

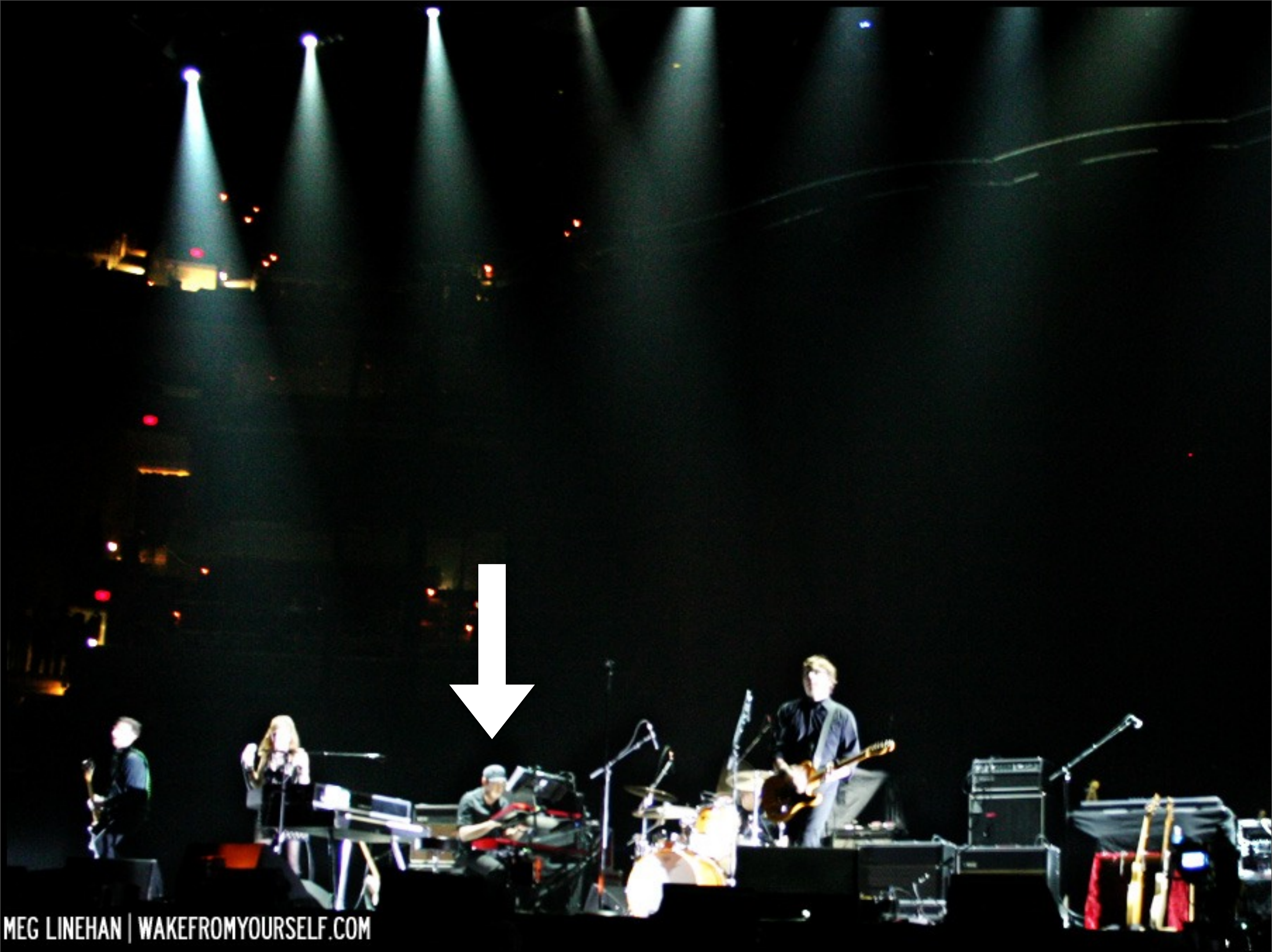


Beautiful Markup

*Curing DIV-itis with Semantic HTML,
CSS and Presenters*

JOHN ATHAYDE, INFOETHER
RAILS CONF 2010 BALTIMORE, MARYLAND





MEG LINEHAN | WAKEFROMYOURSELF.COM

Wednesday, June 9, 2010



<http://www.flickr.com/photos/frengo/992920521/>



Beautiful Markup

*Curing DIV-itis with Semantic HTML,
CSS and [maybe] Presenters*

JOHN ATHAYDE, INFOETHER
RAILS CONF 2010 BALTIMORE, MARYLAND

*(or please stop killing baby animals
with your HTML and CSS)*

Let's Assume...

- » YOU KNOW **HTML** & **CSS** BASICS
- » YOU KNOW **RAILS**
- » YOU WANT TO MAKE IT **BETTER**

STOP

Please.



Let's Talk About...

what the problem is in the first place.



Views are still...

The Wild West

http://www.flickr.com/photos/toby_d1/3588461929/

An example... a 60 line partial (called 30 times in one page)

[illegible]

<http://gist.github.com/78365>

A quick cleanup...

```
<ol class="latest_uploads">
  <% playlist.listings.each_with_index do |listing, i| -%>
    <% if i < show %>
      <% composition = listing.composition %>
      <% position = i %>
      <li class="playlist_item">
        <p class="track_title"><%= link_to awesome_truncate(composition.title,30), composition_path(composition), :title =>
composition.title %></p>
        <p class="sm2_link"><a href="<%= composition.preview_url %>"><span class="hide_me">Preview</span>&nbsp;&nbsp;&nbsp;</a>
        <% if composition.network? %>
          <%= link_to image_tag("button_playlist_download.gif", :alt => "Download"), downloads_path(:audio_item_id =>
composition.audio_items.first.id), :method => :post%>
        <% else %>
          <%= link_to(image_tag("cart.gif", :alt => "Add to cart", :title => "Add to cart"), selections_path(:salable_id =>
composition.id, :salable_type => composition.class.name), :method => :post) %>
        <% end -%></p>
        <p class="artist_name"><%= linked_profiles(composition) %></p>
        <ul class="playlist_social_actions">
          <%= render :partial => "/ratings/rating_list", :locals => {:class_name => composition.class.superclass.name, :ratable =>
composition}%>
          <li>
            <%= link_to_function(image_tag("icon_load_in_player.png", :alt => "Add to Playlist",:title => "Add to Playlist"),
"addCompositionToPlayerPlaylist("#{composition.id}")" %>
          </li>
        </ul>
      </li>
    <% end %>
  <% end -%>
</ol>
```


Do The Math.

60 LOC Partial* 30 = 1800 LOC

24 LOC Partial* 30 = 720 LOC

(that's a whole lotta books)

Didn't even touch:

✱ *Helper methods*

✱ *Presenters*

✱ *Brazillian Jujitsu*

Haml.

(don't use this)

Bake Off!

Haml.

