CoffeeScript

@20th Ruby Tuesday

高見龍 

photo by gillyberlin

@20th Ruby Tuesday

You can get this slide on

www.eddie.com.tw/slides

Who am I ?

I'm a Flash guy.

photo by JD Hancock

hmm..

maybe only

60% Flasher

and

40% Rubyist

photo by Crashmaster007

高見龍

a.k.a Eddie or Aquarianboy

- Live and work in Taipei, Taiwan.
- Serving in my own little tiny company.
- Flash / AS3 / Ruby / Rails / Python programming for living.
- A little bit Objective-C for personal interests.
- Technical Education and Consultant.
- PTT Flash BM (since 2007/4).
- Adobe Certificated Flash Developer (Since 2006/7).
- Linux Professional Institute Certification (Since 2005/3).

photo by Eddie

or just google with the keyword "高見龍"

JavaScript becomes hugely popular.

9th on TIOBE, Nov 2011

photo by gr3m

Ruby on Rails

Sustainable productivity for web-application development

photo by Andrew

JavaScript
seems
easy..

photo by apple apple

But it's also easy to get messy!

photo by Marcus Q

CoffeeScript

We're not talking about this kind of coffee.

photo by Nick Humphries

Not this either.

photo by naotoj

WTF?

CoffeeScript is JavaScript

just written in different syntax

Who won't need this?

people who already know javascript(the good parts) very well.

What's the problems?

photo by Steve Ganz

Bad smell..

Will it be equal? that's the question.

```
0 == ""                 // true
1 == true               // true
1 == "1"                // true
1 == "1.0"              // true
"true" ==  true         // true
"true" ===  true        // false
```

What the..?

photo by Bryan Gosline

> I really don't know how to write good JavaScript!

> CoffeeScript exposes the good parts of JavaScript in a simple way.

Syntax borrowed from Python and Ruby.

I love Python & Ruby

> If you're already familiar with Ruby, you've probably already learned about 70% CoffeeScript.

Really friendly for Rubyist.

```
love_coffeescript == true if rubyist
```

and compiles into JavaScript code.

*.coffee -> *.js

Your Brain ->

JavaScript ->

Browser

Your Brain ->
Friendly CoffeeScript ->
JavaScript ->
Browser

> But that doesn't mean you can have no knowledge about JavaScript.

CoffeeScript is NOT used to replace JavaScript.

Let’s get our feet wet!

photo by jlhopes

What are we looking at today?

- Installation
- Usage
- Syntax

Install

photo by Daniel Dionne

Requirements

You need to install some software first..

Node.js

```
> git clone git://github.com/joyent/node.git
> cd node
> ./configure
> make
> sudo make install
```

Requirements

You need to install some software first..

NPM, the “node package manager”

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

Install CoffeeScript

CoffeeScript

```
> npm install coffee-script
> coffee -v
CoffeeScript version 1.1.3
```

