

The REBOL Documentation Project

-- FR - Documentation REBOL - Articles Techniques --

Articles Techniques

Ruby vs REBOL

REBOLtof

Première publication : 18 novembre 2005, et
mis en ligne le vendredi 18 novembre 2005

Résumé :

Cet article est une traduction du message posté par Gregg Irwin sur la mailing list anglophone : [Ruby vs REBOL](#).

Cet article est une traduction du message posté par Gregg Irwin sur la mailing list anglophone : [Ruby vs REBOL](#).

Version : 1.0

Date : 18-nov-2005

- [1 Historique](#)
- [2 Comparaison des fonctionnalités](#)
- [3 Du point-de-vue du programmeur pragmatique](#)

1 Historique

Date	Version	Commentaires	Auteur	Email
18-nov-2005	1.0	initial	REBOLtoF	reboltoF—at—yahoo—dot—com

2 Comparaison des fonctionnalités

- ▶ Ruby est un langage de programmation destiné aux programmeurs. REBOL est un langage utilisé afin d'échanger de l'information. Et parfois, cette information est un script que l'ordinateur doit exécuter.
- ▶ Ruby a environ 40 mots réservés. REBOL n'en a pas.
- ▶ Ruby a environ 7 types de données. REBOL en a plus de 30. Contrairement à REBOL, Ruby dispose de nombreuses classes standards (p.e bignums).
- ▶ Certaines fonctionnalités qui sont intégrées à REBOL (par exemple l'accès au réseau), ne sont disponibles que comme librairies pour Ruby.
- ▶ Ruby est sensible à la casse et utilise des signets (NdT : je n'en ai pas retrouvé trace dans la doc de Ruby, et je ne vois pas par quoi d'autre le traduire : sigil = "A seal, a signet or A sign or an image considered magical"). REBOL n'est pas sensible à la casse et les mots (variables) peuvent contenir tous les caractères à l'exception de quelques cas particuliers.
- ▶ Ruby supporte un système de blocs et d'itérateurs très pratique. En REBOL, tout est donnée. L'évaluation d'une donnée peut vous faire penser que vous êtes en train d'écrire du code.
- ▶ Ruby est purement orienté objet et a des "mixins" (NdT : inclusion d'un module dans la définition d'une classe) plutôt qu'un système d'héritage multiple. REBOL crée des objets à partir de prototypes. Un objet en REBOL fournit un contexte et agit comme un espace de nommage.
- ▶ En Ruby, vous devez programmer en orienté objet (méthode d'objet). REBOL a été conçu afin de vous laisser exprimer des concepts de la manière que vous le désirez.
- ▶ Ruby a des variables incluses spéciales, comme Perl. REBOL n'en a pas.
- ▶ Ruby dispose d'expressions régulières (NdT : "regexes"). REBOL utilise la fonction PARSE.

3 Du point-de-vue du programmeur pragmatique

- ▶ Ruby dispose d'un débogueur et d'un profileur en standard. REBOL n'en a pas.

- ▶ REBOL dispose d'une fonction d'aide (NdT : "HELP") intégrée et d'une fonction permettant de visualiser les sources du code (NdT : "SOURCE"), avec le support de la réflexion (chaînes de documentation). Ruby utilise un outil séparé afin d'extraire la documentation incluse, qui est écrite dans un langage de formatage de texte léger.
- ▶ Vous pouvez écrire des extensions de Ruby en C. REBOL peut faire appel à des DLL.
- ▶ Comme fonctionnalité particulière, Ruby dispose de "Ruby on Rails" (NdT : un framework de développement, voir [rubyonrails](http://rubyonrails.org)). REBOL dispose du Rebcode (NdT : une sorte de bytecode pour REBOL, dont l'évaluation est jusqu'à 58 fois plus rapide, voir le blog de Carl).

Note du traducteur

la traduction du paragraphe précédent a été adaptée selon les dernières informations provenant de la mailing-list.

- ▶ REBOL/View a un système intégré de création d'interface utilisateur (composition en plein 32-bit pour chaque face, plus de 14 effets pipeline, engin de dessin AGG). Ruby doit utiliser un produit comme Tcl/Tk pour les interfaces utilisateur.
- ▶ Ruby est plus populaire que REBOL, et semble évoluer plus rapidement.
- ▶ REBOL peut créer des exécutables sans dépendances externes.
- ▶ Le REBOL/Core compressé ne prend que environ 190K. 350K pour View. L'installateur de Ruby pour Windows pèse environ 14.8M (il contient certains extra comme Tcl/Tk).
- ▶ Il n'est pas facile de comparer REBOL avec un langage de programmation, bien qu'il se défende sur ce terrain. C'est comme comparer des poires et des pommes... Il est vraiment nécessaire de comparer REBOL à Ruby+XML par exemple.
- ▶ Ruby est un mélange d'idées provenant d'autres langages de programmation, avec quelques nouveautés (sauf le respect de Matz. Ruby est ce que j'utiliserai si REBOL n'existait pas). REBOL est différent, et nous ne savons pas encore comment l'exploiter pleinement.
- ▶ Ruby est destiné aux programmeurs. REBOL est destiné aux humains. Ruby est meilleur si vous venez de Perl ou du monde orienté objet, et si vous ne voulez pas trop torturer votre esprit. Faites simplement les choses dans un langage similaire à ce que vous connaissez, avec quelques nouveautés.
- ▶ REBOL est parfait si vous souhaitez assouplir votre esprit et apprendre à penser d'une nouvelle manière. C'est également un excellent choix si vous désirez construire un langage spécifique à un domaine (NdT : "Domaine Specific Language - DSL"). Ce concept existe déjà depuis longtemps, mais est devenu une nouvelle mode dans le collimateur de MS et d'autres. Le système d'interface utilisateur intégré de REBOL, ainsi que sa petite taille, sont aussi des atouts non négligeables.