JavaScript for Pros

DevDay 2013

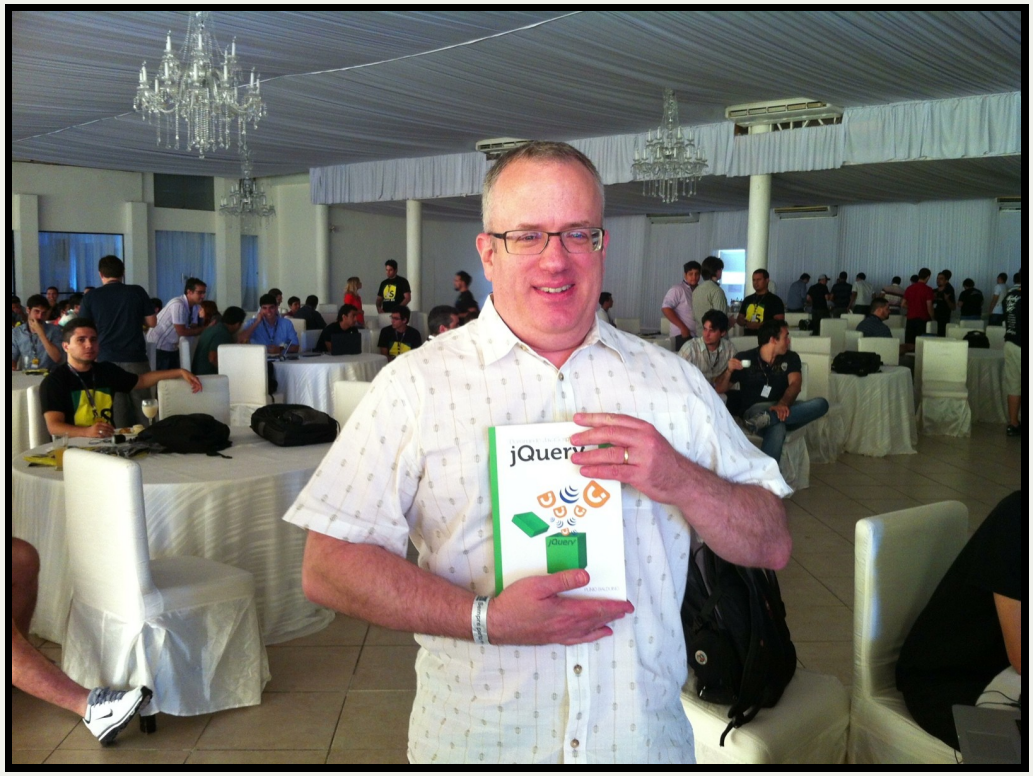
Belo Horizonte - 19/Out/2013

Plínio Balduino / @p_balduino

Plínio Balduino

A past century software developer

Brendan Eich



Developed JavaScript in 1995

Misconceptions

Not Java

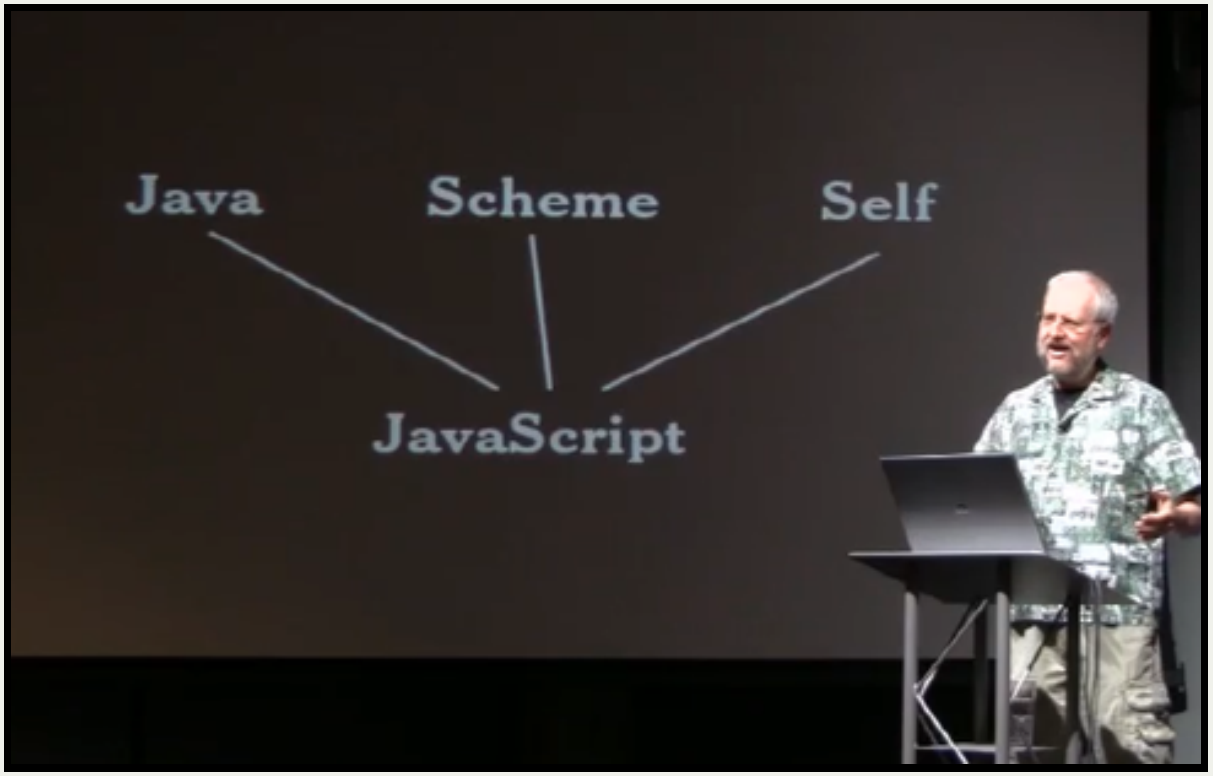
Not a toy language

Not web only

Rules of thumb

1. The clause `var` is not optional. Use it.
2. Sometimes `;` can be optional. Use it.
3. Prefer `===` and `!==` over == and `!=`
4. Don't blame the tool.

