Introducing *CoffeeScript*

Mattt Thompson (***@mattt***)
Austin on Rails, May 2011

3.1

Sass

CoffeeScript

Sass → CSS

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

.content-navigation
  border-color: $blue
  color: darken($blue, 9%)

.border
  padding: $margin / 2
  margin: $margin / 2
  border-color: $blue
```

```
.content-navigation {
  border-color: #3bbfce;
  color: #2ca2af; }

.border {
  padding: 8px;
  margin: 8px;
  border-color: #3bbfce; }
```

CoffeeScript → JavaScript

```coffeescript
Account = (customer, cart) ->
  @customer = customer
  @cart = cart

  $('.shopping_cart').bind ↩
'click', (event) =>
    @customer.purchase @cart
```

```javascript
var Account;
var __bind = function(fn, me)
{ return function(){ return
fn.apply(me, arguments); }; };
Account = function(customer, cart)
{
  this.customer = customer;
  this.cart = cart;
  return $('.shopping_cart').bind
('click', __bind(function(event) {
    return this.customer.purchase
(this.cart);
  }, this));
};
```

Introducing CoffeeScript

What We'll Cover In This Talk

- Installation
- Integration With Rails
- Syntax
- Features

Installation

CoffeeScript

"It's just Javascript"

- *Less Syntactic Cruft*
- *JavaScript Design Patterns are Features of CoffeeScript*

node.JS

Evented I/O for V8 JavaScript.

An example of a web server written in Node which responds with "Hello World" for every request.

```
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/');
```

To run the server, put the code into a file `example.js` and execute it with the node program:

```
% node example.js
Server running at http://127.0.0.1:1337/
```

Homebrew

The missing package manager for OS X

Homebrew is the easiest and most flexible way to install the UNIX tools Apple didn't include with OS X.

```
$ brew install wget
```

Packages are installed into their own isolated prefixes and then symlinked into `/usr/local`.

```
$ cd /usr/local
$ find Cellar
Cellar/wget/1.12
Cellar/wget/1.12/bin/wget
Cellar/wget/1.12/share/man/man1/wget.1

$ ls -l bin
bin/wget -> ../Cellar/wget/1.12/bin/wget
```

Just extract the tarball and straight away you have a working Homebrew installation.

Install Homebrew Today!

Create new Homebrew packages in seconds.

```
$ brew create http://foo.com/bar-1.0.tgz
Created /usr/local/Library/Formula/bar.rb
```

Easily adapt Homebrew formula to your needs. And since it's all Git underneath your changes are merged automatically with upstream updates.

```
$ brew edit wget # opens in TextMate!
```

Homebrew formula are simple Ruby scripts:

```
require 'formula'

class Wget < Formula
  homepage 'http://www.gnu.org/wget/'
  url 'http://ftp.gnu.org/wget-1.12.tar.gz'
  md5 '308a5476fc096a8a525d07279a6f6aa3'

  def install
```

```
$ homebrew install node
```

npm is a package manager for node. You can use it to install and publish your node programs. It manages dependencies and does other cool stuff.

One Line Install

```
curl http://npmjs.org/install.sh | sh
```

More Than One Line Install

1. Get the code.
2. Do what the README says to do.

Other Cool Stuff

- README
- FAQ
- Search for Packages
- Mailing List
- Bugs

```
$ curl http://npmjs.org/install.sh | sh
```

```
$ npm install -g coffee-script
```

$ coffee

jashkenas / coffee-script-tmbundle

Watch Fork 219 34

Source Commits Network Pull Requests (3) Issues (6) Graphs Branch: master

Switch Branches (1) Switch Tags (0) Branch List

A TextMate Bundle for CoffeeScript — Read more

Downloads

HTTP Git Read-Only https://github.com/jashkenas/coffee-script-tmbu Read-Only access

fixing herecomment highlighting.

jashkenas (author)
1 day ago

commit 24589a08cbc0a7cb03e8
tree f0ec7c57ac224d12dd7a
parent bacb597254a9ca0cc4ac

coffee-script-tmbundle /

name	age	message	history
Commands/	May 09, 2011	merging #27 [jashkenas]	
Preferences/	May 08, 2011	Issue #13, indendation rules for function arrows. [jashkenas]	
Snippets/	December 23, 2010	Issue #14 ... removing old colon-style assignment. [jashkenas]	
Syntaxes/	1 day ago	fixing herecomment highlighting. [jashkenas]	
README.markdown	December 05, 2010	Issue #11. TextMate/PATH [jashkenas]	
info.plist	May 09, 2011	Structured the items in the menu so you can find w... [mads379]	

README.markdown

CoffeeScript.tmbundle

A **TextMate Bundle** for the **CoffeeScript** programming language.

Installation:

