Javascript Orientado a Objetos

Christiano Milfont

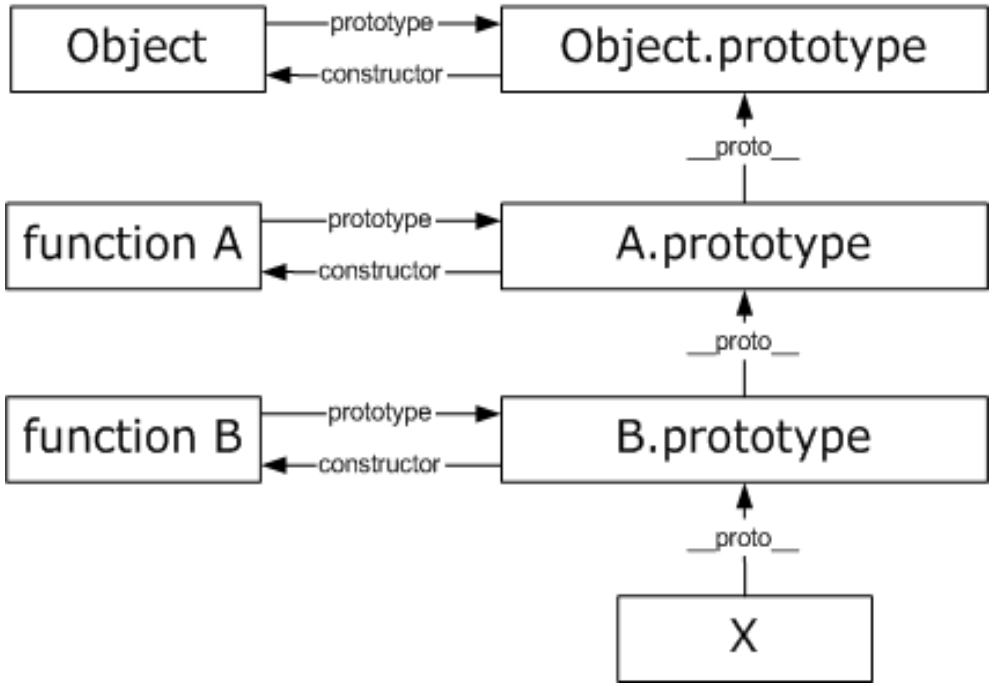
Introdução

"Procedural code gets information then makes decisions. Object-oriented code tells objects to do things."
Alec Sharp

```
function oldSchool() {
    this.getNome = function(variableString){
        if(variableString == 'Milfont'){
            return 'Permissão concedida';
        } else {
            return 'Permissão negada';
        }
    }
}
var oldies = new oldSchool();
alert(oldies.getNome('Milfont'));
```

```
var newSchool = function(){
    return {
        getNome: function(variableString){
            if(variableString == 'Milfont'){
                return 'Permissão concedida';
            } else {
                return 'Permissão negada';
            }
        }
    };
}();
alert(newSchool.getNome('Milfont'));
```