http://blog.eddie.com.tw/2011/08/03/install-coffeescript/

Install CoffeeScript

CoffeeScript on MacOS

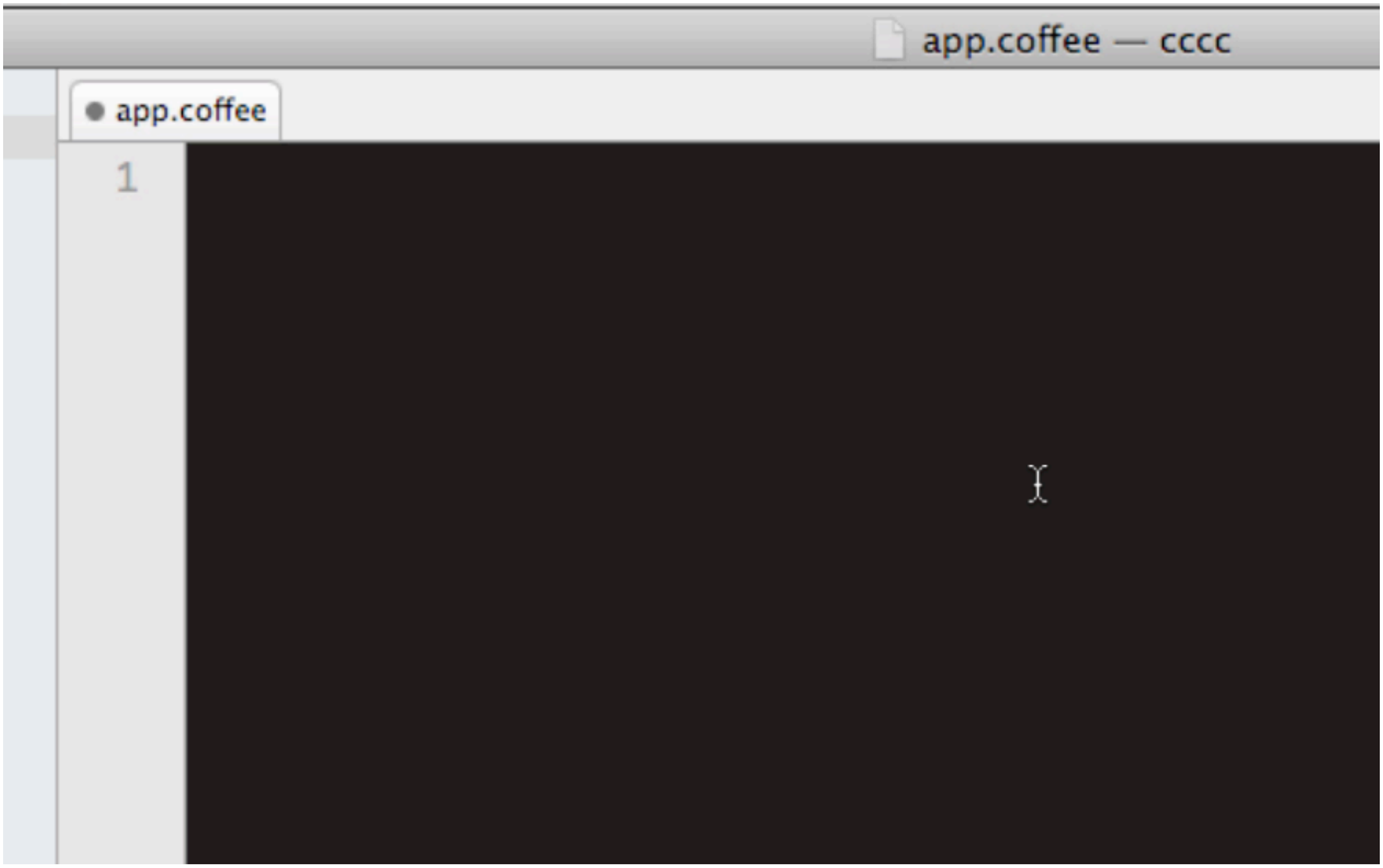
```
> brew install coffee-script
```

> ..and don't forget install the bundle for CoffeeScript if you're using TextMate.

http://blog.eddie.com.tw/2011/08/03/install-coffeescript/

Install CoffeeScript

TextMate bundle forCoffeeScript



https://github.com/jashkenas/coffee-script-tmbundle

Install CoffeeScript

Windows?