```
mkdir -p ~/Library/Application\ Support/TextMate/Bundles
cd ~/Library/Application\ Support/TextMate/Bundles
```

example.coffee — presentation

example.coffee

```coffeescript
# Assignment:
number   = 42
opposite = true

# Conditions:
number = -42 if opposite

# Functions:
square = (x) -> x * x

# Arrays:
list = [1, 2, 3, 4, 5]

# Objects:
math =
  root:   Math.sqrt
  square: square
  cube:   (x) -> x * square x

# Splats:
race = (winner, runners...) ->
  print winner, runners

# Existence:
alert "I knew it!" if elvis?

# Array comprehensions:
cubes = (math.cube num for num in list)
```

Line: 21 Column: 31 | CoffeeScript | Soft Tabs: 2 | race

example.js — presentation

example.js

```
1  var cubes, list, math, num, number, opposite, race, square;
2  var __slice = Array.prototype.slice;
3  number = 42;
4  opposite = true;
5  if (opposite) {
6    number = -42;
7  }
8  square = function(x) {
9    return x * x;
10 };
11 list = [1, 2, 3, 4, 5];
12 math = {
13   root: Math.sqrt,
14   square: square,
15   cube: function(x) {
16     return x * square(x);
17   }
18 };
19 race = function() {
20   var runners, winner;
21   winner = arguments[0], runners = 2 <= arguments.length ?
   __slice.call(arguments, 1) : [];
22   return print(winner, runners);
23 };
24 if (typeof elvis !== "undefined" && elvis !== null) {
25   alert("I knew it!");
26 }
27 cubes = (function() {
28   var _i, _len, _results;
29   _results = [];
```

Line: 17 Column: 4 mongoDB (JavaScript) Soft Tabs: 2 square

Integration With Rails

