the ASSET PIPELINE and our POST-MODERN, HYBRID, JavaScript Future
THE PROBLEM
THE PAIN
account_invoice.js
application.js
autolabel.js
calendar_date.js
calendar_date_format.js
calendar_date_select.js
categories.js
color.js
color_picker.js
color_scheme.js
comments.js
common.js
companies.js
custom.js
date.js
easing.js
fancyzoom.js
files.js
flash_support.js
layout.js
locations.js
login.js
milestone_quick_window.js
milestone_templates.js
milestone_writing_activity.js
milestones.js
move_operation.js
msg.js
nubbins.js
participants.js
pending_attachments.js
people.js
privacy.js
project_templates.js
search.js
settings.js
templates.js
time.js
todo_dates.js
todo_shortcuts.js
todos.js
touch_support.js
upload_debugging.js
upload_preference.js
widget.js
writeboards.js
Envy27:javascripts david$ ls
account_invoice.js
application.js
autolabel.js
calendar_date.js
calendar_date_format.js
calendar_date_select.js
categories.js
color.js
color_picker.js
color_scheme.js
comments.js
common.js
companies.js
custom.js
date.js
easing.js
fancyzoom.js
files.js
flash_support.js
layout.js
locations.js
login.js
milestone_quick_window.js
milestone_templates.js
milestone_writing_activity.js
milestones.js
move_operation.js
msg.js
nubbins.js
participants.js
pending_attachments.js
people.js
privacy.js
project_templates.js
search.js
settings.js
templates.js
time.js
todo_dates.js
todo_shortcuts.js
todos.js
touch_support.js
upload_debugging.js
upload_preference.js
widget.js
writeboards.js
>13 =
>13 =
class PeopleController
  def index() end
  def show() end
  def create() end
  def edit() end
  def update() end
  def destroy() end
end
Thursday, May 19, 2011

```
return $('#remove_ + id).style.display !== "none"
},

before: function(id) {
    Element.show("remove_spinner_" + id)
},

failure: function(request, id) {
    Element.hide("remove_spinner_" + id)
    alert("An error prevented the writeboard being deleted. Please try again.")
},

success: function(request, id) {
    new Effect.Fade("writeboard_" + id, {
        afterFinish: function() {
            Element.remove("writeboard_" + id)
            writeboards.hideQuotaBlock()
            writeboards.checkEmpty() }})
},

privacy: {
    toggle: function(id) {
        this.visible(id) ? this.hide(id) : this.show(id)
    },

    show: function(id) {
        Element.hideIf('remove_' + id, 'show_' + id)
        new Effect.BlindDown('privacy_' + id, {duration: 0.2})
    },

    hide: function(id) {
        new Effect.BlindUp('privacy_' + id, {duration: 0.2,
            afterFinish: function() {
                Element.show('show_' + id) }})
    },

    visible: function(id) {
        return $('#privacy_' + id).style.display !== "none"
    },

    before: function(id) {
        Element.show("privacy_spinner_" + id)
        Element.hide("public_" + id)
    },

    failure: function(request, id) {
        Element.hide("privacy_spinner_" + id)
        Element.show("public_" + id)
        $('#public_' + id).checked = !$('#public_' + id).checked
        alert("An error prevented your request from completing. Please try again.")
    }
```
<% javascript_include_tag :all, :cache => true %>
People do what's easy
THE SOLUTION
INNOVATION
models
module ApplicationHelper
end
A place for everything and everything in its place
app/assets, lib/assets, vendor/assets

images  stylesheets  javascripts
config.assets.paths << "disneyland"

# GET /assets/goofy.gif
# => disneyland/goofy.gif
# signal_id.gem

$ cat /app/assets/javascripts/signal_id/index.js

//= require ./identity_validation
//= require ./launchbar
//= require ./toggle_credentials

# basecamp

$ cat /app/assets/javascripts/application.js

//= require signal_id
The best way to manage your application’s dependencies

Bundler
Copying assets
# jquery-rails.gem

$ ls vendor/assets/javascripts
jquery-ui.js  jquery.js  jquery_ujs.js

$ rails new future
$ cat app/assets/javascripts/application.js

//= require jquery
//= require jquery_ujs
//= require_tree .
# gem 'jquery-rails'
gem 'prototype-rails'
Blueprint is a **CSS framework**, which aims to cut down on your development time. It gives you a solid foundation to build your project on top of, with an easy-to-use grid, sensible typography, useful plugins, and even a stylesheet for printing.
THE PLEASURE
Include CoffeeScript in Gemfile

josh (author)
April 13, 2011

Showing 1 changed file with 4 additions and 2 deletions.

railties/lib/rails/generators/app_base.rb
Include CoffeeScript in Gemfile

josh (author)
April 13, 2011

Showing 1 changed file with 4 additions and 2 deletions.

railties/lib/rails/generators/app_base.rb
@perfectline I agree.