TriadWorks

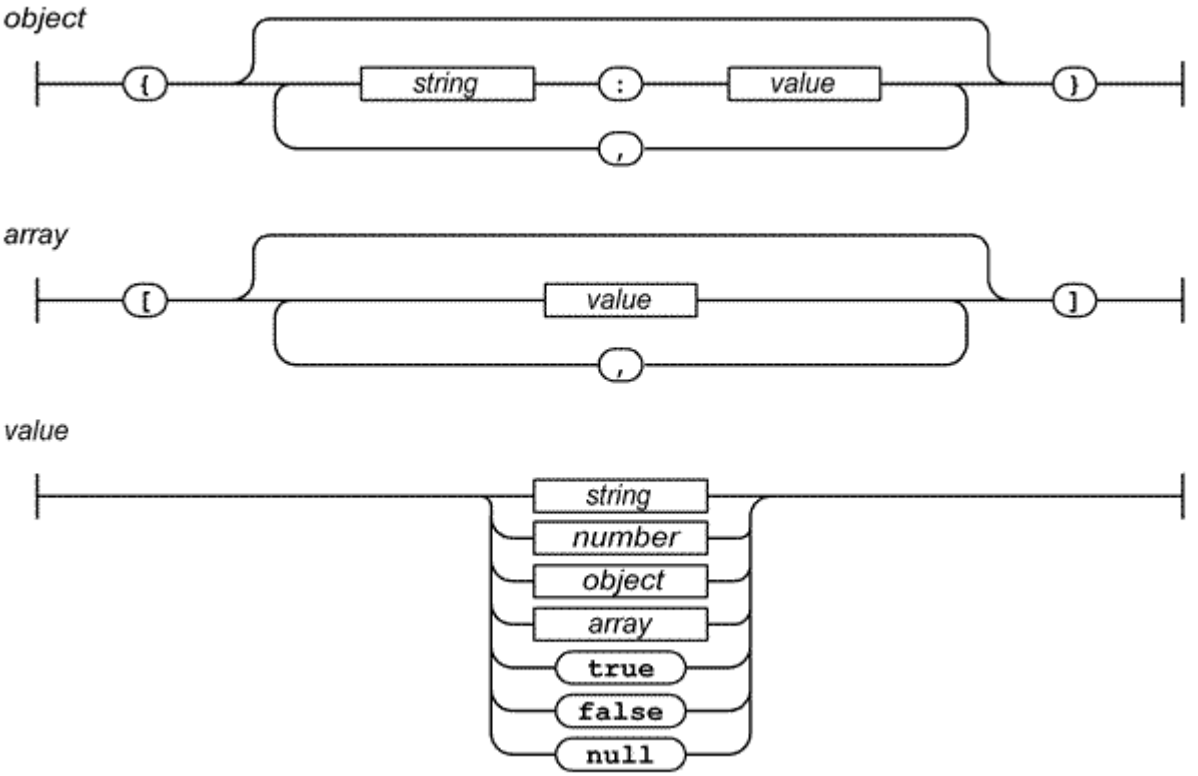


Prototype-based

```
if (typeof TRIADWORKS == "undefined") { var TRIADWORKS = {}; }
TRIADWORKS.prototype.MemoryProxy = function(data){
    TRIADWORKS.MemoryProxy.superclass.constructor.call(this);
    this.data = data;
};
```

TriadWorks

JSON (Javascript Object Notation)




```
var obj = {
    a : 'variavel string',  // Uma vari
    b : ['','bj2','three',4,'obj5'],  // Uma variável que recebe um array
    c : function() {  // uma função
        alert(this.a);
    }
}
//Como em javascript a declaração já é a instanciação,
// podemos fazer referencia direta ao objeto criado.
obj.c();        //Reposta: "variavel string"
alert(obj.a);   //Reposta: "variavel string"
```

TriadWorks

Factory Vs Construtor Vs Prototype

```
function createObject()  {
    var temp = new Object;
    temp.name = "Triadworks";
    temp.value = '$10000000';
    temp.show = function(){
        alert("Os mais bonitos");
    }
    return temp;
}
var temp1 = createObject();
var temp2 = createObject();

function createObject(name, value)  {
    var temp = new Object;
    temp.name = name;
    temp.value = value;
    temp.show = function(){
        alert("Os mais bonitos");
    }
    return temp;
}
var temp1 = createObject('Milfont','$10000000');
var temp2 = createObject('Chris','$10000000');
```

```
function Triadworks(name, value){
    this.name = name;
    this.value = value;
}
var temp1 = new Triadworks("Handerson", "-1");

function Triadworks(){};
Triadworks.prototype.name = '';
Triadworks.prototype.value = '';
Triadworks.prototype.setValue = function(value){
    this.value = value;
}
```