```
gem 'rails', '3.1.0rc1'
```

```
app/
  assets/
    javascripts/
      store.js.coffee
```

Syntax

```
jQuery(function() {
  $("#title").show();
});
```

```
jQuery function()
$("#title").show()
```

```
jQuery ->
$("#title").show()
```

```
Store.prototype.add = function(item) {
  this.items.push(item);
}
```

```
Store.prototype.add = function(item)
  this.items.push(item)
```

```
Store.prototype.add = function(item)
  @items.push(item)
```

```
add: function(item)
  @items.push(item)
```

```
add: ->          (item)
  @items.push(item)
```

```
add: (item) ->
  @items.push(item)
```

- *Redundant Punctuation Omitted*
- `function` *becomes* `->`
- `this.` *becomes* `@`
- `{}` *becomes* ⇥

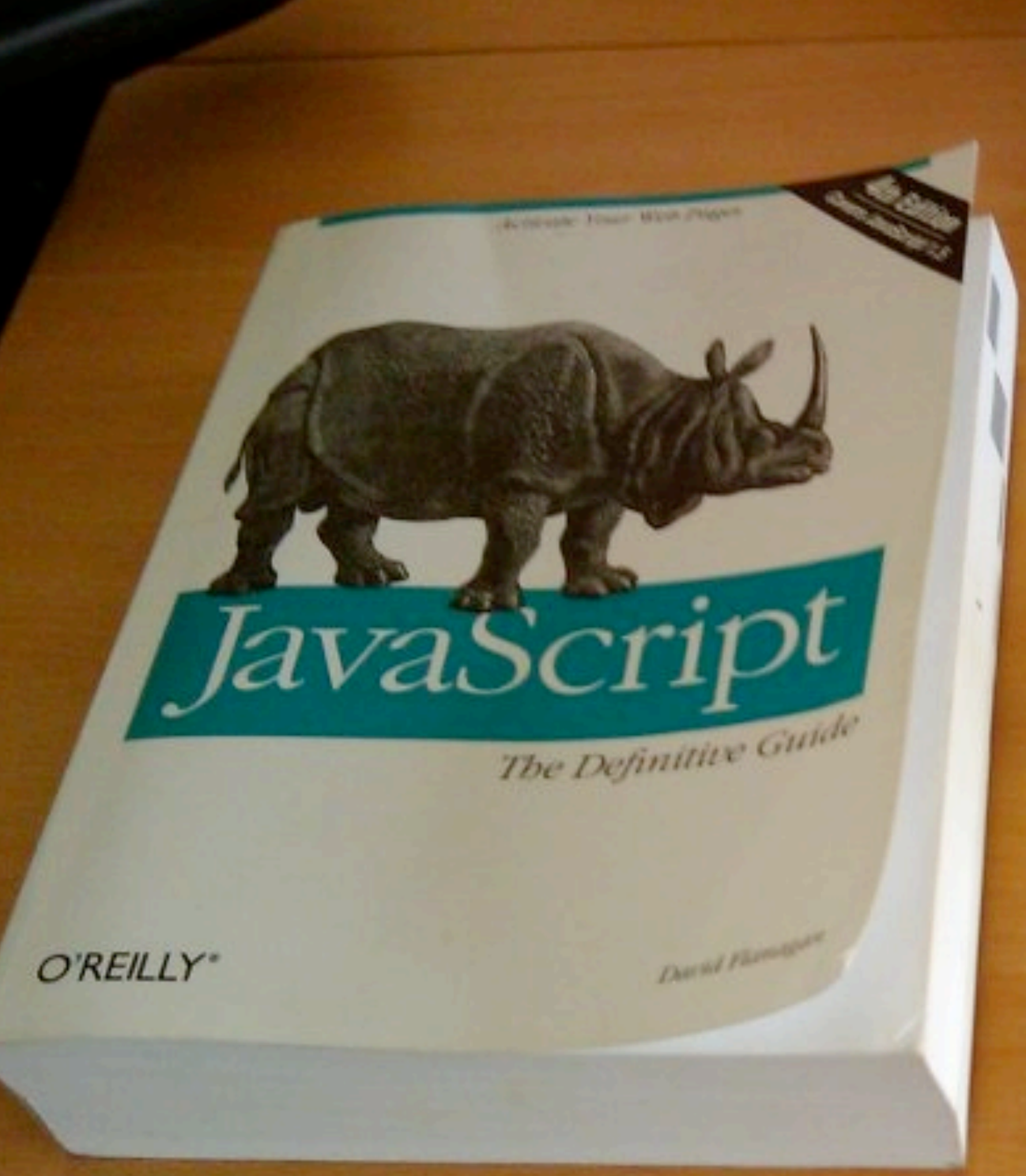
Features

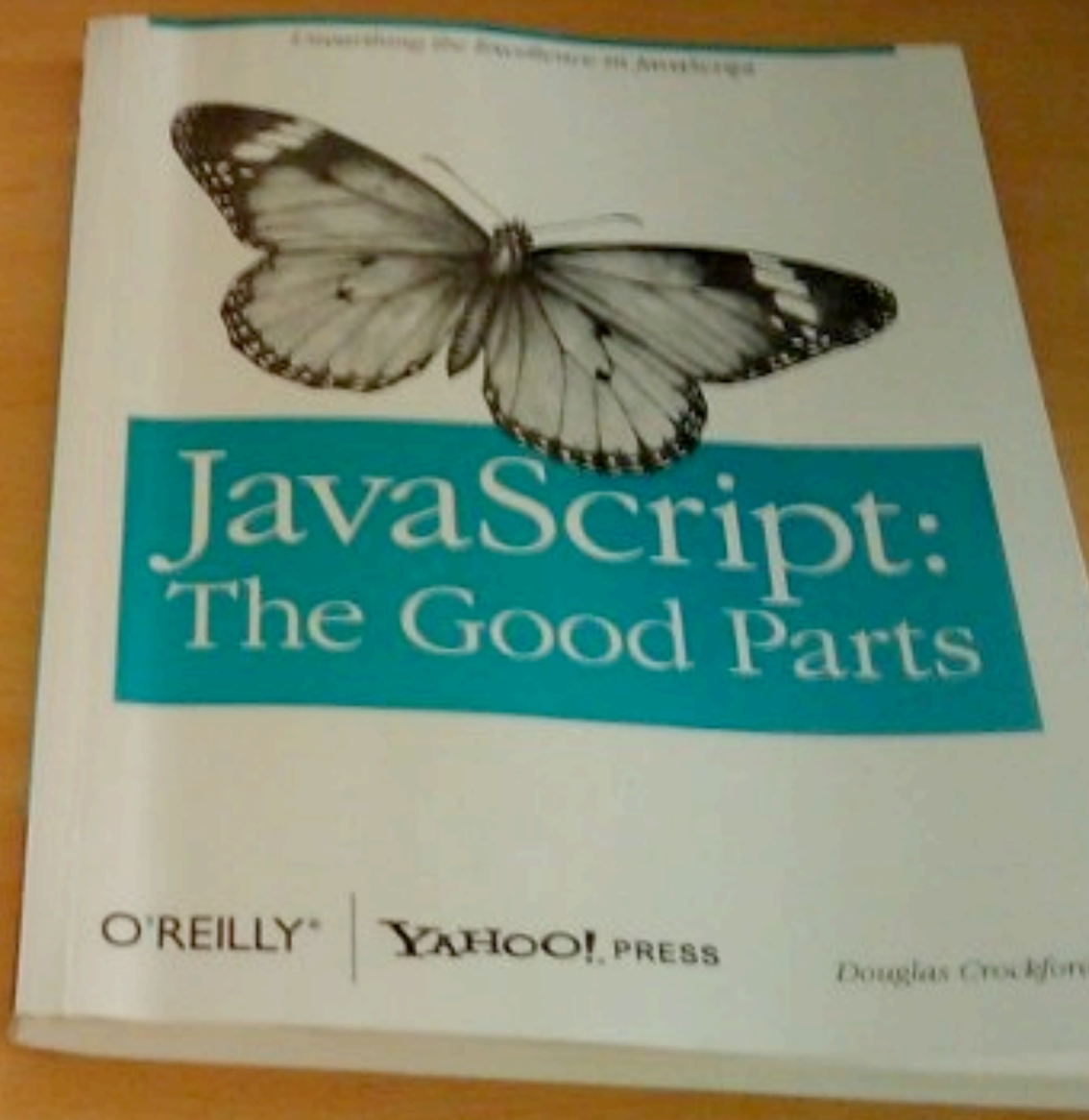
Unearthing the excellence in JavaScript

JavaScript: The Good Parts

O'REILLY® | YAHOO! PRESS

Douglas Crockford


JavaScript
The Definitive Guide
O'REILLY


JavaScript:
The Good Parts
O'REILLY
YAHOO! PRESS
Douglas Crockford

Class Pattern