Coffeescript is a *toy*, it is nothing more than an aggregation of features from a few languages, and in fact complicates an extremely simplistic language. If I wrote a layer on top of Ruby, took out the `end` keywords etc, it would be a huuuuuugely newbish thing to do, however it seems that this is "the thing to do" with JavaScript. So lame
This is not a sane default by any stretch of the imagination. Just because 37signals built a webserver with coffeescript does not mean everyone wants it to be the default. There's something to be said for leading people in the right direction a la the switch to bundler, and it's another thing entirely to pick a ridiculous default that nobody uses. Thanks but no thanks.
augustl added a note to 9f09aeb

I smell a fork.

(I wish I could be constructive, but CoffeScript makes me nerdrage.)
RAILS DEFAULT CHANGES

FREAK THE FUCK OUT

meme generator.net
COFFEESCRIPT BY DEFAULT ON RAILS

NOT FUNNY DHH
MAKES COFFEESCRIPT DEFAULT

DOESN'T INCLUDE HAML
“CoffeeScript is well done and more convenient to use than JavaScript”
“CoffeeScript is well done and more convenient to use than JavaScript”

- Brendan Eich, inventor of JavaScript
“We should continue to grow the language, keeping support for old forms while adding new forms to help users themselves grow the language”
Active Support
$(document).bind 'pageUpdated', ->
$('li[data-subscriber-id]').each ->
  if $(this).attr('data-subscriber-id') is $.cookie('user_id')
    $(this).hide()
$(document).bind('pageUpdated', ->
    $('li[data-subscriber-id]').each ->
        if $(this).attr('data-subscriber-id') is $.cookie('user_id')
            $(this).hide()\

$(document).bind('pageUpdated', function() {
    return $('li[data-subscriber-id]').each(function() {
        if ($(this).attr('data-subscriber-id') === $.cookie('user_id')) {
            return $(this).hide();
        }
    });
});
for filename in list
do (filename) ->
    fs.readFile filename, (err, contents) ->
    compile filename, contents.toString()
var filename, _fn, _i, _len;
_fn = function(filename) {
    return fs.readFile(filename, function(err, contents) {
        return compile(filename, contents.toString());
    });
};
for (_i = 0, _len = list.length; _i < _len; _i++) {
    filename = list[_i];
    _fn(filename);
}
launch() if ignition is on

volume = 10 if band isn't SpinalTap

letTheWildRumpusBegin() unless answer is no

if car.speed < limit then accelerate()

winner = yes if pick in [47, 92, 13]
class Animal
constructor: (@name) ->

move: (meters) ->
    alert @name + " moved " + meters + " m."
END-TO-END COFFEEESCRIP

10:45am Wednesday, 05/18/2011

General

Location: Room 347

• Presented by Trevor Burnham
Haml.

{markup haiku}
Haml.
{markup haiku}

Sass.
{style with attitude}
Requiring Haml to use SCSS feels like inviting somebody you don't like over to your house because they have a cute friend.
@rjs
Ryan Singer

Requiring Haml to use SCSS feels like inviting somebody you don't like over to your house because they have a cute friend.
Requiring Haml to use SCSS feels like inviting somebody you don't like over to your house because they have a cute friend.