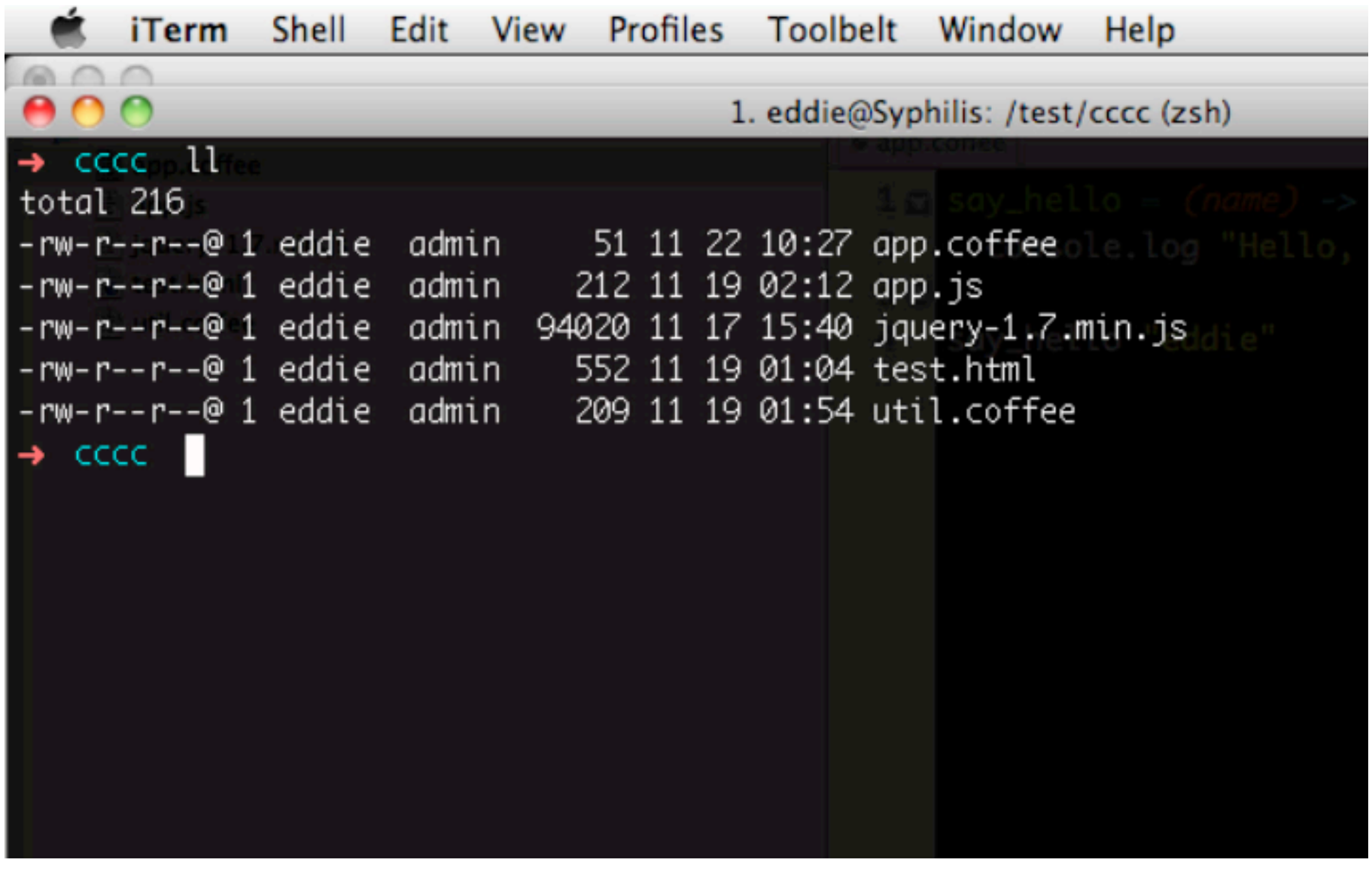
get a mac

How to use

Usage

Compile *.coffee into *.js

```
> coffee --watch --compile app.coffee
```



http://blog.eddie.com.tw/2011/08/03/how-to-use-coffeescript-compiler/

Usage

you got a REPL, just like irb.

Don’t like to compile?

You can run CoffeeScript in HTML directly

Use CoffeeScript on the fly

```
<script type="text/javascript" src="http://
jashkenas.github.com/coffee-script/extras/coffee-
script.js"></script>

<script type="text/coffeescript">
  alert 'test'
</script>

<script type="text/coffeescript" src="myapp.coffee"></script>
```

don’t do this in your production.

Syntax

photo by zigazou76

}

}

}

}

No { }

indentations rule!

whitespace matters!

( ) is not always necessary.

Just like Ruby

but..

hello +5 means hello(+5)

+ : convert string to number

No trailing semicolon.

Return is not necessary.

everything is an expression, just like Ruby.