TriadWorks

Private vs Public

```
function objetoY()  {
    var variavel_privada = "Essa variável é privada";
    this.variavel_publica = "Essa variável é pública";
}

var ExemploFLISOL = function(){
    this.itemPublico = 'ok!';
    var itemPrivado = "nao pode acessar";
    return {
        getItemPrivado:function(){

        }
    }
}();
var exemplo = new ExemploFLISOL();
exemplo.getItemPrivado();
```

TriadWorks

Polimorfismo

```
//polimorfismo
var shape = new Shape();
var circle = new Circle( 10 );
var rectandle = new Rectangle( 10, 20 );
shape.Draw(); // Alert: Drawing generic shape
circle.Draw(); // Alert: Drawing circle
rectangle.Draw(); // Alert: Drawing rectangle
//O método Draw() é polifórmico porque tem diferentes
// características para invocação de cada objeto.
```

TriadWorks

Associação

```
//associacao
function Associacao(){
    this.composicao = new Passaro();
    this.agregacao = function(){
        return new Passaro();
    }
}

var teste = new Associacao();
alert(teste.composicao.valor);
alert(teste.agregacao().valor);
```

TriadWorks

Overloading e Overriding

```
var Objeto = new function(){
    return {
        teste: function(){
            alert("Sem parametros");
        },
        teste: function(x, y){
            alert(x + ' overrindig ' + y);
        }
    }
}();
//A execução do metodo teste mesmo
//que não contenha parâmetros apresentaria:
//"undefined overriding undefined".
```

TriadWorks

Alternativa de Overloading

```
var Objeto = new function(){
    return {
        metodo: function(){
            var argumentos = arguments.length;
            switch(argumentos){
                case 0 : alert('Function sem argumentos'); break;
                case 1 : alert(arguments[0]); break;
                case 2 : alert(arguments[0] + ' - ' + arguments[1]); break;
            }
        }
    }
}();
```

TriadWorks

Herança via Object Masquerading

```
function A(sColor) {
    this.color = sColor;
    this.sayColor = function () {
        alert(this.color);
    };
}

function B(sColor, sName) {
    this.newMethod = A;
    this.newMethod(sColor);
    delete this.newMethod;
    this.name = sName;
    this.sayName = function () {
        alert(this.name);
    };
}
```

```
var objA = new A('red');
var objB = new B('blue', 'Milfont');
objA.sayColor(); //outputs 'red'
objB.sayColor(); //outputs 'blue'
objB.sayName(); //outputs 'Milfont'
alert(objB instanceof A ); //false
alert(objB instanceof B ); //true
```

TriadWorks

Herança via call e apply

```
function nome(sPrefixo, sSufixo) {
    alert(sPrefixo + this.nome + sSufixo);
};
var obj = new Object();
obj.nome = 'Milfont';
nome.call(obj, 'O ', ' é o mais elegante. ');
nome.apply(obj, new Array('O ', ' é o mais elegante. '));
```

call(Objeto, argumento1, argumento2, ...)
apply(Objeto, array-de-argumentos)

TriadWorks

Herança via call e apply - cont.

```
function A(sNome) {
    this.nome = sNome;
    this.digaNome = function () {
        alert(this.nome);
    };
}

function B(sNome, sName) {
    A.apply(this, arguments);
    this.sobrenome = sName;
    this.digaSobrenome = function () {
        alert(this.sobrenome);
    };
}
```

```
var objA = new A('martins');
var objB = new B('Christiano', 'milfont');
objA.digaNome();  // 'martins'
objB.digaNome();  // 'Christiano'
objB.digaSobrenome(); // 'milfont'
```

TriadWorks

Herança via Prototype

```
//herança
function Passaro(){
    this.nome = 'papagaio';
}
function Produto(){
    this.valor = 'R$10,00';
}
Passaro.prototype = new Produto;
Passaro.prototype.imprimir = function(){ return this.valor + ' - ' + this.nome;}
var passaro = new Passaro();
alert(passaro instanceof (Produto)); // saida: true
alert(passaro instanceof (Passaro)); // saida: true
alert(passaro.imprimir()); // saida: "R$10,00 - papagaio"
```

TriadWorks