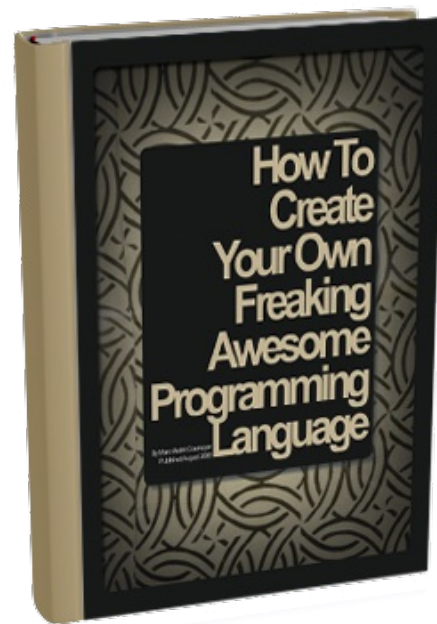


CoffeeScript

dotan@conduit.com, @jondot

jashkenas (Jeremy Ashkenas)

<https://github.com/jashkenas>



Make Javascript more like Ruby

Compile to JS

Not really a compiler

"Transformer"

First implemented in Ruby

Compiler: Racc

Later ported to Javascript/Node

Compiler: jison

Fun: see branches 0.1.0 vs current.

Syntax

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

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

Yes, thats **significant whitespace**

Pythonistas laugh

Rubyists cry

Deal with it.

Syntax (xformed)

```
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)));  
};
```


node.js Hello

```
http = require 'http'  
  
http.createServer (req, res) ->  
  res.writeHead 200  
  res.end 'hello world dooee' if true  
  
.listen 4000
```

```
(function() {  
  var http;  
  http = require('http');  
  http.createServer(function(req, res) {  
    res.writeHead(200);  
    if (true) {  
      return res.end('hello world dooee');  
    }  
  }).listen(4000);  
}).call(this);
```

So, really, this:

Instead of this:

```
{ { ( ; } ; ( {  
{ ) } ;  
{ { { } ; } ) ) ;  
! @ # $ ! @ % ^ $ # % ^ # % .  
;
```

Braces Free

```
foo 'argument'  
foo('argument')  
// Also, no curlies or `;`
```

var Free

```
foo = 'poo'  
var foo = 'poo';  
// don't worry:  
// lexical scoping is taken care of.
```

Functions

`f = (x) -> x*x`

`f = (x, y='default') -> x+y`

Keep using your Arrs and Hashes

a = [...]
a = { ... }

Cooler hash inits

```
h =  
  first_key:  
    address: 'palo alto'  
  second_key:  
    address: 'tiberias'
```

```
h= { first_key: { address: 'palo alto'},  
    second_key: { address: 'tiberias'}};
```

Conditionals

```
mood = greatlyImproved if singing  
date = if friday then sue else jill
```

Arguments*

```
f = (first, second, rest...) ->  
  // `rest` is like `params`/C# or `*`/Ruby.  
  // aka `splat`
```

Pythonish Comprehensions

```
countdown = (num for num in [10..1])
```

Everything Returns

```
num = 6  
lyrics = while num -= 1  
eldest = if 24 > 21 then "Liz" else "Ike"
```

Slice & Splice

```
numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
copy    = numbers[0...numbers.length]
middle  = copy[3..6]
```

Comparisons

```
//Coffee's == is JS's ==  
//Coffee's `is` is JS's ==  
//`isnt` is !=  
// `true, yes, on` are JS's `true`  
// `false, no, off` are JS's `false`  
// `@, this` is `this`  
letTheWildRumpusBegin() unless answer is no
```


Nulls

```
blah = true if some_var?  
init_var ?= 4
```

Useful WTF

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

// is this

```
var zip, _ref;  
zip = typeof lottery.drawWinner === "function" ?  
    (_ref = lottery.drawWinner().address) != null ?  
        _ref.zipcode : void 0 : void 0;
```

Classes!

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

  move: (meters) ->
    alert @name + " moved #{meters}m."
```

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

```
class Horse extends Animal
  move: ->
    alert "Galloping..."
    super 45
```

```
sam = new Snake "Sammy the Python"
tom = new Horse "Tommy the Palomino"
```

```
sam.move()
tom.move()
```

Assignments

```
[theBait, theSwitch] = [theSwitch, theBait]  
[city, temp, forecast] = weatherReport "Berkeley, CA"
```

Awesomeness (matching)

```
futurists =  
  sculptor: "Umberto Boccioni"  
  painter:  "Vladimir Burliuk"  
  poet:  
    name:    "F.T. Marinetti"  
    address: [  
      "Via Roma 42R"  
      "Bellagio, Italy 22021"  
    ]  
  
{poet: {name, address: [street, city]}} = futurists
```

Param binding

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

```
$('.shopping_cart').bind 'click', (event) =>  
  @customer.purchase @cart  
// remember: `@` is `this` ?
```

```
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)));  
};
```

Switch

```
switch day
  when "Mon" then go work
  when "Tue" then go relax
  when "Thu" then go iceFishing
  when "Fri", "Sat" //multi value - ruby.
    if day is bingoDay
      go bingo
      go dancing
    when "Sun" then go church
  else go work
```


Chained Compares

```
healthy = 200 > cholesterol > 60 // Python
```

String Interp.

```
//like ruby  
sentence = "#{ 22 / 7 } is approx:  $\pi$ "
```

Heredocs

```
foo = '''  
    great  
    multi lined  
    text  
    '''
```

Herecomments

```
###  
great  
multi lined  
comment  
###
```

HereRegex

```
foo = /// ^ (  
?: [-=>] > # function  
| [-+*/%<>&|^!?=] = # compound assign  
| >>>=? # zero-fill right shift  
| ([-+:])\1 # doubles  
| ([&|<>])\2=? # logic / shift  
| \?\.  
| \.{2,3} # range or splat  
) ///
```

Getting It

`npm install coffeescript`

`gem install coffeescript`

Load coffeescript in JS host

Using it

`coffee`

`coffee --compile --watch *.coffee`

In some envs, JS host works (Rails3.1)

Read Some Code



Thanks

dotan@conduit.com, @jondot