No { }, (), and ;

```
if(age > 20){
    vote();
}
```

// javascript

```
if age > 20
    vote()
```

coffeescript

Variable & Function

Variable

You don't have to declare it before using it.

```
lang = ["php", "python", "perl", "ruby"]
name = "Eddie"
```

coffeescript

```
var lang, name;
lang = ["php", "python", "perl", "ruby"];
name = "Eddie";
```

// javascript

Variable

Destructuring Assignment

```coffeescript
x = 100
y = 10

[x, y] = [y, x]
```

coffeescript

```javascript
var x, y, _ref;
x = 100;
y = 10;
_ref = [y, x], x = _ref[0], y = _ref[1];
```

// javascript

Variable

Destructuring Assignment

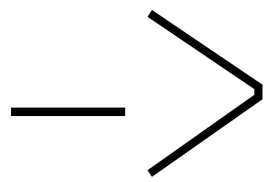
```
weatherReport = (location) ->
  [location, 72, "Mostly Sunny"]

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

coffeescript

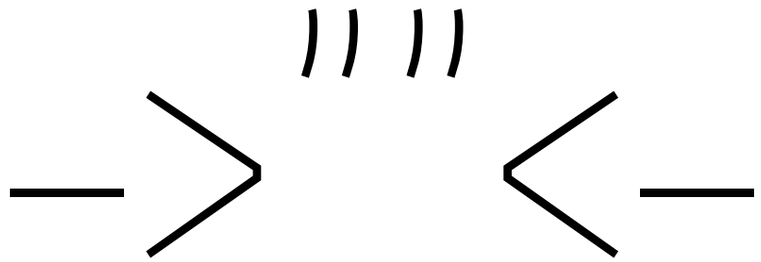
```
tag = "<awesome>"
[open, contents..., close] = tag.split("")
```

coffeescript



dash rocket

Not



enjoy coding :)

-> "hello, Ruby Tuesday"

```
(function() {
  return "hello, Ruby Tuesday";
});
```

Function

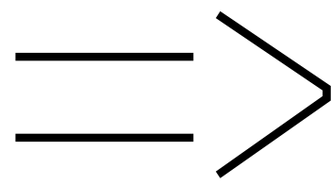
```
say_hello = (guest1, guest2 = "Nayumi") ->
    "Hello #{guest1} and #{guest2}"

say_hello "Eddie"
```

coffeescript

```
var say_hello;
say_hello = function(guest1, guest2) {
    if (guest2 == null) {
        guest2 = "Nayumi";
    }
    return "Hello " + guest1 + " and " + guest2;
};
say_hello("Eddie");
```

// javascript



fat arrow

http://blog.eddie.com.tw/2011/11/18/dash-rocket-vs-fat-arrow-in-coffeescript/

Splats

Just like Ruby

Splats

```
sum = (nums...) ->
  result = 0
  result += n for n in nums
  result

console.log sum(1, 2, 3, 4, 5)
```

coffeescript

Array

Array

```
heroes = [
    'Spider Man',
    'Captain America',
    'X-men',
    'Iron Man'
]

students = [1..10]
teachers = [1...10]
```

coffeescript

```
var heroes, students, teachers;
heroes = ['Spider Man', 'Captain America', 'X-men', 'Iron Man'];

students = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
teachers = [1, 2, 3, 4, 5, 6, 7, 8, 9];
```

// javascript

Array

```
heroes[0..2]

heroes[1..2] = ["Batman", "ThunderCat"]
```

coffeescript

```
heroes.slice(0, 3);

var _ref;
[].splice.apply(heroes, [1, 2].concat(_ref = ["Batman", "ThunderCat"])), _ref;
```

// javascript

Object

Object

```coffeescript
eddie = { name: "Eddie Kao", age: 18, speciality: "eat" }
```

coffeescript

```javascript
var eddie;
eddie = {
  name: "Eddie Kao",
  age: 18,
  speciality: "eat"
};
```

// javascript

Object

```coffeescript
eddie =
  name: "Eddie Kao"
  age: 18
  lovers:
    nayumi:
      name: "Nayumi Hung"
      age: 18
    mary:
      name: "Mary Bloody"
      age: 20
```

coffeescript

