

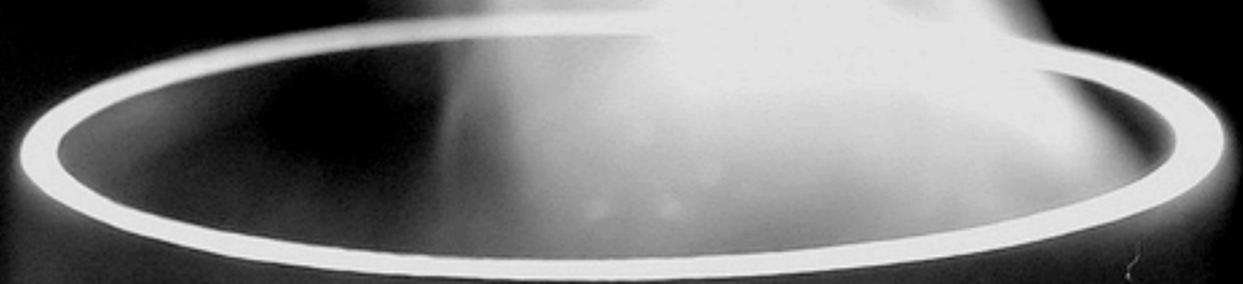
# CoffeeScript

*JavaScript's Less Ostentatious Kid Brother*

*Ryan McGeary*

<http://ryan.mcgeary.org>

*@rmm5t*





*JavaScript's less  
ostentatious kid brother*

*one-to-one  
with JavaScript*

*better  
functional syntax*

*compiles to  
the good parts*

<http://jashkenas.github.com/coffee-script/>

# Jeremy Ashkenas



# *JavaScript*

```
if (typeof elvis !== "undefined" && elvis !== null) {  
    alert("I knew it!");  
}
```



```
alert "I knew it!" if elvis?
```

# JavaScript

```
var cube, square;  
square = function(x) {  
    return x * x;  
};  
cube = function(x) {  
    return square(x) * x;  
};
```



square = (x) -> x \* x

cube = (x) -> square(x) \* x

# JavaScript

```
var _i, _len, _ref, _result, food, lunch;
lunch = (function() {
  _result = []; _ref = ['toast', 'cheese', 'wine'];
  for (_i = 0, _len = _ref.length; _i < _len; _i++) {
    food = _ref[_i];
    _result.push(eat(food));
  }
  return _result;
})();
```



```
lunch = (eat food for food in ['toast', 'cheese', 'wine'])
```

# JavaScript

```
var _i, _j, _len, _len2, _ref, _ref2, roid, roid2;
_ref = asteroids;
for (_i = 0, _len = _ref.length; _i < _len; _i++) {
  roid = _ref[_i];
  _ref2 = asteroids;
  for (_j = 0, _len2 = _ref2.length; _j < _len2; _j++) {
    roid2 = _ref2[_j];
    if (roid !== roid2) {
      if (roid.overlaps(roid2)) {
        roid.explode();
      }
    }
  }
}
```



```
for roid in asteroids
  for roid2 in asteroids when roid isnt roid2
    roid.explode() if roid.overlaps roid2
```

# *Installation*

```
$ brew install npm
```

```
# Add /usr/local/share/npm/bin to PATH
```

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

# Usage

```
$ coffee -c path/to/script.coffee
```

```
$ coffee --watch experimental.coffee
```

```
$ coffee --print *.coffee > all.js
```

# *Significant Whitespace*

```
if happy and knowsIt
  cLapsHands()
  chaChaCha()
else
  showIt()
```



# *Significant Whitespace*

```
if (happy && knowsIt) {  
    cLapsHands();  
    chaChaCha();  
} else {  
    showIt();  
}
```

*JavaScript*

# Functions

`square` =  $(x) \rightarrow x * x$

`area` =  $(x, y) \rightarrow x * y$

`noop` =  $\rightarrow$



# Functions

```
var area, noop, square;  
square = function(x) {  
    return x * x;  
};  
area = function(x, y) {  
    return x * y;  
};  
noop = function() {};
```

*JavaScript*

# Objects

```
kids =  
  brother:  
    name: "Max"  
    age: 11  
  sister:  
    name: "Ida"  
    age: 9
```



```
var kids;  
kids = {  
  brother: {  
    name: "Max",  
    age: 11  
  },  
  sister: {  
    name: "Ida",  
    age: 9  
  }  
};
```