```coffee
class Animal
  constructor: (name) ->
    @name = name
```

```
var Animal;
Animal = (function() {
  function Animal(name) {
    this.name = name;
  }
  return Animal;
})();
```

No Global Variables

```coffee
window.Animal = class Animal
  constructor: (name) ->
    @name = name
```

Defaault Arguments

```coffee
window.Animal = class Animal
  constructor: (name = "Unknown") ->
    @name = name
```

String Interpolation

```
class Bear extends Animal
  constructor: (name) ->
    @name = "#{name}, the Bear"
```

Conditional Suffixes

```
@price = amount if amount > 0.00
```

Operator Aliases

```
@lovesTacos = false unless @name is "Mattt"
```

CoffeeScript	*JavaScript*
is	===
isnt	!==
not	!
and	&&
or	||
true, yes, on	true
@, this	this
of	in
in	N/A

Deconstructing Assignment

```
[dollars, cents] = input.match /(\d+)/g
```

Existential Operator

?

```
[dollars, cents] = (input.match /(\d+)/g) ? [0, 0]
```

Splats

```coffee
gold, silver, bronze, rest = ""

awardMedals = (first, second, third, others...) ->
  gold   = first
  silver = second
  bronze = third
  rest   = others

pieEatingCompetitors = [
  "Damon Clinkscales",
  "Richard Schneeman",
  "Dave Rupert",
  "Rob Mack",
  "Adam Michela",
  "Steve Stedman",
  "Chris Continanza",
  "Keith Gaddis"
]

awardMedals contenders...
```

List Comprehensions

```
eat food for food in ['toast', 'cheese', 'wine']
```

```
eat food for food in ['toast', 'cheese', 'wine'] ↩
  when food is "toast"
```

```
languages = {
  "Javascript": 3,
  "CoffeeScript": 9,
  "Ruby": 9,
  "Python": 6,
  "Objective-C": 7,
  "Potion": 10
}

favorites = language for language, awesomeness of ↩
  languages when awesomeness >= 7
```

Introducing CoffeeScript

What We've Covered In This Talk

- Installation
- Integration With Rails
- Syntax
- Features

CoffeeScript is a little language that compiles into JavaScript. Underneath all of those embarrassing braces and semicolons, JavaScript has always had a gorgeous object model at its heart. CoffeeScript is an attempt to expose the good parts of JavaScript in a simple way.

The golden rule of CoffeeScript is: "*It's just JavaScript*". The code compiles one-to-one into the equivalent JS, and there is no interpretation at runtime. You can use any existing JavaScript library seamlessly (and vice-versa). The compiled output is readable and pretty-printed, passes through JavaScript Lint without warnings, will work in every JavaScript implementation, and tends to run as fast or faster than the equivalent handwritten JavaScript.

Latest Version: 1.1.1

Overview

CoffeeScript on the left, compiled JavaScript output on the right.