<%= ERB %>

```
#profile
  .left.column
    #date= print_date
    #address=
current_user.address
  .right.column
    #email= current_user.email
    #bio= current_user.bio
```

```
<div id="profile">
  <div class="left column">
    <div id="date"><%= print_date %></div>
    <div id="address">
      <%= current_user.address %></div>
    </div>
  <div class="right column">
    <div id="email">
      <%= current_user.email %></div>
    <div id="bio"><%= current_user.bio %></div>
  </div>
</div>
```


Bake Off!

Haml.

<%= ERB %>

```
#profile
  .left.column
    #date= print_date
    #address=
current_user.address
  .right.column
    #email= current_user.email
    #bio= current_user.bio
```

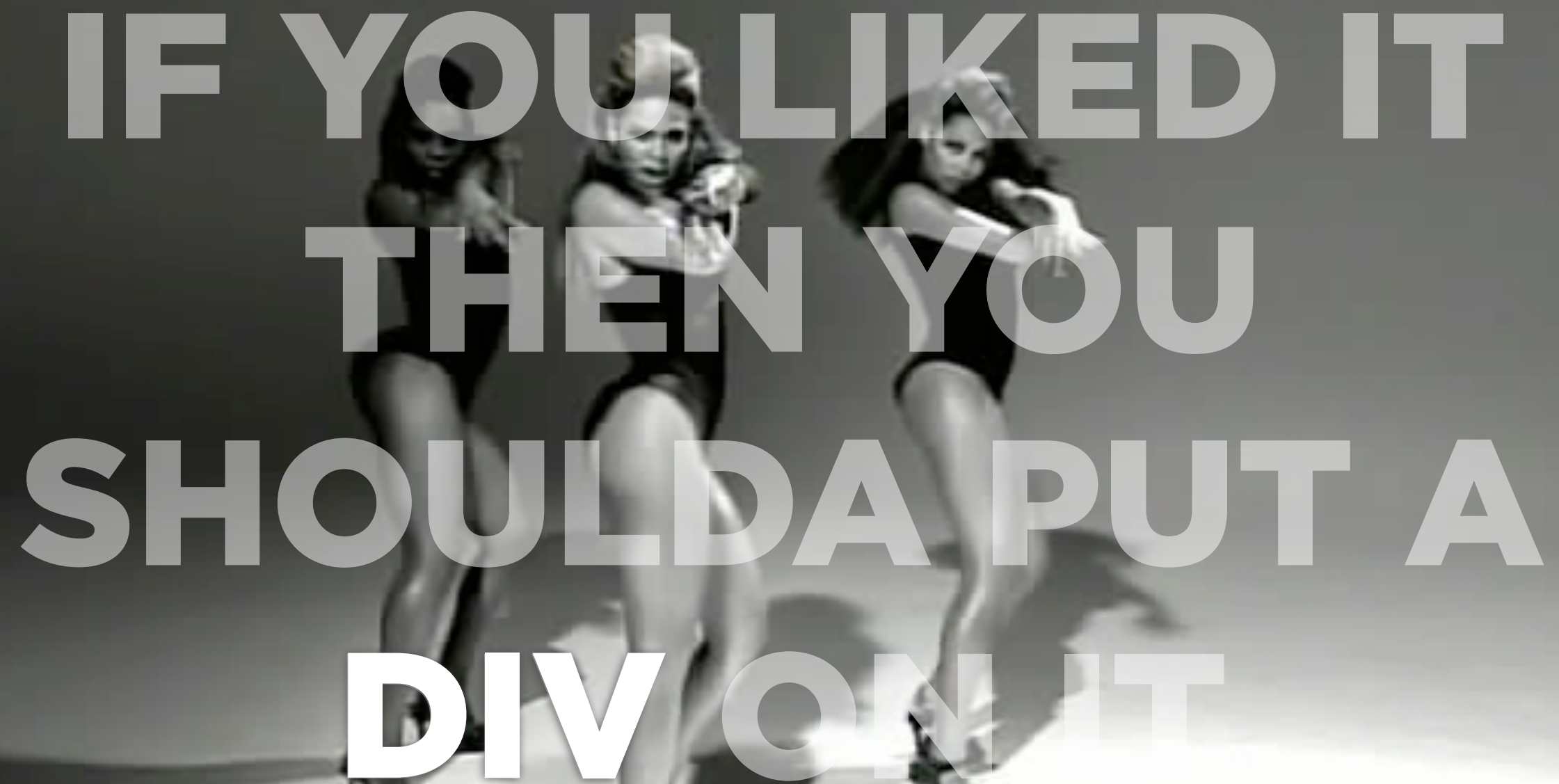
```
<div id="profile">
  <div class="left column">
    <div id="date"><%= print_date %></div>
    <div id="address">
      <%= current_user.address %></div>
    </div>
  <div class="right column">
    <div id="email">
      <%= current_user.email %></div>
    <div id="bio"><%= current_user.bio %></div>
  </div>
</div>
```

DIV

DIVD
DIVD



(OMG.)



IF YOU LIKED IT
THEN YOU
SHOULDA PUT A
DIV ON IT

(oh yes, he went there.)

Revision.

Haml.

(don't use this)

unless you know [^]what you're doing



Don't Hate haml

The tool is not the problem...



**The hand coders
do it too...**

Learn the elements

They aren't scary.

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td

Helping hands

Cheat Sheets!

[http://www.smashingmagazine.com/2010/05/13/
css-2-1-and-css-3-help-cheat-sheets-pdf/](http://www.smashingmagazine.com/2010/05/13/css-2-1-and-css-3-help-cheat-sheets-pdf/)

[http://www.smashingmagazine.com/2009/07/06/
html-5-cheat-sheet-pdf/](http://www.smashingmagazine.com/2009/07/06/html-5-cheat-sheet-pdf/)

HIERARCHY

Where it's at.