I hope the haml topic didn't overshadow the important points of the talk. Thanks for the feedback so far. #wcr10
Nesting

```scss
// SCSS //

li {
  font: {
    family: serif;
    weight: bold;
    size: 1.2em;
  }
}

table.hl {
  margin: 2em 0;
  td.ln {
    text-align: right;
  }
}

/* CSS */

li {
  font-family: serif;
  font-weight: bold;
  font-size: 1.2em;
}

table.hl {
  margin: 2em 0;
}

table.hl td.ln {
  text-align: right;
}
```
Variables

```scss
$.scs, $.sass

$blue: #3bbfce;
$margin: 16px;

.content-navigation {
  border-color: $blue;
  color: $blue;
  background-color: $blue;
}

.border {
  padding: $margin / 2;
  margin: $margin / 2;
  border-color: $blue;
}

/* CSS */

.content-navigation {
  border-color: #3bbfce;
  color: #2b9eab;
}

.border {
  padding: 8px;
  margin: 8px;
  border-color: #3bbfce;
}
SASS: THE FUTURE OF STYLESHEETS

10:45am  Wednesday, 05/18/2011

General

Location: Ballroom II

• Presented by Chris Eppstein
$ rails g resource post
invoke assets
create  app/assets/javascripts/posts.js.coffee
create  app/assets/stylesheets/posts.css.scss
app/assets/stylesheets/posts.css.scss.erb
app/assets/stylesheets/posts.css.scss.erb

```css
a.delete {
  width: 10px;
  background: url(<%= asset_path "trash.gif" %>);
}
```
The dynamic stylesheet language.

Write some LESS:

```less
.box-shadow (@x: 0, @y: 0, @blur: 1px, @alpha) {
    @val: @x @y @blur rgba(0, 0, 0, @alpha);
    box-shadow: @val;
    -webkit-box-shadow: @val;
    -moz-box-shadow: @val;
}
.box { @base: #f938ab;
    color: saturate(@base, 5%);
    border-color: lighten(@base, 30%);
    div { .box-shadow(0, 0, 5px, 0.4) }
}
```
# Asset template engines

gem 'sass'
gem 'coffee-script'
# Asset template engines
# gem 'sass'
# gem 'coffee-script'
THE SCALES
$ rake assets:precompile
# public/assets/application.js
# public/assets/application.css
# public/assets/*.(gif|jpg|png)
$ ls public/assets/
application-109dccf4633654a1ed7232bf4faf1e23.css
application-2e8ef48961e28640c5b6e4e7c0388d16.js
<%= javascript_include_tag "application" %>
<script src="/assets/application-2e8ef48961e28640c5b6e4e7c0388d16.js" type="text/javascript"></script>
✓ File-based, not params-based
✓ MD5, not timestamp
# Compress both stylesheets and JavaScripts
config.assets.js_compressor = :uglify
config.assets.css_compressor = :scss
SPROCKETS v2

helps you turn messy JavaScript into clean modules for development and a single file for deployment.
SPROCKETS v2

helps you turn messy JavaScript into clean modules for development and a single file for deployment.
But when can I get my hands on this amazing technology?!?
Rails 3.1: Release Candidate THIS WEEK!
1000 projects in your pocket. Basecamp is now on your mobile phone.

No apps required. Simply visit basecamphq.com on your phone’s browser, and you’re good to go!
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, "127.0.0.1");
console.log('Server running at http://127.0.0.1:1337/');
All JavaScript
All the time
“A Modernist has to decide whether this is true OR that is true. The Modernist believes in OR more than AND. Postmodernists believe in AND more than OR.”

- Larry Wall
It's 02:32:47 PM

- home
- dinosaurs
- aliens

Pjax loads html from your server into the current page without a full page load. It's ajax with real permalinks, page titles, and a working back button that fully degrades.

Check the box to toggle pjax.

Whenever the time changes, a full page load has happened. If the time doesn't change, no full page load has occurred.

The idea is you can't tell the difference between pjax page loads and normal page loads. On complicated sites, browsing just "feels faster."

view this example's source code
Backbone supplies structure to JavaScript-heavy applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions, views with declarative event handling, and connects it all to your existing application over a RESTful JSON interface.
<% if @projects.length: %>
  <% for project in @projects: %>
    <a href="<%= project.url %>"><%= project.name %></a>
    <p><%= project.description %></p>
  <% end %>
<% else: %>
  No projects
<% end %>
<% form_for Comment.new, remote: 'true' do |form| %>

# Controller
def create
  @comment = @project.comments.create params[:comment]
end

# View/response
$('comments.new').before('<%=j render @comment %>');
$('comments.new textarea').val('');
the ASSET PIPELINE and our POST-MODERN, HYBRID, Javascript Future