

The REBOL Documentation Project

-- FR - Documentation REBOL - Divers --

Divers

BIND étend le contexte

Philippe Le Goff

Première publication : 8 juin 2006, et mis en
ligne le jeudi 8 juin 2006

Résumé :

Traduction de l'article 26 du Blog de REBOL 3.0, au sujet de la modification de BIND

BIND étend les contextes

Carl Sassenrath, CTO REBOL Technologies 16-May-2006 23:28 GMT Article #0026

Voici de la matière à réflexion pour REBOL 3.0 :

```
obj : make object ! [a : 1] word : bind 'b obj set word 10 print obj a : 1 b : 10
```

La fonction bind non seulement lie le mot au contexte, mais actuellement l'y ajoute.

Ce résultat est la conséquence d'un effet de bord de ceci : REBOL 3.0 n'utilise plus un environnement de contexte global ; il utilise des contextes modulaires.

Cela signifie que l'expression :

```
word : make word ! "xyz"
```

crée un mot (symbole) 'xyz, mais pas une variable. C'est raisonnable parce qu'un symbole peut être autonome (c'est une chose utile), mais une variable exige un contexte. La fonction make word ! ne spécifie aucun contexte, de sorte qu'aucun contexte n'est pris en compte. Si vous essayez ceci :

```
set word 10 **Script Error : xyz word has no context (not bound)
```

Dans REBOL 3.0 le comportement par défaut pour bind est de lier le mot au contexte, indépendamment de la façon dont cela se produit (un comportement qui est exigé par les modules pour fonctionner, lorsque de nouvelles définitions de mots sont créées via des scripts complémentaires, etc.)

Quelques uns d'entre vous demanderont : pourquoi avoir fait cela ? La principale raison est la mémoire (quelque chose que nous optimisons en REBOL 3.0). Pourquoi imposer une allocation en mémoire pour des variables "free" (variables de module) qui ne sont jamais employées ?

Certains d'entre vous ont peut-être des remarques là-dessus, qu'ils me les signalent. Tout ceci n'est pas encore figé... pour l'instant.

(Traduction : Philippe Le Goff)