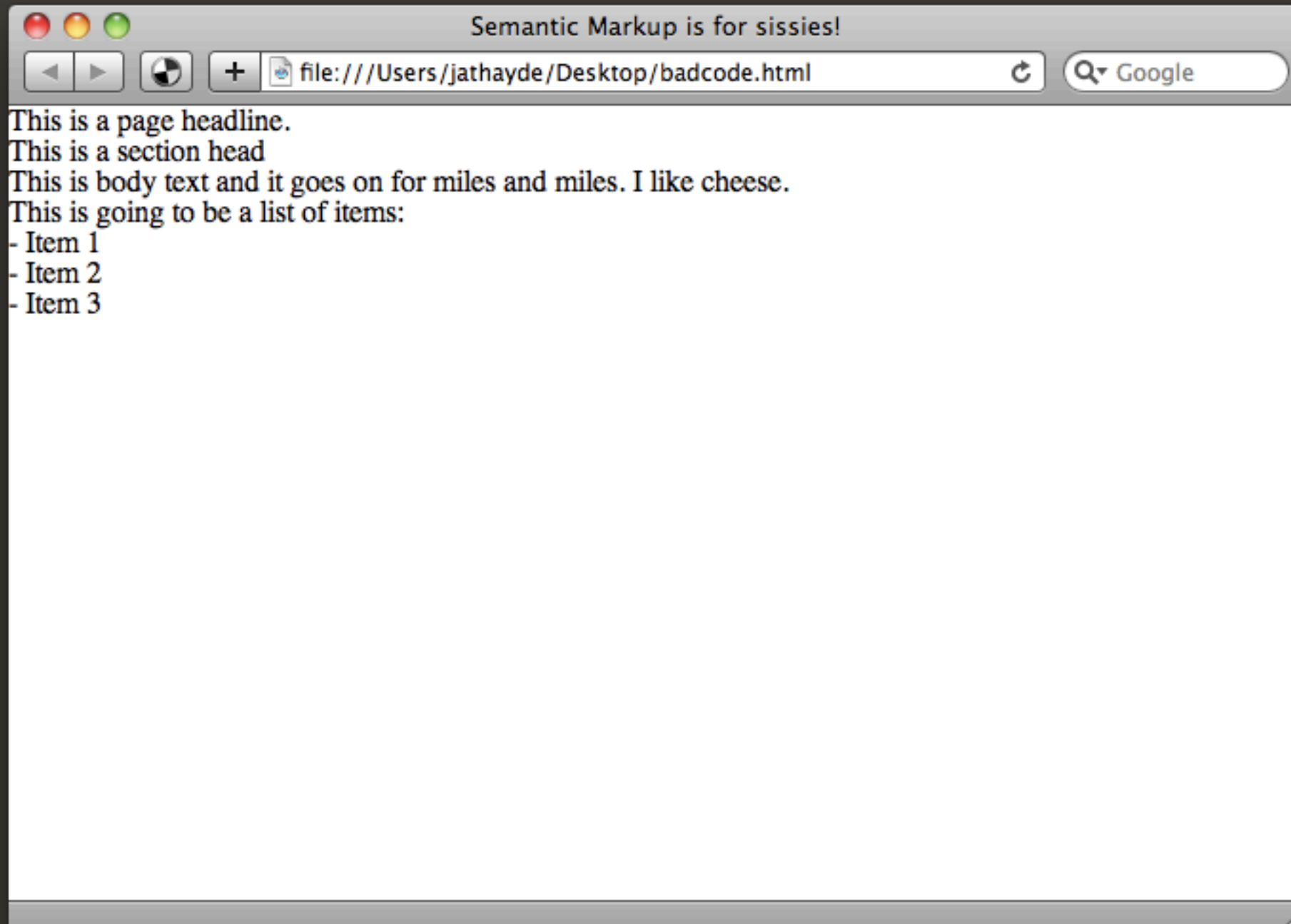
No Hierarchy

Not semantic, everything is the same.

```
<div class="headline">This is a page headline.</div>
<div class="subhead">This is a section head</div>
<div class="body">This is body text and it goes on for
miles and miles. I like cheese.</div>
<div class="list">This is going to be a list of items:<br />
- Item 1<br />
- Item 2<br />
- Item 3<br />
</div>
```

No Hierarchy

Not semantic, everything is the same.



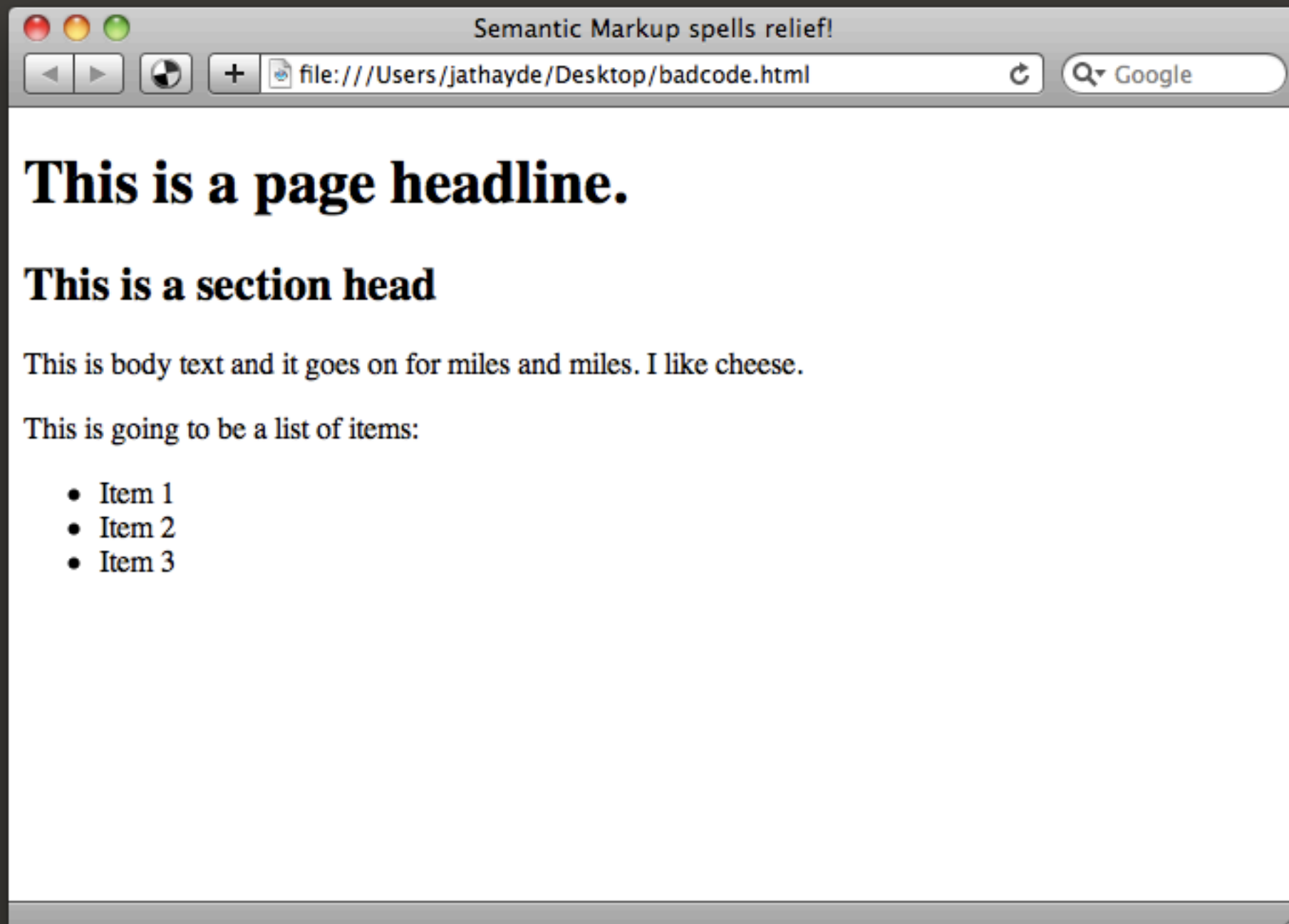
Semantic.