```
# Assignment:
number   = 42
opposite = true

# Conditions:
number = -42 if opposite

# Functions:
square = (x) -> x * x

# Arrays:
list = [1, 2, 3, 4, 5]

# Objects:
math =
  root:   Math.sqrt
  square: square
  cube:   (x) -> x * square x

# Splats:
race = (winner, runners...) ->
  print winner, runners

# Existence:
alert "I knew it!" if elvis?

# Array comprehensions:
cubes = (math.cube num for num in list)
```

```
var cubes, list, math, num, number, opposite, race, square;
var __slice = Array.prototype.slice;
number = 42;
opposite = true;
if (opposite) {
  number = -42;
}
square = function(x) {
  return x * x;
};
list = [1, 2, 3, 4, 5];
math = {
  root: Math.sqrt,
  square: square,
  cube: function(x) {
    return x * square(x);
  }
};
race = function() {
  var runners, winner;
  winner = arguments[0], runners = 2 <= arguments.length ?
__slice.call(arguments, 1) : [];
  return print(winner, runners);
};
if (typeof elvis !== "undefined" && elvis !== null) {
  alert("I knew it!");
}
cubes = (function() {
  var _i, _len, _results;
```

Meet

COFFEE SCRIPT

with

Geoffrey Grosenbach

Technical Review by Jeremy Ashkenas & Michael Ficarra

#267 CoffeeScript Basics

May 23, 2011 | 11 minutes | Rails 3.1

CoffeeScript allows you to write JavaScript in a concise, elegant fashion. Here I convert JavaScript code to CoffeeScript in a Rails 3.1 app.

Click to Play Video ▶

Download: source code mp4 m4v webm ogv

Resources

- CoffeeScript
- CoffeeScript TextMate Bundle

orders.js.coffee

```
CreditCard =
  cleanNumber: (number) -> number.replace /[- ]/g, ""

  validNumber: (number) ->
    total = 0
    number = @cleanNumber(number)
    for i in [(number.length-1)..0]
      n = +number[i]
      if (i+number.length) % 2 == 0
        n = if n*2 > 9 then n*2 - 9 else n*2
      total += n
    total % 10 == 0

jQuery ->
  $("#order_credit_card_number").blur ->
    if CreditCard.validNumber(@value)
      $("#credit_card_number_error").text("")
    else
      $("#credit_card_number_error").text("Invalid credit card number.")
```

Creating your first programming language is easier than you think.

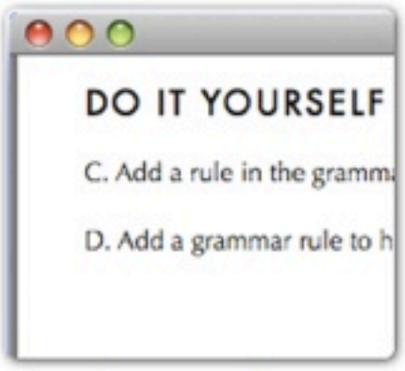
"The book I want to read." — Matz, creator of the Ruby language

Want to create a programming language, but don't feel like going through one of those expensive and boring 700+ pages book ? Well, you're not alone ...

The best system to create your first programming language.

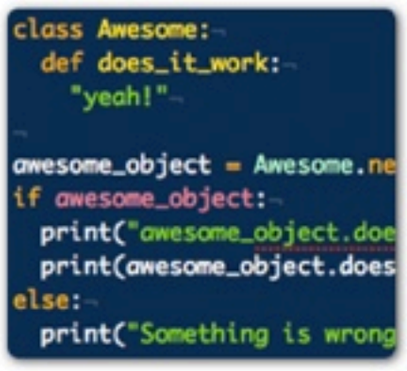
The eBook

A 53 pages PDF detailing core concepts and applying them to a custom language.



Exercises & solutions

Proposed extensions to the language with solutions at the end of the book.



Two languages

Full source code of two language in Ruby & Java. Easy to extend and play with.

A screencast

Explaining step by step how to extend the JVM language.

AUSTIN

3rd Thursdays, 7-9 at Norris Conference Center

http://austinrb.org

Introducing *CoffeeScript*

Mattt Thompson (***@mattt***)
Austin on Rails, May 2011