Origins



Definition

Dynamic
Object-oriented
Prototipe-based
Functional
Very flexible

Dynamic...

... like Ruby

```
var x = 1;

var y = "oh yeah!";

eval("alert('meh');");
```

Object-oriented

```
// JavaScript
function Dog() {

  this.bark = function() {
    return "WHOOF!"
  };
}

var rex = new Dog();

rex.bark();
```

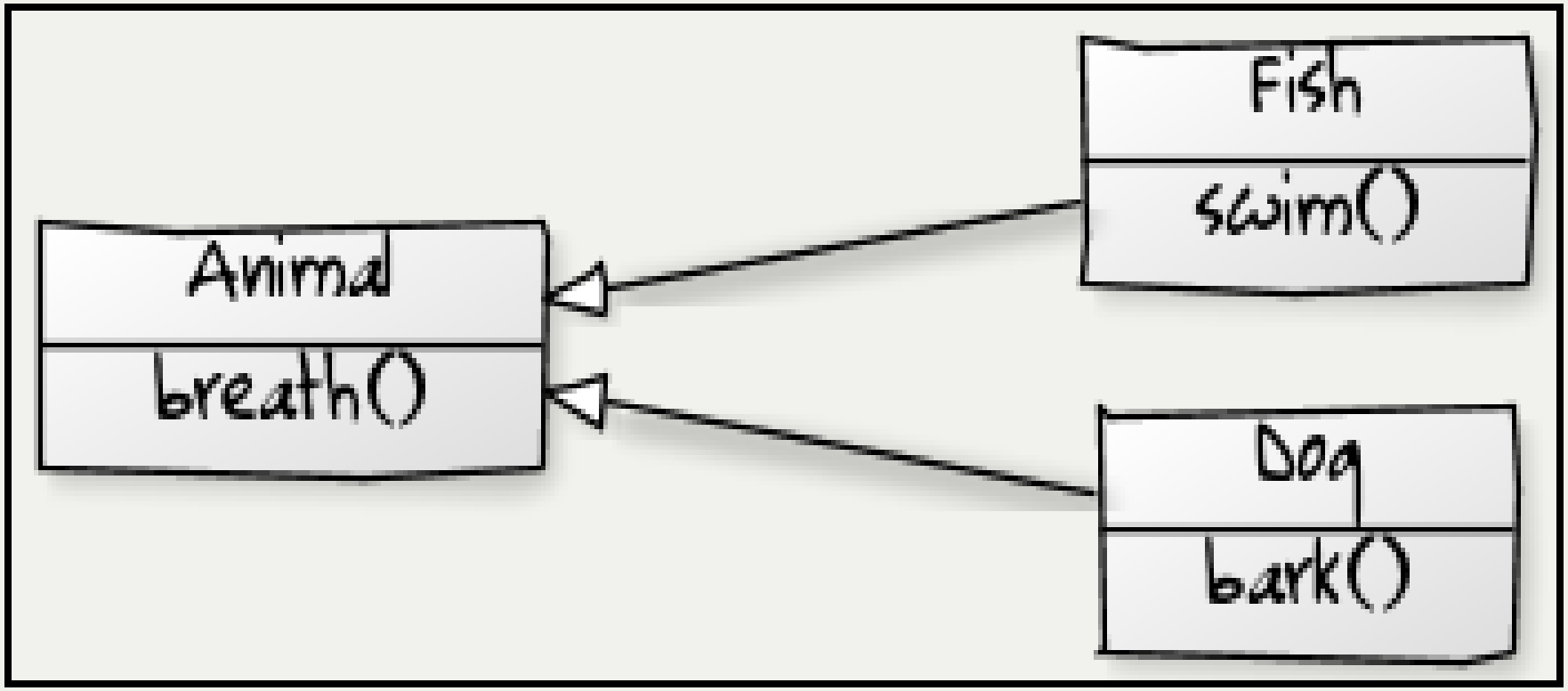
Object-oriented

```
// Java
public class Dog() {

  public String bark() {
    return "WHOOF!"
  };
}

Dog rex = new Dog();

rex.bark();
```