# Lexical Scoping / Variable Safety

```
outer = 1  
change = ->  
  inner = -1  
  outer = 10  
inner = change()
```



```
(function() {  
  var change, inner, outer;  
  outer = 1;  
  change = function() {  
    var inner;  
    inner = -1;  
    return (outer = 10);  
  };  
  inner = change();  
}).call(this);
```

# Aliases

== >>> ==

is >>> ==

!= >>> !=

isnt >>> !=

# Aliases

on >>>> true

yes >>>> true

off >>>> false

no >>>> false

# Aliases

@property >>> this.property

# Aliases

and  $\gggg$   $\&\&$

or  $\gggg$   $\|\|$

not  $\gggg$   $!$

unless  $\gggg$   $\text{if } !$

# Aliases

winner = yes if pick in [47, 92, 13]

render = yes if key of { a: 1, b: 2 }

# OOP

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

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

class Snake extends Animal
  move: ->
    alert "Slithering..."
    super 5

sam = new Snake "Sammy the Python"
sam.move()
```



# OOP

```
var Animal, Snake, sam;
var __extends = function(child, parent) {
  var ctor = function(){};
  ctor.prototype = parent.prototype;
  child.prototype = new ctor();
  child.prototype.constructor = child;
  if (typeof parent.extended === "function") parent.extended(child);
  child.__super__ = parent.prototype;
};
Animal = function(_arg) {
  this.name = _arg;
  return this;
};
Animal.prototype.move = function(meters) {
  return alert(this.name + " moved " + meters + "m.");
};
Snake = function() {
  return Animal.apply(this, arguments);
};
__extends(Snake, Animal);
Snake.prototype.move = function() {
  alert("Slithering...");
  return Snake.__super__.move.call(this, 5);
};
sam = new Snake("Sammy the Python");
sam.move();
```

*JavaScript*

# Pattern Matching

```
theBait = 1000  
theSwitch = 0
```

```
[theBait, theSwitch] = [theSwitch, theBait]
```

---

```
weatherReport = (location) ->  
  [location, 72, "Mostly Sunny"]
```

```
[zip, temp, forecast] = weatherReport "20175"
```



# Pattern Matching

```
var _ref, forecast, temp, theBait, theSwitch, weatherReport, zip;  
theBait = 1000;  
theSwitch = 0;  
_ref = [theSwitch, theBait];  
theBait = _ref[0];  
theSwitch = _ref[1];
```

---

```
weatherReport = function(location) {  
  return [location, 72, "Mostly Sunny"];  
};  
_ref = weatherReport("20175");  
zip = _ref[0];  
temp = _ref[1];  
forecast = _ref[2];
```

*JavaScript*

# Existential Operator

```
alert "I knew it!" if elvis?
```

```
speed ?= 140
```

```
root = exports ? this
```

```
lottery.drawWinner()?.address?.zipcode
```



# Existential Operator

```
var _ref, _ref2, root, speed;  
if (typeof elvis !== "undefined" && elvis !== null) {  
  alert("I knew it!");  
}  
  
speed = (typeof speed !== "undefined" && speed !== null) ? speed : 140;  
  
root = (typeof exports !== "undefined" && exports !== null) ? exports : this;  
  
(typeof (_ref2 = ((_ref = lottery.drawWinner()))) === "undefined" || _ref2 ===  
null) ? undefined : _ref2.address === null ? undefined : _ref2.address.zipcode;
```

*JavaScript*

# String and RegExp Interpolation

```
quote = "A picture is a fact."  
author = "Wittgenstein"  
phrase = "#{quote} -- #{author}"  
  
sentence = "#{ 22 / 7 } approximates  $\pi$ "  
  
sep = "[.\ \/\- ]"  
dates = /\d+#{sep}\d+#{sep}\d+/g
```



# String and RegExp Interpolation

```
var author, dates, phrase, quote, sentence, sep;  
quote = "A picture is a fact.";  
author = "Wittgenstein";  
phrase = ("" + (quote) + " -- " + (author));  
  
sentence = ("" + (22 / 7) + " is a decent approximation of  $\pi$ ");  
  
sep = "[.\\|\\- ]";  
dates = (new RegExp("\\d+" + (sep) + "\\d+" + (sep) + "\\d+", "g"));
```

*JavaScript*

# Splat Arguments

```
awardMedals = (first, second, others...) ->  
  alert("Gold: #{first}");  
  alert("Silver: #{second}");  
  alert("The Field: #{others}");
```



# Splat Arguments

