JavaScript: agora é sério

- Quem sou eu:
  - Luciano Ramalho, programador desde 1978
  - Desenvolvedor Web desde 1994
  - Entusiasta de linguagens
    - Python, Ruby, Scheme, Java, PHP, JavaScript, Perl, C, C++, Pascal, Smalltalk, Tcl/Tk, Processing, BASIC, VB, HyperTalk, Lingo, Assembly Z-80, 8086, HP-25, TI-58...
  - Usando principalmente Python desde 1998
  - Experimentando JavaScript em aplicações com banco de dados CouchDB desde 2010
    - E gostando!

JavaScript, a linguagem

- Gramática
  - Sintaxe: como se constroi comandos corretos
    - Quem não sabe comete erros sintáticos
  - Semântica: significado (”paradigma”)
    - Quem não sabe comete erros lógicos e sub-utiliza a linguagem
  - Léxico: vocabulário (funções, bibliotecas)
    - Quem não sabe reinventa a roda
- Pragmática: linguagem em seu contexto de uso
  - Ambientes (cliente, servidor, embutida)
  - Ecosistema: ferramentas, frameworks

JavaScript: um nome maldito

- Não tem nada a ver com Java
  - Foi uma decisão de marketing
  - Ambas imitam a sintaxe de C e C++
  - Não é um Java simplificado, mas essa percepção complica entender a semântica de JavaScript
  - Marca pertencia à Sun, agora pertence à Oracle
  - Sun autorizou Netscape a usar a marca, Mozilla Foundation deve ter herdado esta autorização
- ECMAScript
  - European Computer Manufacturers Association
  - Padrões: ECMAScript 3, ECMAScript 5

JavaScript: um nome maldito

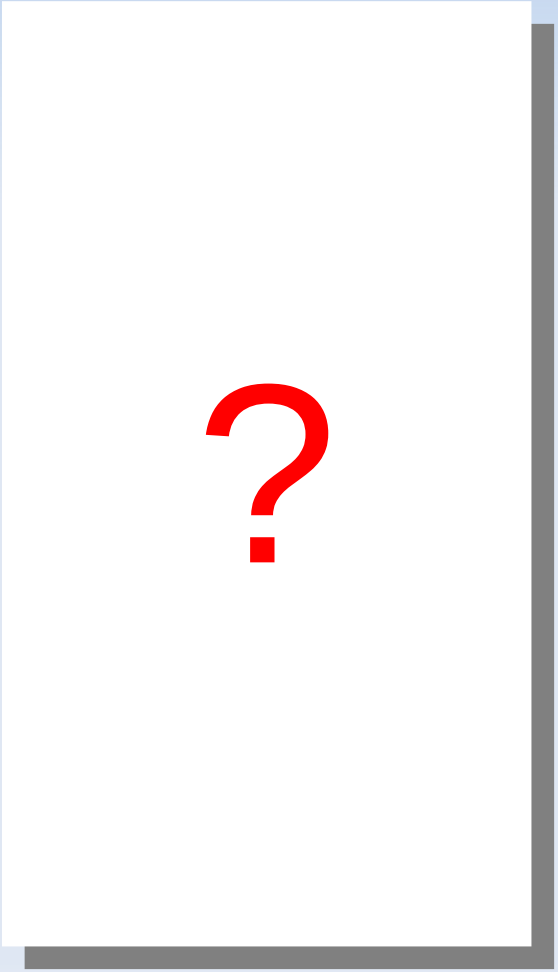
- Reunião de equívocos de outras linguagens de scripting (Perl, a fonte; PHP bebeu na mesma)
  - Variáveis automáticas
    - Esconde erros difíceis de localizar
  - Sintaxe imitando C, mesmo quando não faz sentido
    - Laço for (_;_;_)
  - Sem noção de módulos, nem mesmo include
  - Sem noção de namespace
  - Tipagem fraca = conversão automática

Python não tem estes problemas

Tipagem fraca

- Conversão automática é coisa do demo

```
'' == '0'
0 == ''
0 == '0'
false == 'false'
false == '0'
false == undefined
false == null
null == undefined
' \t\r\n ' == 0
```



Tipagem fraca

- Conversão automática é coisa do demo

```
'' == '0'           // false
0 == ''             // true
0 == '0'            // true
false == 'false'    // false
false == '0'        // true
false == undefined  // false
false == null       // false
null == undefined   // true
' \t\r\n ' == 0     // true
```