HTML Tags used for meaning

```
<h1>This is a page headline.</h1>
<h2>This is a section head</h2>
<p>This is body text and it goes on for miles and miles.
  I like cheese.</p>
<p>This is going to be a list of items:</p>
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

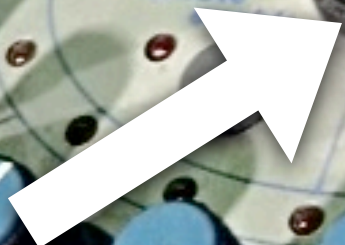
Semantic.

HTML Tags used for meaning



Generated Markup

**THIS ONE
IS DANGEROUS**



div_for?

Don't use this.

```
<%= div_for @product do %>  
<% end %>
```

```
<div id="product123" class="products"> </div>
```


content_tag

But only if you must...

```
<%= content_tag_for(:li, @product, :class => "bar") %>  
<% end %>
```

```
<li id="product_123" class="product bar">
```

```
li.product.bar {}
```

Could be cleaner

Write html.

```
<ul class="products">
  <% @product.each do |product| %>
    <li id="product_<%= @product.id %>" class="bar">
  <% end %>
</ul>
```

```
<ul class="products">
  <li id="product_123" class="bar">
    .
    .
    .
</ul>
```

```
ul.products {}
ul.products li {}
ul.products li.bar {}
```

Helpers

Avoid Markup

```
def full_name(person)
  "<p class='name'>#{@person.first_name} #{@person.last_name}</p>"
end
```

```
def full_name
  "#{@person.first_name} #{@person.last_name}"
end
```

Helpers

Summarization is where they shine.

```
def referrer_for(account)
  if account.referrer
    referral = "Referred_by #{account.referrer.name}"
    if account.web_contact?
      "#{referral} via Web registration"
    else
      referral
    end
  else
    "Web registration"
  end
end
```


HTML IS CODE

Stop having code generate code.

Zen Coding

Use a tool, not a generator

```
div#page>img.logo+ul#navigation>li*5>a
```

```
<div id="page">
  <img src="" class="logo" />
  <ul id="navigation">
    <li><a href=""></a></li>
    <li><a href=""></a></li>
    <li><a href=""></a></li>
    <li><a href=""></a></li>
    <li><a href=""></a></li>
  </ul>
</div>
```

PROJECT FILES:

code.google.com/p/zen-coding/



Why it matters

Not all user agents are created equal...

WEB ACCESSIBILITY

Take care of your users



Google is a blind user.

§508

www.section508.gov

START SMALL

Cover the basics all the time.

WAI Level 1 Checklist:

www.w3.org/TR/WCAG10/full-checklist.html

Web Accessibility

ERB Examples

```
link_to @product.name, product_path(@product), :title =>  
"Take a look at #{@product.name}"
```

```
image_tag "product_12758.png", :alt => "#{@product.name}"
```

```
link_to (image_tag "product_12758.png", :alt => "#{  
#{@product.name}"), product_path(@product), :title => "Take a  
look at #{@product.name}"
```

Web Accessibility
=
Well Formed HTML

Not just an afterthought.



Legacy issues...

from all those classic UI hacks!

Rounded Corners.

Seriously.

```
<div class="b">
  <div class="l">
    <div class="r">
      <div class="bl">
        <div class="br">
          <div class="tl">
            <div class="tr box">
              <%= content %>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Rounded Corners.

“CSS3” The Beast.

```
<div class="box-to-be-rounded">  
  <%= content %>  
</div>
```

```
.box-to-be-rounded {  
  border: 1px solid #ccc;  
  border-radius: 5px; /* IE9, Opera 10.5 */  
  -webkit-border-radius: 5px; /* Safari, Chrome */  
  -moz-border-radius: 5px; /* Firefox */  
}
```

Required Fields

Love the form you're with.

```
<%= f.label :first_name %>  
<span class="required">*</span><br />  
<%= f.text_field :first_name %>
```

```
span.required {color: red;}
```


Required Fields

Progressively enhance!

```
<%= f.label :first_name, :class => "required" %>  
<%= f.text_field :first_name %>
```

```
label { display: block; }  
  
label.required { color: red; }  
  
label.required:after {  
    content: "*";  
}
```

Not supported in MSIE 7 and below,
8 does not accept images for content

REFERENCE:

www.quirksmode.org/css/beforeafter.html

Required Fields

Hackety hacked for MSIE 7

[code redacted to protect the children]

<http://nanobox.chipx86.com/blog/2006/07/before-and-after-in-ie7-and-below.php>



Presenting...

Let's get down to the sexy: Rails Presenters

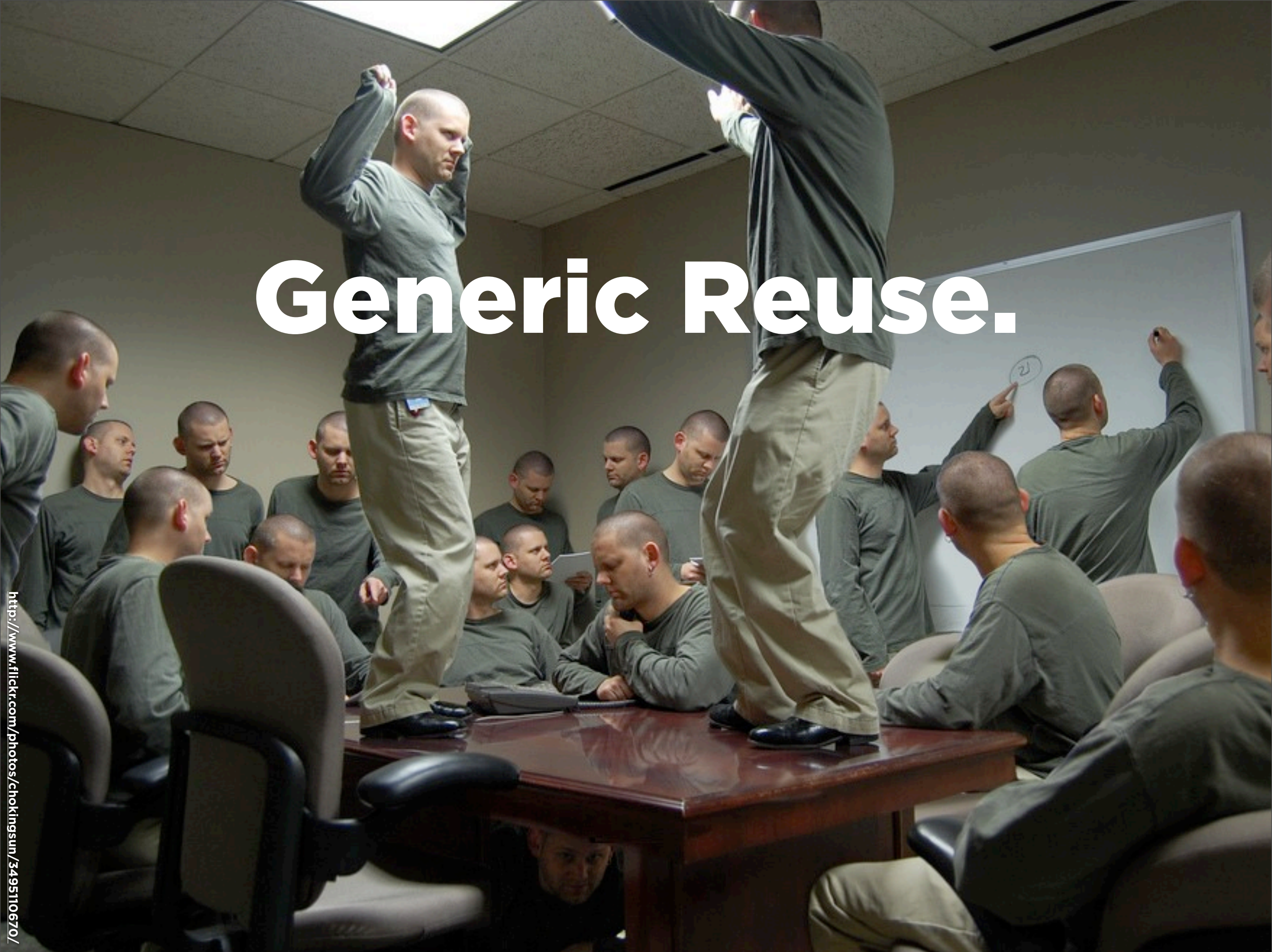
Encapsulation.



