



# CoffeeScript

## The Awesome

# “A Propos de Moi”

**James Hughes**  
Technicial Architect

Email: j.hughes@kainos.com  
Tel: +44 (0) 28 9057 1228

Kainos House  
4-6 Upper Crescent  
Belfast, BT7 1NT  
[www.kainos.com](http://www.kainos.com)



@kouphax



<http://yobriefca.se>



<http://github.com/kouphax>



# A Few CoffeeScript Facts

- Created by Jeremy Ashkenas circa late 2009
- Spawnsed from reading “Create Your Own Programming Language”
- CoffeeScript compiles to, and executes as, JavaScript in the browser.
- CoffeeScript simply offers an initial compilation step prior to the JavaScript being interpreted – what's the harm in that?
- CoffeeScript does not require you to abandon all your JS libraries
  - Yes you can still use jQuery or anything else you want
- JavaScript produced by the CoffeeScript compiler is performant, valid and well formatted.

# A Few (More) CoffeeScript Facts

- Debugging CoffeeScript directly is currently not possible so this adds another layer of complexity
  - In reality mapping between CoffeeScript and the generated JavaScript isn't difficult
- Could be considered just syntactical sugar with some extra syntax benefits
  - No more random semi-colon insertion
  - No more accidental global scoping of declarations
  - Scoping syntax built right in
  - Automatic “hoisting” of variable declarations
- Not everyone is a massive fan

fizzbuzz.js #

```
1 function fizzbuzz(start, end){  
2     for(int i = start; i <= end; i++){  
3         var output = ""  
4  
5         if((i % 3) == 0){  
6             output += "fizz";  
7         }  
8  
9         if((i % 5) == 0){  
10            output += "buzz";  
11        }  
12  
13         console.log(output);  
14     }  
15 }
```

## Range Comprehension

```
[ ]
```

## Loop Comprehension

```
[ ]
```

\* **Never** do this in real life kthanxai

```
fizzbuzz.coffee #  
1 fizzbuzz = (start, end) -> console.log "#{if i % 3 is 0 then 'fizz' else ''}#{if i % 5 is 0 then 'buzz' else ''}" for i in [start..end]
```

Variable Declaration

Function Declaration

Function Call

English Notation

String Interpolation

compiled.js #

```
1 var fizzbuzz;
2 fizzbuzz = function(start, end) {
3   var i, _results;
4   _results = [];
5   for (i = start; start <= end ? i <= end : i >= end; start <= end ? i++ : i--) {
6     _results.push(console.log((i % 3 === 0 ? 'fizz' : '') + (i % 5 === 0 ? 'buzz' : '')));
7   }
8   return _results;
9 };
```

# **CoffeeScript: A Guided Tour**

**- or -**

**CoffeeScript: A Guided Tour of the Website, Pretty  
Much, Except in Slide Form but Not Because I am Lazy,  
No, More so Because It's a Fairly Comprehensive  
Document of the Features of CoffeeScript.**

# Functions

```
functions.coffee #
```

```
1 add = (a,b) -> a + b
```

```
functions.js #
```

```
1 function add(a,b) {
2     return a + b;
3 }
```

# Functions

```
functions1.coffee #
```

```
1 add = (a = 0, b = 0) -> a + b
```

```
functions1.js #
```

```
1 function add(a,b){  
2     if (a == null) {  
3         a = 0;  
4     }  
5     if (b == null) {  
6         b = 0;  
7     }  
8     return a + b;  
9 }
```

# Objects

objects.coffee #

```
1 ratings = {talk1: "Weak", talk2: "Lame"}  
2  
3 ratings =  
4   talk1:  
5     name: "refORM: Death to ORMs"  
6     value: "Weak"  
7   talk2:  
8     name: "CoffeeScript the Awesome"  
9     value: "Lame"
```

objects.js #

```
1 var ratings = {  
2   talk1: "Weak",  
3   talk2: "Lame"  
4 };  
5  
6 var ratings = {  
7   talk1: {  
8     name: "refORM: Death to ORMs",  
9     value: "Weak"  
10   },  
11   talk2: {  
12     name: "CoffeeScript the Awesome",  
13     value: "Lame"  
14   }  
15 };
```

# Conditionals

```
ifunless.coffee #
```

```
1 rating = increase() if crackingJokes
2
3 rating = decrease() unless crackingJokes
4
```

```
ifunless.js #
```

```
1 var rating;
2 if (crackingJokes) {
3   rating = increase();
4 }
5
6 var rating;
7 if (!crackingJokes) {
8   rating = decrease();
9 }
```

# Operators and Aliases

ifunless.coffee #

```
1 -----
2 CoffeeScript | JavaScript
3 -----
4 is          | ==
5 isnt        | !=
6 not         | !
7 and         | &&
8 or          | ||
9 true, yes, on | true
10 false, no, off | false
11 @, this     | this
12 of          | in
13 in          | N/A
14 -----
```

inis.coffee #

```
1 badTalk    = if average in [1..2]
2 decentTalk = if average is 3
3 goodTalk   = if average in [4..5]
```