```javascript
var eddie;
eddie = {
  name: "Eddie Kao",
  age: 18,
  lovers: {
    nayumi: {
      name: "Nayumi Hung",
      age: 18
    },
    mary: {
      name: "Mary Bloody",
      age: 20
    }
  }
};
```

// javascript

Loop

But we Rubyist barely use for-loop

List Comprehension

```coffeescript
alert i for i in [1..10]



alert i for i in [1..10] when i
% 2 == 0



 alert i for i in [1..10] by 2



 alert i * 2 for i in [1..10]
```

coffeescript

```javascript
var i, _step;
for (i = 1; i <= 10; i++) {
  alert(i);
}
for (i = 1; i <= 10; i++) {
  if (i % 2 === 0) {
    alert(i);
  }
}
for (i = 1, _step = 2; i <= 10; i +=
_step) {
  alert(i);
}
for (i = 1; i <= 10; i++) {
  alert(i * 2);
}
```

// javascript

Condition

Switch

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

coffeescript

Modifier

You can put "if", "unless", "while", "until" behind

```
if (age > 20) {
   vote();
}
```

```
// javascript
```

```
vote() if age > 20
```

```
# coffeescript
```

Syntactic Sugar

Syntactic Sugar

write more readable code by using syntactic sugar.

# coffeescript	// javascript
is	===
isnt	!==
true, on, yes	true
false, off, no	false
not	!
and	&&
or	||
unless	if not
until	while not
of	in

Syntactic Sugar

```
alert "I can't see anything" if light is off

alert "It's impossible!" if eddie isnt handsome

if girl is not single
    alert "Don't Touch! Be Careful!"
```

coffeescript

Syntactic Sugar

```
if (light === false) {
    alert("I can't see anything");
}
if (eddie !== handsome) {
    alert("It's impossible!");
}
if (girl === !single) {
    alert("Don't Touch! Be Careful!");
}
```

// javascript

Syntactic Sugar

Which answer do you like to hear when you say ...

Eddie: Will you marry me?
Nayumi: **yes**!

or

Eddie: Will you marry me?
Nayumi: **true**!

Syntactic Sugar

```
alert "I'll marry you!" if answer is yes
```

coffeescript

```
if (answer === true) {
    alert("I'll marry you!");
}
```

// javascript

Syntactic Sugar

Chained Comparison

```coffeescript
console.log "I'm awesome!" if 20 < my_girl_friends < 100
```

coffeescript

```javascript
if ((20 < my_girl_friends && my_girl_friends < 100)) {
  console.log("I'm awesome!");
}
```

// javascript

Syntactic Sugar

Existential Operator

```
age ?= 18
```

coffeescript

```
if (typeof age !== "undefined" && age !== null) {
    age;
} else {
    age = 18;
};
```

// javascript

Raw JavaScript

If you still prefer the original way..

Raw JavaScript

```
say_hello = `function(name){
    return "Hello, " + name
}`
```

coffeescript

OOP

CoffeeScript's classes are syntactic sugar only.

Prototype?

IMHO, prototype-based OO is not elegant

OOP - new

```
class Animal
    constructor: (name, age) ->
        this.name = name
        this.age = age

animal = new Animal("eddie", 18)
alert animal
```

coffeescript

```
var Animal, animal;
Animal = (function() {
  function Animal(name, age) {
    this.name = name;
    this.age = age;
  }
  return Animal;
})();
animal = new Animal("eddie", 18);
alert(animal);
```

// javascript

OOP - @ = this.

app.coffee — cccc

app.coffee