Prototype-based

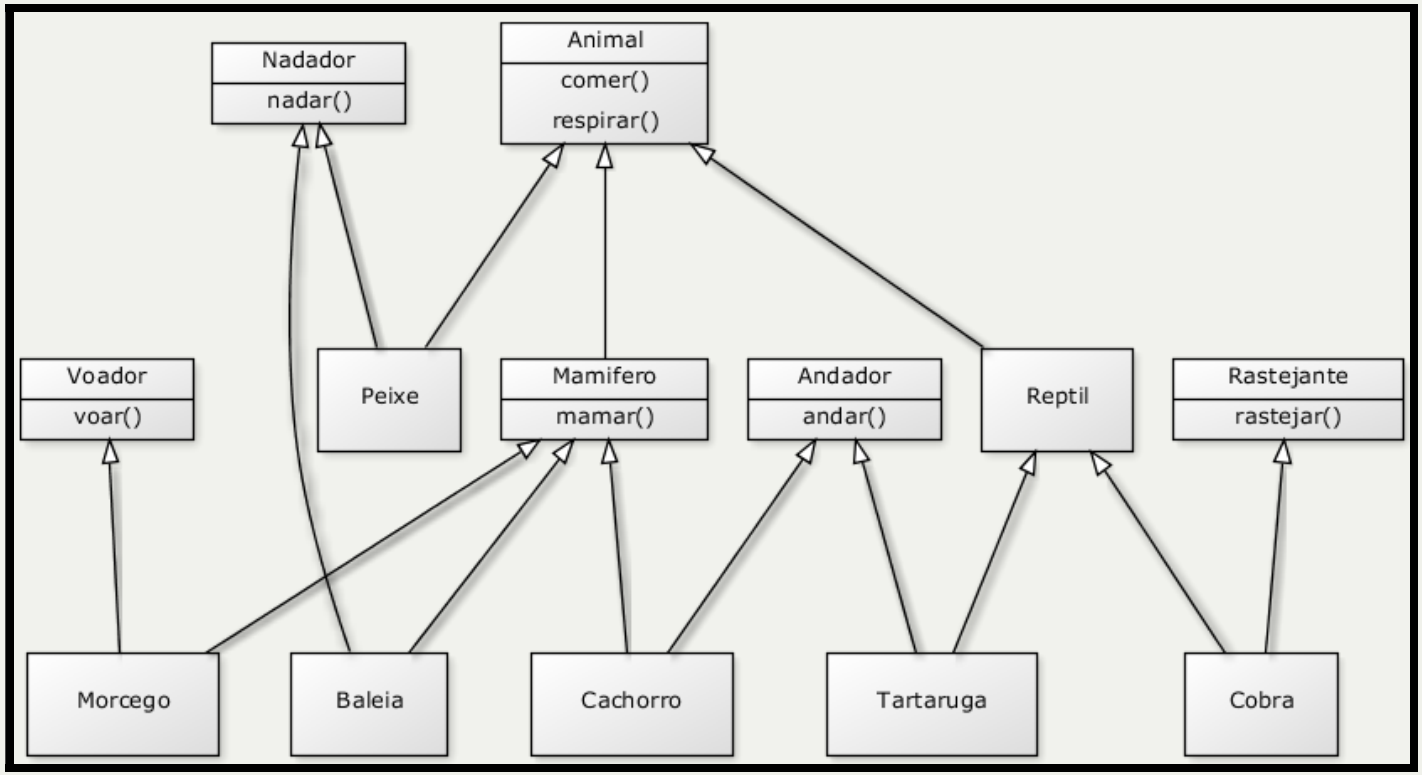
Because JavaScript is object-oriented
Not class-oriented