<http://www.flickr.com/photos/shuffle-art/2810247509/>

Wednesday, June 9, 2010

Generic Reuse.



<http://www.flickr.com/photos/chokingsun/3495110670/>

A close-up photograph of a pair of hands gently cupping three deep red roses. The roses are in full bloom, with many layers of petals visible. The hands are positioned at the bottom and sides of the roses, with fingers slightly curled. The background is a soft, out-of-focus mix of warm and cool tones, suggesting an indoor setting with natural light. The overall mood is romantic and tender.

Present it.



CSS Tricks...

*The good, bad and ugly of CSS Templates and
Dynamic CSS*

Dynamic CSS



<http://www.flickr.com/photos/shinez/4004798080/>

Why Generate?

There are a few justified cases

- * *You are supporting themes*
- * *You have an insanely large CSS set*
- * *You have an app that you redeploy to many clients (aka, theming)*

Get SASSy

Haml back again

```
!darkgrey = #221f10
```

```
#sidebar  
  background-color: !darkgrey
```

generates...

```
#sidebar {  
  background-color: #221f10;  
}
```


Sprites.

<http://www.flickr.com/photos/gettygirl/3481084323/>

Wednesday, June 9, 2010

Not Very Fairy

reduce the server calls for icons/images



BACKGROUND:

www.alistapart.com/articles/sprites

Sprites in Helpers

Classic like...

```
<%= image_tag ("email.gif"), :class => "email" %>
```

```
<%= image_sprite :email, :class => "email", :title => "Email" %>
```

DETAILS:

37signals.com/svn/posts/2336-using-css-sprites-with-rails-helper-methods

```

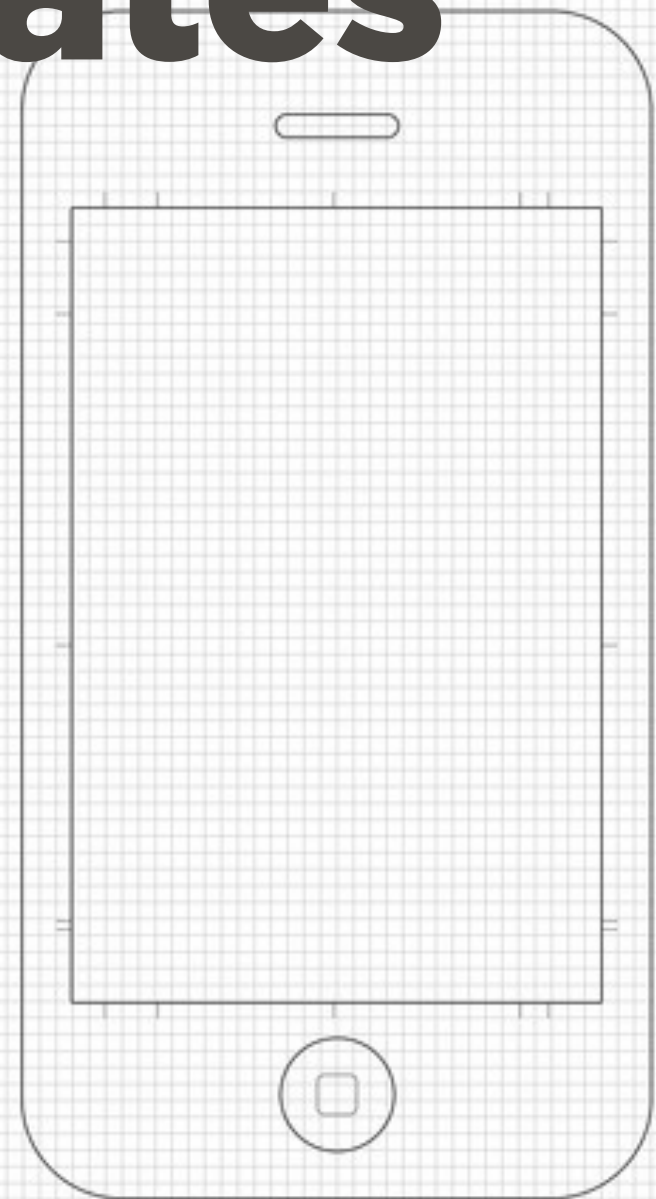
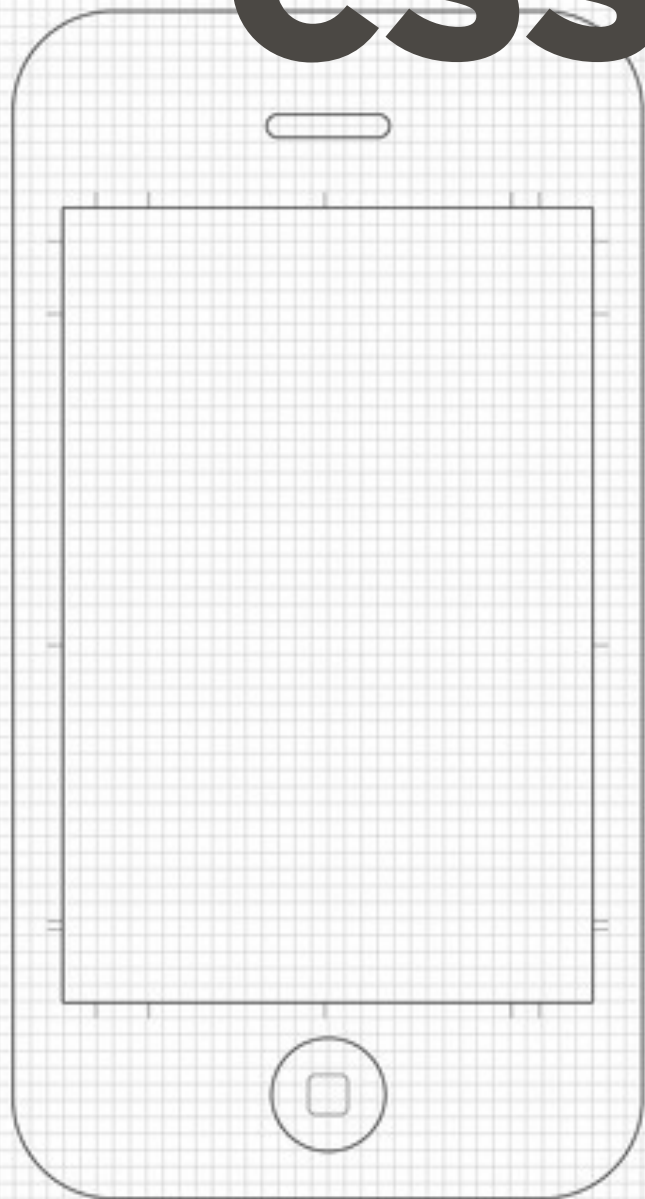
def image_sprite(image, options = {})
  sprites = {
    :add_icon      => {:w => 16, :h => 16, :x => 0, :y => 0},
    :email         => {:w => 26, :h => 16, :x => 41, :y => 0},
    :print         => {:w => 25, :h => 17, :x => 68, :y => 0},
    :trash         => {:w => 10, :h => 11, :x => 94, :y => 0},
    :comments      => {:w => 13, :h => 13, :x => 105, :y => 0},
    :comments_read => {:w => 13, :h => 13, :x => 120, :y => 0},
    :comments_unread => {:w => 13, :h => 13, :x => 135, :y => 0},
    :rss           => {:w => 14, :h => 14, :x => 150, :y => 0},
    :ical          => {:w => 14, :h => 16, :x => 166, :y => 0},
    :drag          => {:w => 11, :h => 11, :x => 360, :y => 0},
    :timeclock     => {:w => 17, :h => 17, :x => 375, :y => 0},
    :timeclock_off => {:w => 17, :h => 17, :x => 392, :y => 0}
  }
  %(<span class="sprite #{options[:class]}" style="background: url("#{path_to_image
    ('/images/basecamp_sprites.png')) no-repeat -#{sprites[image][:x]}px -#{sprites
    [image][:y]}px; width: #{sprites[image][:w]}px; padding-top: #{sprites[image][:h]}px;
    #{options[:style]}" title="#{options[:title]}">#{options[:title]}</span>)
end