Tipagem fraca

- Conversão automática é coisa do demo

```
'' == '0'                 // false
0 == ''                   // true
0 == '0'                  // true
false == 'false'          // false
false == '0'              // true
false == undefined        // false
false == null             // false
null == undefined         // true
r\n ' == 0                // true
```

Python não tem estes problemas

JavaScript: o lado bom

- O melhor do scripting
  - Gerenciamento automatico de memória
  - Objetos nativos com sintaxe conveniente e expressiva
    - Arrays (como listas)
    - Objects (como dicionários ou arrays associativos)
    - Expressões regulares (integradas como em Perl)
    - Funções e closures (como em Scheme)

Funções de primeira classe

- First-class functions
  - Como em ”First-class citizens”
    - Cidadãos que gozam de todos os direitos civis
  - Funções como objetos de primeira classe
- Uma função pode ser:
  - Criada em uma expressão, em tempo de execução
  - Atribuída a uma variável
  - Inserida em uma estrutura (array, object)
  - Passada como parâmetro para outra função
  - Devolvida como resultado de uma outra função

Funções de ordem superior

- Higher-order functions
  - aceitam outras funções como argumentos
- Exemplos clássicos:
  - Map
  - Filter
  - Reduce
- Criação dinâmica de funções

Funções de primeira classe

- Aplicações:
  - Callbacks
  - Sistemas orientados a eventos
  - Programação assíncrona
  - GUIs
  - Clientes e servidores de protocolos de redes altamente escaláveis
- Conceito tão poderoso que graças a ele muitas das limitações conceituais de JavaScript podem ser superadas ou contornadas

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox

Print/export

Languages
Afrikaans
العربية
Беларуская
Беларуская (тарашкевіца)
Български
Català
Чăвашла
Česky
Dansk
Deutsch
Eesti
Ελληνικά
Español
Esperanto
Euskara

JavaScript

From Wikipedia, the free encyclopedia

Not to be confused with Java (programming language).

For the use of JavaScript on Wikipedia, see Wikipedia:JavaScript.

JavaScript is an implementation of the ECMAScript language standard and is typically used to enable programmatic access to computational objects within a host environment. It can be characterized as a prototype-based object-oriented[6] scripting language that is dynamic, weakly typed and has first-class functions. It is also considered a functional programming language[1] like Scheme and OCaml because it has closures and supports higher-order functions.[7]

JavaScript is primarily used in the form of client-side JavaScript, implemented as part of a web browser in order to provide enhanced user interfaces and dynamic websites. However, its use in applications outside web pages—for example in PDF-documents, site-specific browsers and desktop widgets—is also significant.

JavaScript uses syntax influenced by that of C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages.[8]

Contents [hide]

JavaScript

Paradigm	Multi-paradigm: scripting, prototype-based, imperative, functional[1]
Appeared in	1995
Designed by	Brendan Eich
Developer	Netscape Communications Corporation, Mozilla Foundation
Stable release	1.8.2[2] (June 22, 2009; 18 months ago)
Preview release	1.8.5[3] (July 27, 2010; 5 months ago)
Typing discipline	dynamic, weak, duck
Major implementations	KJS, Rhino, SpiderMonkey, V8, WebKit
Influenced by	C, Scheme, Java, Perl, Python, Self
Influenced	JScript, JScript .NET, Objective-J, TIScript

JavaScript at Wikibooks

Pragmática: novos usos

- AJAX: XMLHttpRequest e JSON
- HTML5
  - Armazenagem no cliente
  - Canvas
- Plug-ins para navegadores e outros aplicativos
- Node.js
  - Plataforma de programação assíncrona
- TiddlyWiki
  - Wiki pessoal, inovador, roda todo no cliente (sugestão de um participante na CampusParty 2011)

Mais usos

- Apache CouchDB
  - BD NoSQL e plataforma CouchApp
- Common.js
  - Uma biblioteca padrão de uso geral
- Aptana Jaxer
  - Servidor para aplicações AJAX
    (sugestão de um participante na CampusParty 2011)

Frameworks

- JQuery – DOM, UI
- YUI
- Prototype
- Dojo
- Mootools
- Closure

Ferramentas

- Consoles
- Debuggers
- Selenium
- QUnit
- Node.js

Vamos continuar esta conversa

- E-mail:
  - luciano@ramgarlic.com
- Grupo de discussão
  - http://groups.google.com/group/jspro
- Referências
  - JavaScript: the good parts (Douglas Crockford)
  - JavaScript patterns (Stoyan Stefanov)
  - 123 links que me interessaram:
    - http://www.delicious.com/ramalho/javascript