Classical inheritance



```
function Animal() {
  this.breath() {
    console.log("I'm breathing");
  }
}
function Dog() {
  this.bark() {
    console.log("WHOOOF!");
  }
}
function Fish() {
  this.swim() {
    console.log("GLOOB GLOOB!");
  }
}
Dog.prototype = new Animal();
Fish.prototype = new Animal();
```

Multiple inheritance



Mixin

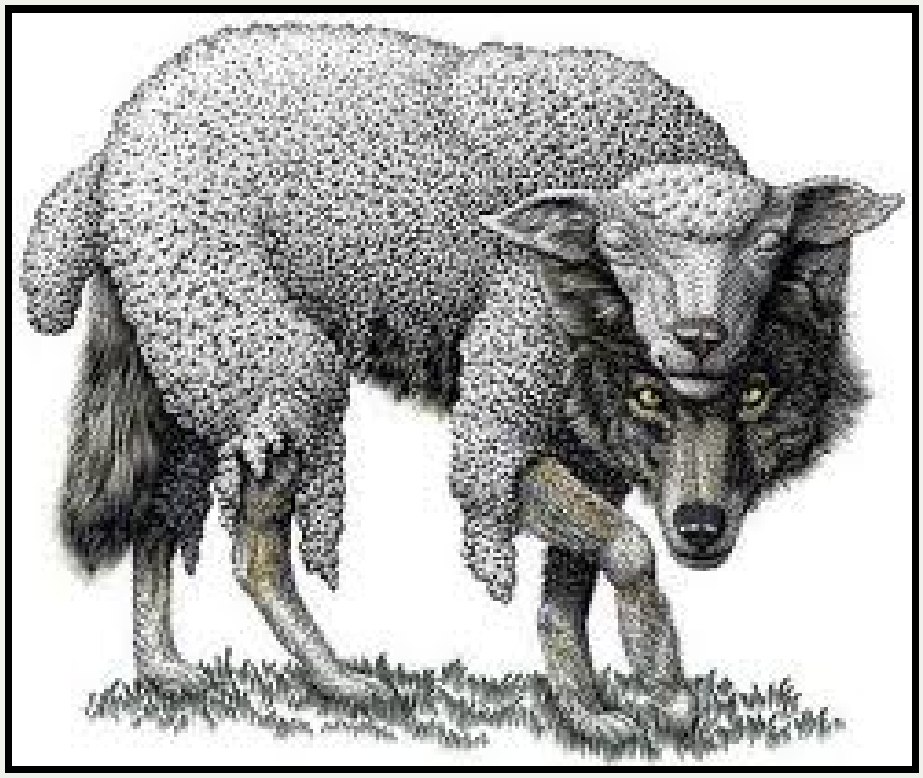
```
function Bubbles() {
  this.bubble() {
    console.log("Making bubbles");
  }
}
```

Mixin

```
Object.prototype.includes = function (constructo
  var obj = new constructor();
  for (var prop in obj) {
    if (obj.hasOwnProperty(prop)) {
      this.prototype[prop] = obj[prop];
    }
  }
};

Fish.includes(Bubbles);
var fish = new Fish();

fish.bubble();
```

