse [X]HTML with regex. Because HTML can't be parsed by regex. Regex is not a tool that can be used to correctly parse HTML. As I have answered in HTML-and-regex questions here so many times before, the use of regex will not allow you to consume HTML. Regular expressions are a tool that is insufficiently sophisticated to understand the constructs employed by HTML. HTML is not a regular language and hence cannot be parsed by regular expressions. Regex queries are not equipped to break down HTML into its meaningful parts. so many times but it is not getting to me. Even enhanced irregular regular expressions as used by Perl are not up to the task of parsing HTML. You will never make me crack. HTML is a language of sufficient complexity that it cannot be parsed by regular expressions. Even Jon Skeet cannot parse HTML using regular expressions. Every time you attempt to parse HTML with regular expressions, the unholy child weeps the blood of virgins, and Russian hackers pwn your webapp. Parsing HTML with regex summons tainted souls into the realm of the living. HTML and regex go together like love, marriage, and ritual infanticide. The <center> cannot hold it is too late. The force of regex and HTML together in the same conceptual space will destroy your mind like so much watery putty. If you parse HTML with regex you are giving in to Them and their blasphemous ways which doom us all to inhuman toil for the One whose Name cannot be expressed in the Basic Multilingual Plane, he comes. HTML-plus-regexp will liquify the nerves of the sentient whilst you observe, your psyche withering in the onslaught of horror. Regex-based HTML parsers are the cancer that is killing StackOverflow *it is too late it is too late we cannot be saved* the trangession of a child ensures regex will consume all living tissue (except for HTML which it cannot, as previously prophesied) *dear lord help us how can anyone survive this scourge* using regex to parse HTML has doomed humanity to an eternity of dread torture and security holes *using* regex as a tool to process HTML establishes a brea*ch between this world* and the dread realm of corrupt entities (like SGML entities, but *more corrupt) a mere glimp*se of the world of reg**ex parsers for HTML will ins**tantly transport a p*rogrammer's consciousness i*nto a wor*l*d of ceaseless screaming, he comes~~, the pestilent sl~~ithy regex-infection wil**l devour your** HT*M*L parser, application and existence for all time like Visual Basic only worse *he comes he comes do not fi*ght h**e comes, 'hi**s unholy radiance des*troying all enlightenment, HTML tags* ***leaking from your eyes like liq****uid* pain, the song of regular exp~~ression parsing~~ will exti*nguish the voices of mor**tal man from the sp**here I can see it can you see it it is beautiful t* he f inal snuf fing o*f the lie**s of Man ALL IS LOST ALL IS L**OST th*e pony he comes* he ~~comes he comes~~ **th*e* ich**or permeates a*ll* MY FACE MY FACE oh god n**o NO NOOOO N**Θ stop th*e an gl* es are n**ot real ZALGO IS TONY THE PONY HE COMES**

stackoverflow.com/questions/1732348/

```
<perch:content type=”text”
  id=”heading” label=”Main heading”
  required=”true” />
```

```
#<perch:content[^>]*/>#
```

```
$Tag = new PerchXMLTag(
'<perch:content id="foo" bar="baz" />');

echo $Tag->bar; // baz
```

```
#([a-z-]*)="([^"]*)"#
```

```
#([a-z-]+)=[\"]([^\"\\\\]*(?:\\\\.[^\"\\\\]*)*)[\"]#
```

-WAIT-

REGULAR EXPRESSIONS ARE SLOW, RIGHT?

REGEX ARE SLOW UNLESS YOU NEED THEM

```
<perch:if exists=″foo”>
   ... loadsamarkup ...
</perch:if>
```

```
#(<perch:if[^>]*>)(((?!
perch:if).)*)</perch:if>#
```

-WE NEEDED-
A NEW WAY
-TO PARSE-
TAG PAIRS

```
<perch:if exists=”a”>
  <perch:if exists=”b”>
    <perch:if exists=”c”>
      Hello, world.
    </perch:if>
  </perch:if>
<perch:else />
  <perch:if exists=”d”>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

```
<perch:if exists=”a”>
  <perch:if exists=”b”>
    <perch:if exists=”c”>
      Hello, world.
    </perch:if>
  </perch:if>
<perch:else />
  <perch:if exists=”d”>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

```
<perch:if exists=”a”>
  <perch:if exists=”b”>
    <perch:if exists=”c”>
      Hello, world.
    </perch:if>
  </perch:if>
<perch:else />
  <perch:if exists=”d”>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

```
<perch:if exists=”a”>
  <perch:if exists=”b”>
    <perch:if exists=”c”>
      Hello, world.
    </perch:if>
  </perch:if>
<perch:else />
  <perch:if exists=”d”>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

```
<perch:if exists=”a”>
  <perch:if exists=”b”>
    <perch:if exists=”c”>
      Hello, world.
    </perch:if>
  </perch:if>
<perch:else />
  <perch:if exists=”d”>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

```
<perch:if exists=″a″>
  <perch:if exists=″b″>
    <perch:if exists=″c″>
      Hello, world.
    </perch:if>
  </perch:if>
<perch:else />
  <perch:if exists=″d″>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

```
<perch:if exists=”a”>
  <perch:if exists=”b”>

      Hello, world.

  </perch:if>
<perch:else />
  <perch:if exists=”d”>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

```
<perch:if exists=”a”>
  <perch:if exists=”b”>

      Hello, world.

  </perch:if>
<perch:else />
  <perch:if exists=”d”>
    Goodbye, cruel world.
  </perch:if>
</perch:if>
```

CLOSING TAGS

-ARE -

EASY

NO MORE
WINDOWS
CRASHES

MAYBE

- WE WILL -

REWRITE IT

THANK YOU

Perch is at grabaperch.com

On most things, I am @drewm