```coffee
class Animal
  constructor: (name, age) ->
    this.name = name
    this.age = age

animal = new Animal("eddie", 18)
alert animal
```

OOP - method

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

    say_hello: (something) ->
        console.log "Hello, #{something}"

animal = new Animal("eddie", 18)
animal.say_hello("CoffeeScript")
```

coffeescript

OOP - inheritance

```
class Animal
    constructor: (@name, @age) ->
    say_hello: (something) ->
        alert "Hello, #{something}"

class Human extends Animal
    walk: ->
        alert "I can walk with my foots!"

eddie = new Human("eddie", 18)
eddie.say_hello "CoffeeScript"
eddie.walk()
```

coffeescript

OOP - inheritance

pp.coffee

```coffeescript
class Animal
    constructor: (@name, @age) ->
    say_hello: (something) ->
        alert "Hello, #{something}"

class Human extends Animal
    walk: ->
        alert "I can walk with my foots!"

eddie = new Human("eddie", 18)
eddie.say_hello "CoffeeScript"
eddie.walk()
```

TL;DR

Extend more function?

Extend more function

```coffeescript
String::repeat = (n) ->
  Array(n + 1).join @

String::downcase = ->
  @toLowerCase()

String::upcase = ->
  @toUpperCase()

String::find = (str) ->
  @indexOf str

String::has = (str) ->
  (@indexOf str) > 0
```

coffeescript

http://blog.eddie.com.tw/2011/11/19/extend-functionality-for-your-class/

Cooperate with other JS library or framework?

working with jQuery

```
$(document).ready(function() {
  wakeup();
  walk_to_toilet();
  pee();
  go_back_to_sleep();
})
```

// javascript

```
$ ->
  wakeup()
  walk_to_toilet()
  pee()
  go_back_to_sleep()
```

coffeescript

working with jQuery

```
$('#girl').animate({
   width: '100px',
   height: '50px',
   opacity: '0.8'
}, 2000, 'easeOutQuad');
```

// javascript

```
$('#girl').animate
   width: '100px'
   height: '50px'
   opacity: '0.8'
   2000
   'easeOutQuad'
```

coffeescript

working with jQuery

```
(function($){
  $.fn.extend({
    isEmail: function(email){
      return /some email validator/.test(email);
    }
  });
})(jQuery);
```

// javascript

```
$ = jQuery

$.fn.extend
  isEmail: (email) ->
    /some email validator/.test email
```

coffeescript

makes jQuery more easy,

and more fun

http://blog.eddie.com.tw/2011/11/14/when-jquery-meets-coffeescript/

References

Websites:

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

http://blog.eddie.com.tw/category/coffeescript/

http://upgrade2rails31.com/coffee-script

Koans:

https://github.com/sleepyfox/coffeescript-koans

References

Book:

http://pragprog.com/book/tbcoffee/coffeescript

http://arcturo.github.com/library/coffeescript/index.html

http://autotelicum.github.com/Smooth-CoffeeScript/

Conclusion

being a 40% Rubyist

I don't like..

var

return

for loop

()

{}

I Love..

I Love..

Coding Style

I love Python & Ruby, of course :)

I Love..

Indentation!

I Love..

Anonymous function

No global function and variable by default

I Love..

String Interpolation

sorry, but string building really sucks :)

I Love..

List Comprehension

I Love..

Syntactic Sugar

I Love..

English-like grammar

alert "of course it is!" if ruby_tuesday is awesome

I Love..

Comparison & Equality

```
"true" ==  true   // true
"true" === true   // false
```

I Love..

Works with other JS frameworks well.

Because it's just JavaScript

I Love..

Compilation

JSLint Approved

What else?

Make, Rake, Cake.

More and more projects are written in CoffeeScript.

Pow.cx

CoffeeScript compiler is written in CoffeeScript.

Titanium Mobile

http://blog.eddie.com.tw/2011/08/03/using-coffeescript-in-titanium-studio/

Immature?