# Splats

```
splats.coffee #
```

```
1 addAll = (start, nums...) -> start + nums.sum()
```

```
splats.js #
```

```
1 function addAll() {
2   var start = arguments[0],
3   var nums = 2 <= arguments.length ? Array.prototype.slice.call(arguments, 1) : [];
4   return start + nums.sum();
5 }
```

# Loops and Comprehensions

```
loops.coffee #
```

```
1 sayHello person for person in ['James', 'Will', 'Stanley']
```

```
loops.js #
```

```
1 var people = ['James', 'Will', 'Stanley'];
2 for (var i = 0; i < people.length; _i++) {
3   var person = people[i];
4   sayHello(person);
5 }
```

# Array Slicing and Ranges

```
ranging.coffee #
```

```
1 x = [1..10]
2
3 y = [lower..upper]
4
5 subx = x[1..x.length-1]
```

```
ranging.js #
```

```
1 var x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
2
3 var y = (function() {
4   var results = [];
5   for (var i = lower; lower <= upper ? i <= upper : i >= upper; lower <= upper ? _i++ : i--) {
6     results.push(i);
7   }
8   return _results;
9 }).apply(this);
10
11 var subx = x.slice(1, (x.length - 1 + 1) || 9e9);
```

# Classes

```
classes.coffee #  
1  class Animal  
2    constructor: (@name) ->  
3  
4    move: (meters) ->  
5      alert @name + " moved #{meters}m."  
6  
7  class Snake extends Animal  
8    move: ->  
9      alert "Slithering..."  
10     super 5  
11  
12 class Horse extends Animal  
13   move: ->  
14     alert "Galloping..."  
15     super 45  
16  
17 sam = new Snake "Sammy the Python"  
18 tom = new Horse "Tommy the Palomino"  
19  
20 sam.move()  
21 tom.move()
```

# Odds and Ends

```
oddsandends.coffee #  
1  theBait    = 1000  
2  theSwitch = 0  
3  
4  [theBait, theSwitch] = [theSwitch, theBait]  
5  
6  # -----  
7  
8  author = "Wittgenstein"  
9  quote  = "A picture is a fact. -- #{ author }"  
10  
11 sentence = "#{ 22 / 7 } is a decent approximation of π"  
12  
13 # -----  
14  
15 html = """  
16   <strong>  
17     cup of coffeescript  
18   </strong>  
19 """  
20  
21 # -----  
22
```

# Using CoffeeScript in .NET

- or -

**What? You think we'd let those script kiddies have all  
the fun? Hells no!**

# Compile On Demand



...and others

*Shameless Plug  
(but I am distracting  
you through the use  
of Comic Sans)*

# Pre-compile

MINDSCAPE } }

# Web Workbench

- Sass ■ Less ■ CoffeeScript

Works with  
Visual Studio 2010

Add New Item - WebApplication5

Installed Templates

Sort by: Default

Search Installed Templates

Type: Visual C#  
A script file containing CoffeeScript code

CoffeeScript1.js X Default.aspx

```
1 (function() {
2     var cubes, list, math, num, number, o
3     var __slice = Array.prototype.slice;
4     number = 42;
5     opposite = true;
6     if (opposite) {
7         number = -42;
8     }
9     square = function(x) {
10         return x * x;
11     };
12     list = [1, 2, 3, 4, 5];
13     math = {
14         root: Math.sqrt,
15         square: square,
16         cube: function(x) {
17             return x * square(x);
18         };
19     };
20     race = function() {
21         var runners, winner;
22         winner = arguments[0], runners = 2
23         return print(winner, runners);
24     };
25     if (typeof elvis !== "undefined" && elvis)
26         alert("I knew it!");
27     cubes = (function() {
28         var _i, _len, _results;
29         _results = [];
30         for (_i = 0, _len = list.length; _i
31             num = list[_i];
32             _results.push(math.cube(num));
33         }
34         return _results;
35     })();
36 }).call(this);
```

CoffeeScript1.coffee X

```
1 # Assignment:
2 number = 42
3 opposite = true
4
5 # Conditions:
6 number = -42 if opposite
7
8 # Functions:
9 square = (x) -> x * x
10
11 # Arrays:
12 list = [1, 2, 3, 4, 5]
13
14 # Objects:
15 math =
16     root: Math.sqrt
17     square: square
18     cube: (x) -> x * square x
19
20 # Splat:
21 race = (winner, runners...) ->
22     print winner, runners
23
24 # Existence:
25 alert "I knew it!" if elvis?
26
27 # Array comprehensions:
28 cubes = (math.cube num for num in list)
```

Solution 'WebApplication5' (1 project)

WebApplication5

- Properties
- References
- Account
- App\_Data
- Scripts
- Styles
- About.aspx
- CoffeeScript1.coffee
  - CoffeeScript1.js
- Default.aspx
- Global.asax
- Site.Master
- Web.config



# Questions?