```

DETAILS:

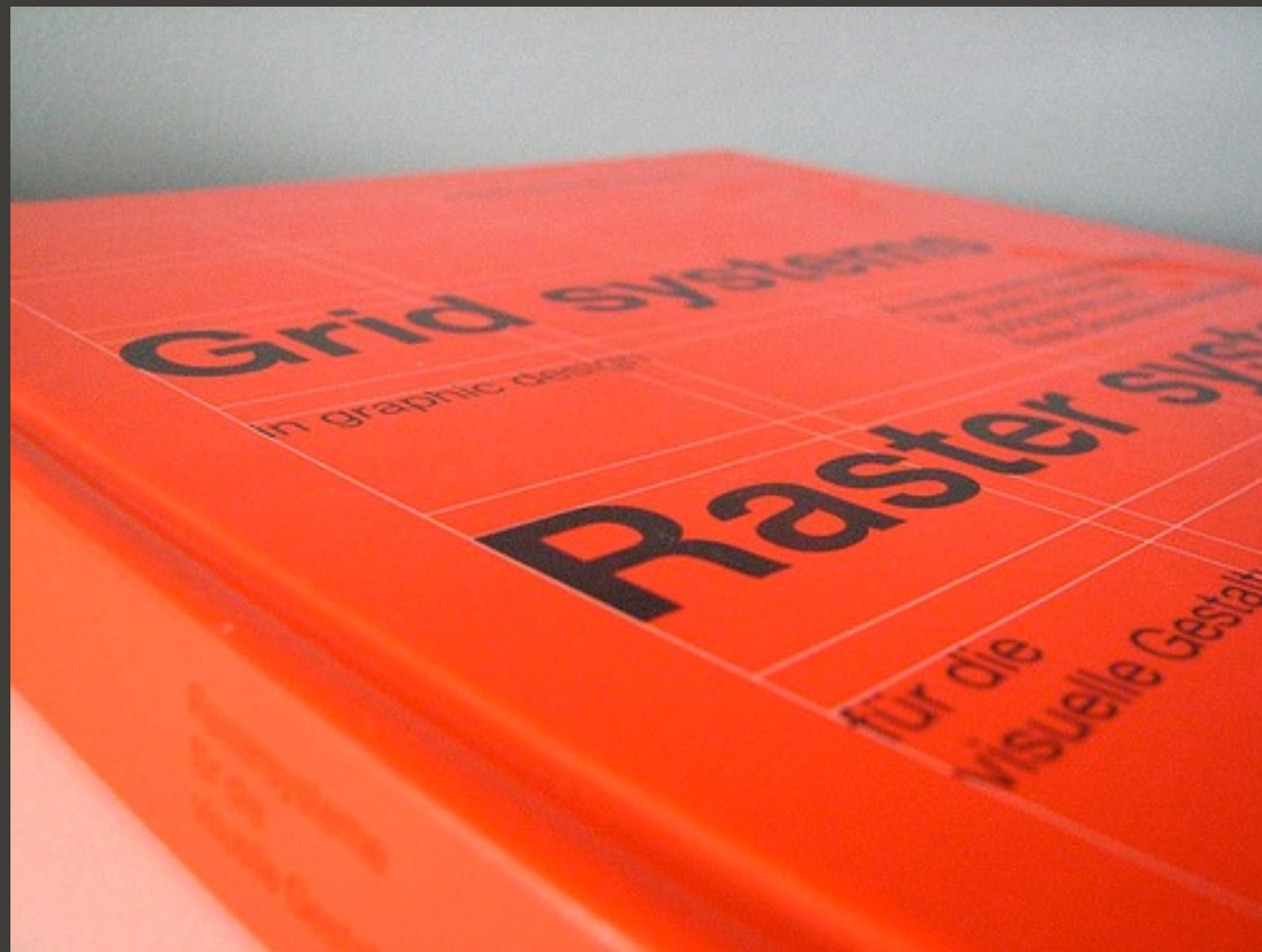
37signals.com/svn/posts/2336-using-css-sprites-with-rails-helper-methods

CSS Templates



Why?

Derived from the print world...