```
var awardMedals;  
var __slice = Array.prototype.slice;  
awardMedals = function(first, second) {  
  var others;  
  others = __slice.call(arguments, 2);  
  alert("Gold: " + (first));  
  alert("Silver: " + (second));  
  return alert("The Field: " + (others));  
};
```

*JavaScript*

# Array and Object Comprehensions

```
foods = ['toast', 'cheese', 'wine']  
lunch = (eat(food) for food in foods)
```

```
yearsOld = max: 10, ida: 9, tim: 11  
ages = for child, age of yearsOld  
      "#{child} is #{age}"
```



# Array and Object Comprehensions

```
var _i, _len, _ref, _result, age, ages, child, food, foods, lunch, yearsOld;
var __hasProp = Object.prototype.hasOwnProperty;
foods = ['toast', 'cheese', 'wine'];
lunch = (function() {
  _result = []; _ref = foods;
  for (_i = 0, _len = _ref.length; _i < _len; _i++) {
    food = _ref[_i];
    _result.push(eat(food));
  }
  return _result;
})();

yearsOld = {
  max: 10,
  ida: 9,
  tim: 11
};
ages = (function() {
  _result = []; _ref = yearsOld;
  for (child in _ref) {
    if (!__hasProp.call(_ref, child)) continue;
    age = _ref[child];
    _result.push("" + (child) + " is " + age);
  }
  return _result;
})();
```

*JavaScript*

# Slicing and Splicing

```
numbers = [0..9]
```

```
threeToSix = numbers[3..6]
```

```
copy = numbers[0..numbers.length]
```

```
numbers[3..6] = [-3, -4, -5, -6]
```



# *Slicing and Splicing*

```
var copy, numbers, threeToSix;  
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];  
  
threeToSix = numbers.slice(3, 6 + 1);  
  
copy = numbers.slice(0, numbers.length);  
  
numbers.splice.apply(numbers, [3, 6 - 3 + 1].concat([-3,  
-4, -5, -6]));
```

*JavaScript*

# Function Binding

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

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



# Function Binding