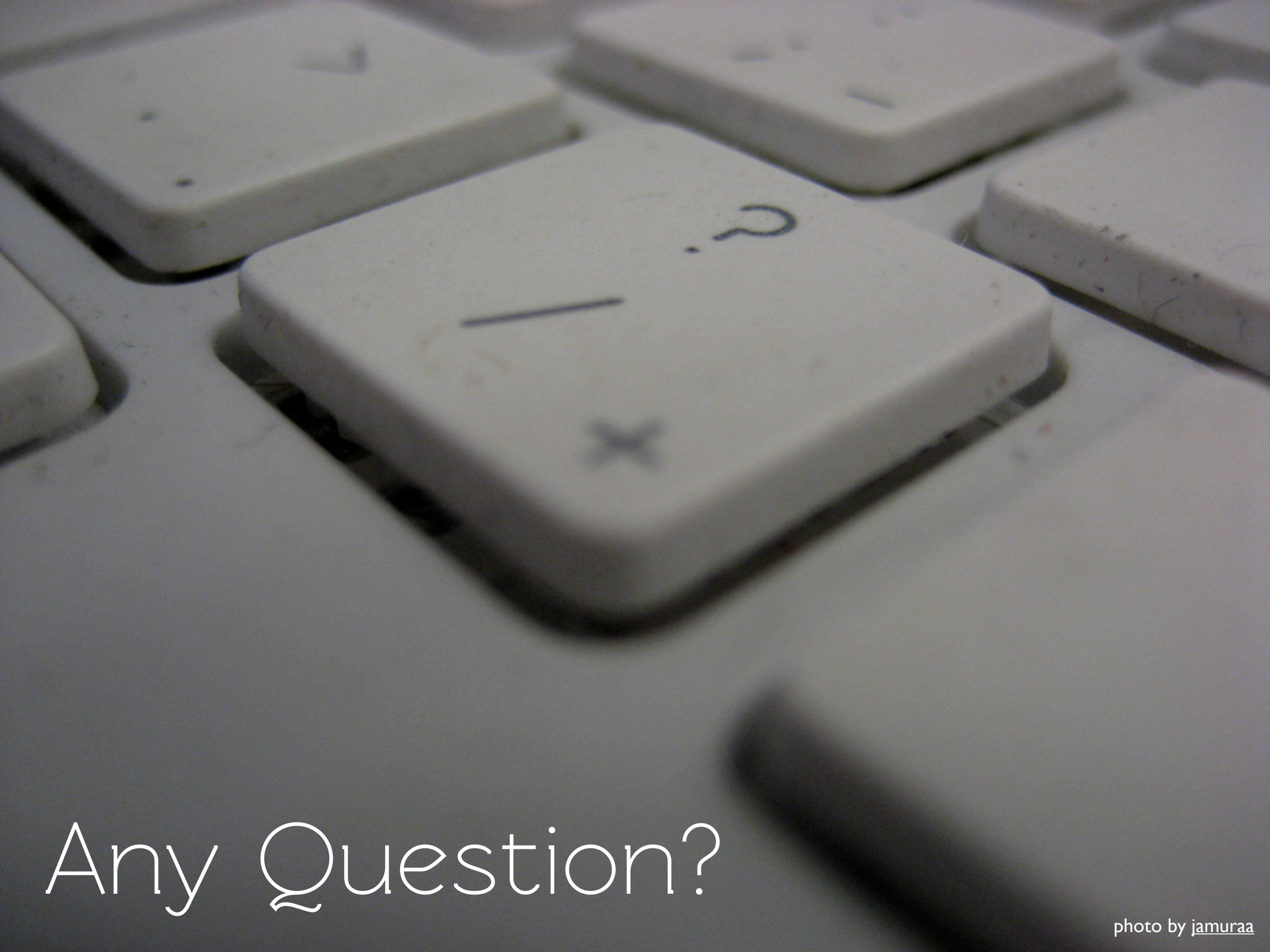
photo by theseanster93

photo by chr1sl4i

photo by cheesemaster23

CoffeeScript Means Giving Up on JavaScript?

Learn JavaScript, and Use CoffeeScript.



Any Question?

photo by jamuraa

Contacts

高見龍

- Website http://www.eddie.com.tw
- Blog http://blog.eddie.com.tw
- Plurk http://www.plurk.com/aquarianboy
- Facebook http://www.facebook.com/eddiekao
- Google Plus http://www.eddie.com.tw/+
- Twitter https://twitter.com/#!/eddiekao
- Email eddie@digik.com.tw
- Mobile +886-928-617-687

photo by Eddie