Sketches for a grid with 32 fields

Skizzen für einen Raster mit 32 Feldern



In sketching a grid, care must be taken to ensure that the sketch corresponds as closely as possible to the proportions of the final printed format. Particulars regarding the text columns and lines should be as precise as they can be. A grid sketch can look like a stepped-down reproduction in print. This implies that the sketched lines representing lines of print and the outlines of the picture formats should be approximately proportional to the final printed work. Such a sketch should enable some idea to be formed of the quantity of text, the size of the type, the length of the

lines and how much of the grid is type matter and how much picture. The designer can easily account to the supervisor for indicate the type with excessively heavy and imprecise strokes of the pencil which give no indication of the size of the letters and the width of the text columns. It needs concentration and sensitivity and also a certain amount of practice, before the sketch begins to look like a precise original from which the printer can work. The sketches shown are 6.5 cm high in the original, i. e. only a few mm larger than the print.

Beim Skizzieren eines Rasters muss darauf geachtet werden, dass die Skizze möglichst genau der Proportion des endgültigen Druckformates entspricht. Die Angaben für die Textspalten und Textzeilen sollten so genau als möglich sein. Eine gute Skizze kann wie eine verkleinerte Druckvorlage wirken. Das bedeutet, dass die skizzierten Linien für die Zeilen und die Umrisse für die Bildformate in der angegebenen Proportion zur definitiven Druckmaske stehen sollten. Eine solche Skizze vermittelt bereits die Vorstellung der Menge Text, der Größe der Schrift, der Länge der Zeilen

und vom Größenverhältnis von Schrift zu Bildfeld zu geben. Der Skizze ist nicht versucht, die Schrift mit zu schweren und häufigen Strichen anzudeuten, die keinen Eindruck der Größe der Buchstaben und der Breite der Textspalten zulassen. Es braucht Konzentration und Sensibilität, aber auch eine gewisse Übung dazu, bis die Skizze den Eindruck der präzisen Druckvorlage liefert. Die abgebildeten Skizzen sind im Original 6,5 cm hoch, d. h. um einige Millimeter größer als sie hier im Druck erscheinen, gezeichnet.

34

Sketches for a grid with 32 fields

Skizzen für einen Raster mit 32 Feldern



The sketches show only a few of the many possibilities of designing with a 32-field grid. Their purpose is to indicate how the sketches should be drawn so as to enable a picture to be formed of the final result. The more accurate the sketch, the more difficult it is to imagine what the finished printed work will look like. The more accurate the sketch, the easier it will be for the designer to test the idea behind it for suitability. There are designers who can produce, even for printers, the exact sketches which represent text columns quite clearly what the result will look like in print.

Skizzen, die hier nur ein paar Beispiele zeigen, sind natürlich weit entfernt von der endgültigen Lösung. Der Designer sollte sich nicht nur auf die Skizzen verlassen, sondern auch die endgültige Druckvorlage prüfen. Ein Designer, der es versteht, kann schon bei der Skizze die endgültige Druckvorlage im Kopf haben. Es gibt Designer, die es verstehen, schon bei der Skizze die endgültige Druckvorlage im Kopf zu haben. Es gibt Designer, die es verstehen, schon bei der Skizze die endgültige Druckvorlage im Kopf zu haben.

Die Skizzen zeigen nur einige wenige Möglichkeiten der Gestaltung mit 32 Feldern. Die Aufgabe des Designers ist es, die Skizzen so zu zeichnen, dass sie den Eindruck der endgültigen Druckvorlage vermitteln. Je genauer die Skizze angefertigt wird, desto schwieriger wird es, sich eine Vorstellung davon zu machen, wie die Druckmaske schließlich aussehen könnte. Je präziser die Skizze ist, desto leichter wird es dem Designer möglich, die dahinter liegenden Ideen auf ihre Verwendbarkeit zu prüfen. Es gibt Designer, die es verstehen,

schon bei der Skizze die endgültige Druckvorlage im Kopf zu haben. Es gibt Designer, die es verstehen, schon bei der Skizze die endgültige Druckvorlage im Kopf zu haben. Es gibt Designer, die es verstehen, schon bei der Skizze die endgültige Druckvorlage im Kopf zu haben. Es gibt Designer, die es verstehen, schon bei der Skizze die endgültige Druckvorlage im Kopf zu haben.

35

960.gs

Not Semantic

```
.container_12 .grid_1{  
    width:60px;  
}  
  
.container_12 .grid_2{  
    width:140px;  
}  
  
.container_12 .grid_3{  
    width:220px;  
}  
  
.container_12 .grid_4{  
    width:300px;  
}  
  
.container_12 .grid_5{  
    width:380px;  
}  
  
.container_12 .grid_6{  
    width:460px;  
}
```

Survey says...



Toupee.

LAYERED APPROACH

Turn it on and off

REFERENCE:

github.com/marchdoe/toupee



Quick Starts...

Saving time with a gem generator to get up and running with a basic setup

ONE FILE

to bring them all and on the server bind them.



Put that in your pipe and smoke it!

Calling All CSS

Classic like...

app/layouts/application.html.erb

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

  <title><%= APP_CONFIG[:name] %></title>

  <!-- The first style sheet calls the rest -->
  <%= stylesheet_link_tag 'application.css', :media => 'screen, projector' %>
```

public/stylesheets/application.css

```
/*
Title:    Screen styles and MSIE patches
Author:   John Athayde
*/

/* import stylesheets and hide from IE/Mac */
@import url("reset.css") screen;
@import url("master.css") screen;
@import url("print.css") print;

/* Plugin stylesheets */
@import url("../javascripts/jquery-ui-1.8.1/css/cupertino/jquery-ui-1.8.1.custom.css") screen;

/* If there are any styles that we want IE5/Mac to see, we put them after this */
```

My Base App Setup

Rails 3 app w/CSS, Javascript, MSIE hacks, etc.

✱ *MSIE 6/7 Specific styles*

✱ *MSIE 24 bit PNG fixes*

✱ *Clearfix*

✱ *Custom Gems (Deadweight/Startups)*

CSS STARTUP:

www.github.com/boboroshi/infoether-framework

A more robust method

that covers javascript too...

JAMMIT

“Jammit is an industrial strength asset packaging library for **Rails**, providing both the CSS and JavaScript concatenation and compression that you’d expect, as well as **YUI Compressor** and **Closure Compiler** compatibility, ahead-of-time gzipping, built-in JavaScript template support, and optional **Data-URI / MHTML image and font embedding**.”

JAMMIT INFO:
documentcloud.github.com/jammit



ENTERING
Startup

STARTUPS GEM:

www.github.com/infoether/startups



Clean up that mess

Code audits, unused CSS and moving to production

Organize it!

Standardize what you do as a team.