```
var Account;  
var __bind = function(func, context) {  
  return function(){ return func.apply(context, arguments); };  
};  
Account = function(customer, cart) {  
  this.customer = customer;  
  this.cart = cart;  
  return $('#checkout').bind('click', __bind(function(event) {  
    return this.customer.purchase(this.cart);  
  }, this));  
};
```

*JavaScript*

# The Rest...

- ✓ *Everything is an expression; always a return value*
- ✓ *Pattern matching with object literals*
- ✓ *Switch/When/Else*
- ✓ *While/Until/Loop*
- ✓ *Try/Catch/Finally*
- ✓ *Chained comparison*
- ✓ *Multiline Strings, Heredocs, and Block Comments*
- ✓ *"text/coffeescript" script tags with extras/coffee-script.js*
- ✓ *Cake and Cakefiles*

<http://jashkenas.github.com/coffee-script/>

*Let's Try It...*

# Ideas for Getting Started

**QUnit**



**timeago**  
a jQuery plugin



Jasmine

**flexselect**  
a jQuery plugin

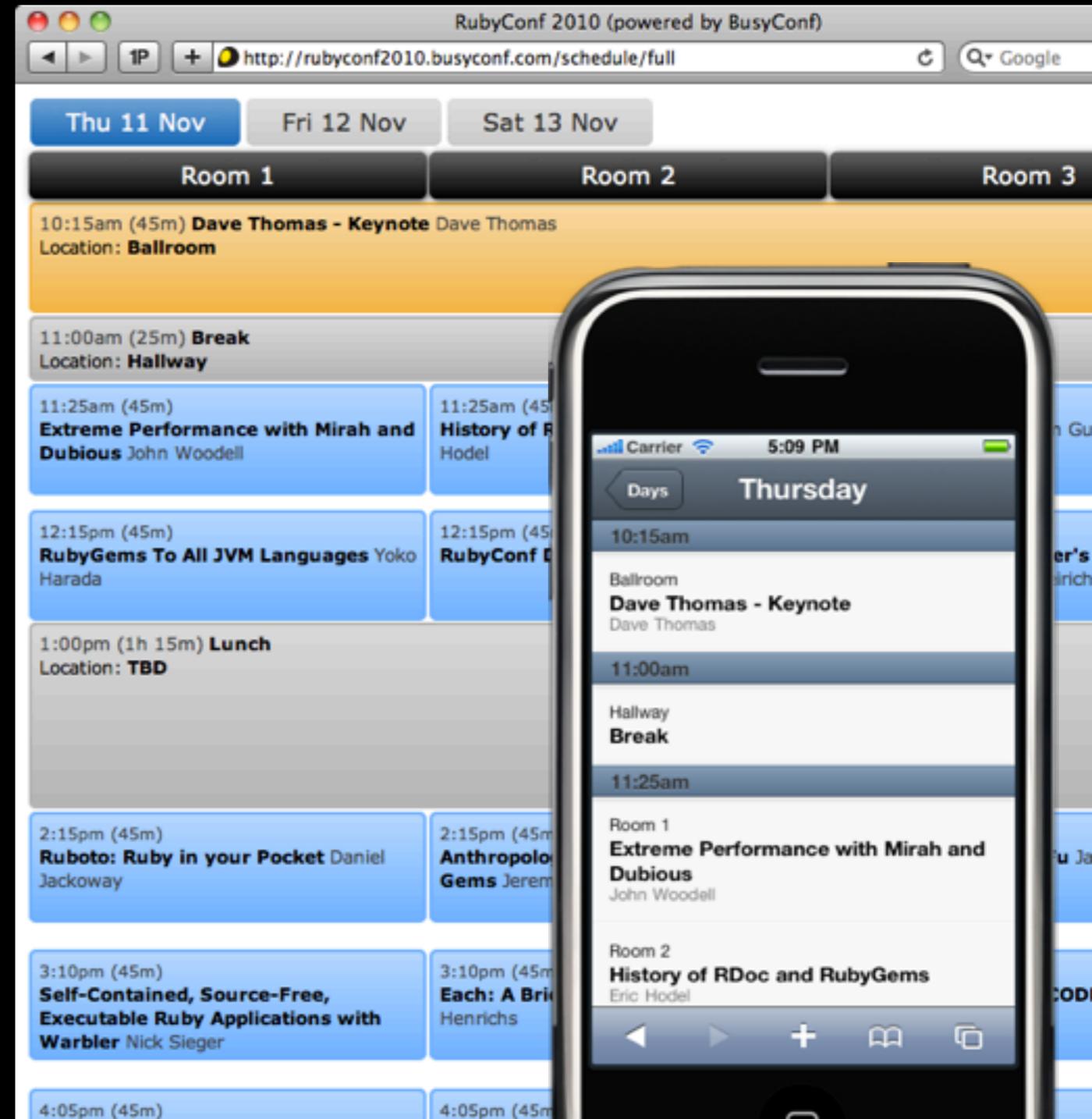
let me **Google** that for you

# Currently In Use



*Making great  
conferences even better*

*busyconf.com*



# Ryan McGeary

<http://ryan.mcgeary.org>

@rmm5t

[ryan@mcgeary.org](mailto:ryan@mcgeary.org)

The logo features the equation E=mc². The 'E' is large and white, followed by an equals sign, 'm', and 'c'. The 'g' is a stylized blue character. A blue dot is positioned above the 'c', with a grey arc connecting it to the top of the 'g'.

McGeary Consulting Group



# Attributions

<http://jashkenas.github.com/coffee-script/>

<http://www.flickr.com/photos/74234765@N00/488955057/>

<http://www.flickr.com/photos/adunne/3974874247/>

<http://www.flickr.com/photos/28111377@N07/2970550798/>

<http://www.flickr.com/photos/7678790@N06/3380560365/>

<http://www.flickr.com/photos/40775750@N00/531138641/>

<http://www.flickr.com/photos/86176561@N00/492795782/>

<http://www.flickr.com/photos/77555797@N00/133942287/>

<http://www.flickr.com/photos/34580986@N03/4985041197/>

<http://www.flickr.com/photos/83275741@N00/291831432/>

<http://www.flickr.com/photos/58115002@N00/3283033324/>

<http://www.flickr.com/photos/15133799@N02/3339157498/>

<http://www.flickr.com/photos/17731548@N00/981372736/>

<http://www.flickr.com/photos/7576193@N07/2476397335/>

<http://www.flickr.com/photos/48553010@N00/408767516/>

[http://www.free-computer-wallpapers.com/pictures/Television wallpaper/Alias 2](http://www.free-computer-wallpapers.com/pictures/Television_wallpaper/Alias_2)

<http://www.flickr.com/photos/44742295@N00/3998772594/>

<http://www.flickr.com/photos/79659919@N00/3413379549/>

<http://www.flickr.com/photos/82402200@N00/523497824/>