Functional

JS is Lisp in C's Clothing (Douglas Crockford)



High Order Functions

```
; SCHEME
(define calculate (lambda (op v1 v2)
                          (op v1 v2)))

(define sum (lambda (a b) (+ a b)))

(define mult (lambda (x y) (* x y)))

(calculate sum 2 3) ; retorna 5

(calculate mult 2 3) ; retorna 6
```

High Order Functions

```
// JavaScript
function calculate(op, v1, v2) {
  return op(v1, v2);
}

function sum(a, b){
  return a + b;
}
function mult(x, y){
  return x * y;
}

calculate(sum, 2, 3); // retorna 5

calculate(mult, 2, 3); // retorna 6
```

jQuery uses all the time

```
$(function(){
  console.log("DOM loaded")
});

$(".clickable").click(function(){
  console.log(this + " was clicked")
});
```

Currying

```
function hey(text, name) {
  console.log(text + ", " + name);
}

hey("Bom dia", "João");

hey("Bom dia", "José");

hey("Bom dia", "Nicolau");
```

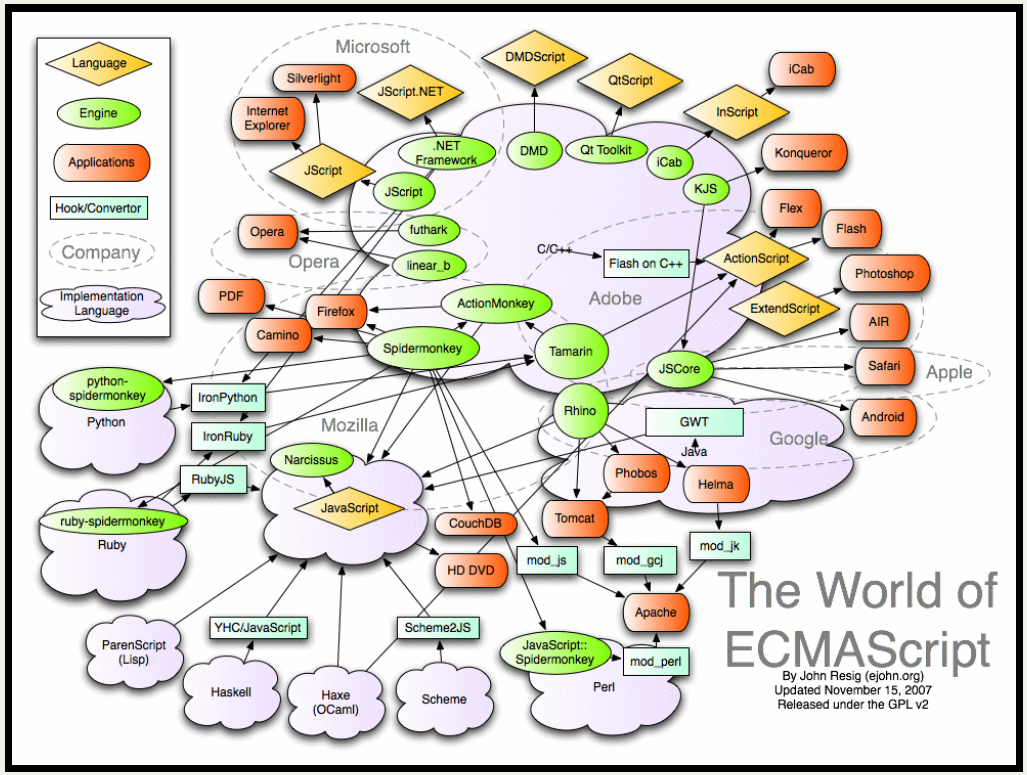
Currying

```
function hey(text) {
  return function(name) {
    console.log(text + ", " + name);
  }
}

var bomDia = hey("Bom dia");

bomDia("João");

bomDia("José");

bomDia("Nicolau");
```

Error handling

```
try{

  cause_an_error();

} catch(err) {

  console.log("Ooops: ", err.description);

};
```

JavaScript is everywhere



Why not Zoidberg?

Thank you

http://pbalduino.github.io/tdc2013/