```
input,  
textarea {  
  border-radius: 3px; /* IE9, Opera 10.5 */  
  -webkit-border-radius: 3px; /* Safari, Chrome */  
  -moz-border-radius: 3px; /* Firefox */  
  font-size: 1.4em;  
  padding: 5px;  
}
```

*Alphabetical, with vendor specific switches
kept next to official attributes.*

Unused Selectors

UI side: Dust Me Selectors

View saved data

View data for: infoether.com

Unused selectors Used selectors Spider log

http://infoether.com/stylesheets/reset.css?1274119959

58 unused selectors:

http://infoether.com/stylesheets/master.css?1274119959

55 unused selectors:

ul#nav li a#products	Line 74
ul#nav li a#media	Line 78
body#team ul#nav li a#team	Line 83
body#products ul#nav li a#products	Line 84
body#articles ul#nav li a#blog	Line 85
body#consulting ul#nav li a#consulting	Line 86
body#clients ul#nav li a#clients	Line 87
body#media ul#nav li a#media	Line 88
body#training ul#nav li a#training	Line 89
body#contact ul#nav li a#contact	Line 90

Found 117 unused selectors among 3 stylesheets

Save as CSV... Clear this data OK

InfoEther | http://infoether.com/

HOME TEAM BLOG CONSULTING CLIENTS TRAINING CONTACT

RICH SAYS: Check out my [HotCocoa presentation](#) from RubyConf 2008

INFOETHER PROUDLY PRESENTS:

RUBY & RAILS ECOSYSTEM WHITE PAPER

Consulting
IMPLEMENTATION
ACTIVITY
Deployment

Months of research pulled together to give you a detailed view of the ecosystem, from top to bottom.

READ NOW >>

FIND OUT WHERE WE ARE
SPEAKING AND PRESENTING
WE SHOW UP ALL OVER THE WORLD

Meet the InfoEther Team Members at
10, 2010
MD
RAILSConf

FIREFOX ADD-ON:
www.sitepoint.com/dustmeselectors

Unused Selectors

Gem: Deadweight

```
# lib/tasks/deadweight.rake

require 'deadweight'

Deadweight::RakeTask.new do |dw|
  dw.mechanize = true

  dw.root = 'http://staging.example.com'

  dw.stylesheets = %w( /stylesheets/style.css )

  dw.pages = %w( / /page/1 /about )

  dw.pages << proc {
    fetch('/login')
    form = agent.page.forms.first
    form.username = 'username'
    form.password = 'password'
    agent.submit(form)
    fetch('/secret-page')
  }

  dw.ignore_selectors = /hover|lightbox|superimposed_kittens/
end
```

GEM REPOSITORY:

www.github.com/aanand/deadweight



TESTS

Don't write tests around markup if you don't have to.



TIDY

Because rendered code is always slightly messy.

Final delivery...

Rack::Tidy

```
sudo gem install rack-tidy
```

config/environment.rb

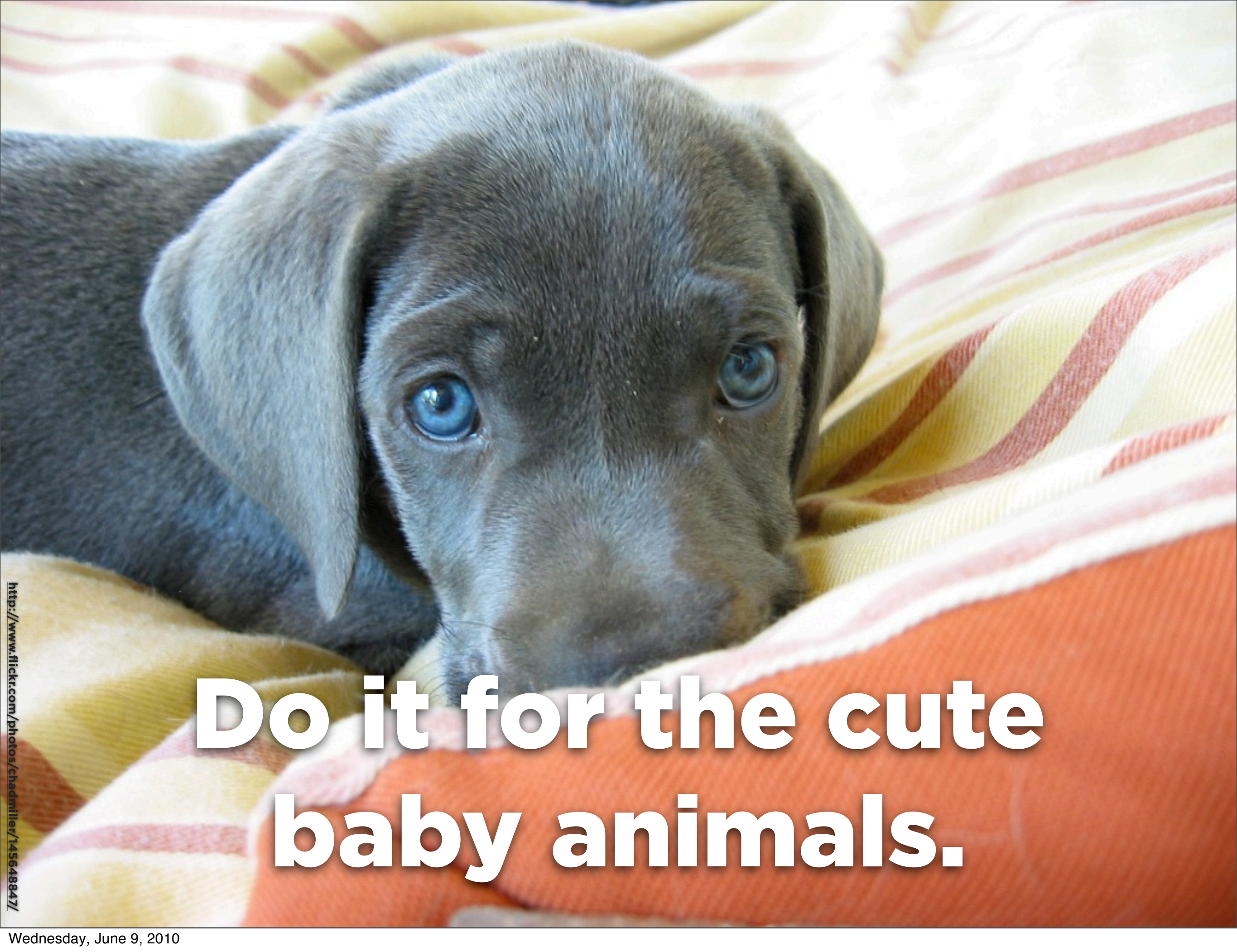
```
# above Rails::Initializer block  
require 'rack/tidy'  
  
# inside Rails::Initializer block  
config.middleware.use Rack::Tidy
```

Rack::Tidy Info:

coderrack.org/users/webficient/middlewares/38-racktidy

TIDY Info:

www.w3.org/People/Raggett/tidy/
tidy.sourceforge.net/docs/quickref.html



**Do it for the cute
baby animals.**

Q&A

Fire when ready.



THANKS!

infoether

SLIDES:

www.infoether.com/railsconf2010/

CSS STARTUP:

www.github.com/boboroshi/infoether-framework

TWITTER:

[@boboroshi](https://twitter.com/boboroshi)

STARTUPS GEM:

www.github.com/infoether/startups