



JavaScript en dehors du navigateur

Bruno Michel <bruno.michel@af83.com>

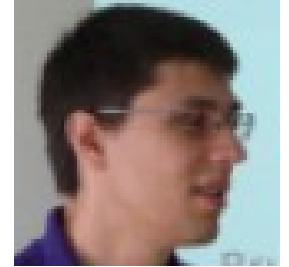
42 bd de Sébastopol
75003 Paris

1436 "A" Howard Street,
San Francisco, California 94103

contact@af83.com
www.af83.com



Moi



 Lead Dév chez af83

 Webmestre de LinuxFr.org

 Membre du CA de Ruby France

 Adhérent April

- 🌀 Créé en 1995 par Breindan Eich
 - Pour Netscape
 - Contexte : la première guerre des browsers
- 🌀 Implémenté dans Internet Explorer en 1996
 - Sous le nom JScript
- 🌀 Standardisé en 1997
 - Sous le nom ECMAScript

Self

- Langage objet à prototypes

Scheme

- Fonctions, objets de premier ordre
- Typage Faible

Java

- La syntaxe
- Le nom

Perl

- Les regexps

Learn to Use Your Web Pages

5th Edition
December 14



JavaScript

The Definitive Guide

O'REILLY*

David Flanagan

Mastering the Fundamentals of JavaScript



JavaScript: The Good Parts

O'REILLY*

YAHOO! PRESS

Douglas Crockford

❌ Mauvaise réputation

- Le DOM
- La sécurité

❌ Pas de module => variables globales

❌ Pas de bibliothèque standard

❌ Des comportements bizarres / dangereux

- `typeof null == 'object'`
- `'' == 0 ; 0 == '0' ; '' != '0'`
- Confusion entre NaN, null et undefined
- Les opérateurs ++ with eval


Lambda

Langage très dynamique

- Typage faible
- Objets dynamiques pouvant être modifiés à la volée

Syntaxe agréable

- `{un: 'objet'} ['tableau'] /regexp/`
- JSON

 <http://googlecode.blogspot.com/2009/03/doug-crockford-javascript-good-parts.html>


- Popularité, mais mauvaise réputation
- Un des langages les plus utilisés au monde
- Un composant indispensable du web 2.0 (AJAX)
- Des frameworks
 - jQuery, Prototype, YUI, Dojo, ExtJS...
- Des implémentations de + en + rapides
 - Rhino, SpiderMonkey, TraceMonkey, SquirrelFish, V8

Rapidement utilisé pour scripter des applications

- Adobe : Acrobat & Acrobat reader
- Photoshop, Illustration, Dreamweaver
- QtScript pour scripter QT
- OpenOffice.org

Mais forte concurrence de lua

- Plus rapide
- Plus économe en mémoire

 Java (Rhino)

 PHP

- PHPJS et J2P5: Pure-PHP ECMAScript Engine

 Perl

- JE et Javascript::Spidermonkey

 Python

- Python-Spidermonkey

 Ruby

- Johnson, Red, Lyndon

🌀 Netscape en faisait déjà en 1996 !

- Netscape Entreprise Server 2.0

🌀 Base de données NoSQL

- CouchDB
- JSON

🌀 Proxy / Serveur applicatif

- Opera Unite
- Apache Sling

🌀 Et ...

 Quelques essais ces dernières années

- Helma
- Torino (2006)
- JSSP (2007)
- **Aptana Jaxer** (2007)
- AppJet (2007 - 2009)
- Et beaucoup de bricolages (sourceforge)

 Une recrudescence ces derniers mois

- (ActiveJS)
- V8cgi
- Flusspferd
- v8-juice
- ESXX
- Helma-ng
- Persevere
- ChironJS
- Node.js
- Narwhal
-

29/01/2009 : What Server Side JavaScript needs

- <http://www.blueskyonmars.com/2009/01/29/what-server-side-javascript-needs/>
- Constat : il manque un écosystème JS
 - Une bibliothèque standard
 - Une manière unique de charger des modules
 - Packaging pour la distribution et le déploiement
- Création d'une ML ServerJS

- Renommé en CommonJS
 - <http://wiki.commonjs.org/>
- Un processus pour les propositions (~ PEP / RFC)
- Des conventions de codage
 - <http://javascript.crockford.com/code.html>
- JSGL : une interface serveur web → application
 - Jack : <http://jackjs.org/>

🌀 Modules interopérables

- <http://wiki.commonjs.org/wiki/Modules/SecurableModules>

🌀 7 modules

- Binary, encodings, IO, file, system, unit testing, socket IO

🌀 D'autres modules à venir

- API bas-niveau : logging, interfaces RDBMS, etc.
- API haut-niveau : HTTP Client, Email, XMPP, etc.

🌀 Des discussions sur le packaging

```
Terminal - Terminal
nono@1000HE:~% On installe narwhal à partir de http://github.com/tlrobinson/narwhal
nono@1000HE:~% export PATH=$PATH:~/narwhal/bin
nono@1000HE:~% cat narwhal/examples/hello
#!/usr/bin/env narwhal
print("Hello, World!");
nono@1000HE:~% narwhal narwhal/examples/hello
Hello, World!
nono@1000HE:~% █
```

```
Terminal - Terminal
nono@1000HE:~% tusk init hello-web
/home/nono/hello-web
nono@1000HE:~% cd hello-web
nono@1000HE:~/hello-web% ./bin/sea
PATH=/home/nono/hello-web/bin:/home/nono/narwhal/engines/rhino/bin:/home/nono/narwhal/bin:/home/nono/bin:/home/nono/gem/bin:/opt/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/home/nono/Desktop/android/tools:/home/nono/narwhal/bin
SEA=/home/nono/hello-web
SEALVL=1
nono@1000HE:~/hello-web% tusk install jack 2>&1 | less
nono@1000HE:~/hello-web% vim jackconfig.js
nono@1000HE:~/hello-web% ~/.vim/macros/cat.sh jackconfig.js
exports.app = function(env) {
  var text = "Hello, Web!";
  return {
    status : 200,
    headers : { "Content-Type" : "text/plain", "Content-Length" : String(text.length) },
    body : [text]
  };
};

nono@1000HE:~/hello-web% jackup
Loading configuration module at /home/nono/hello-web/jackconfig
Jack is starting up using Simple on port 8080
```

```
Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils Aide
http://localhost:8080/
narwhal - quick start x http://localhost:8080/ x
Hello, Web!
```

Bogart

```
app.js (~/.osdc-fr) - GVIM
var Bogart = require("bogart").App;
var URLMap = require("jack").URLMap;

var hello = new Bogart(function() {
  this.GET("/:name", function() {
    return this.text("hello " + this.params["name"]);
  });
});

var calc = new Bogart(function() {
  this.GET("/add/:first/:second", function() {
    var first = parseInt(this.params["first"]);
    var second = parseInt(this.params["second"]);
    return this.text("addition result: " + (first + second).toString());
  });
});

exports.app = URLMap({
  "/hello": hello,
  "/calc": calc
});

~
~
app.js
"app.js" 22L, 536C
```



Slides bientôt disponibles sur <http://dev.af83.com/>
Licence : CC by-sa 3.0+
<http://creativecommons.org/licenses/by-sa/3.0/deed.fr>

Paris Office
42, bd Sébastopol
73 003 Paris
France
+33 1 40 27 83 83

San Francisco Office
1436 "A" Howard Street,
San Francisco, Californi 94103
United States
+1 415 578 4664

www.af83.com
contact@f83.com