The REBOL Documentation Project

-- FR - Documentation REBOL - Divers --

Divers

Le concept des Set-Functions

Philippe Le Goff

Première publication : 27 mai 2006, et mis en ligne le samedi 27 mai 2006

Résumé :

Traduction de l'article 19 du blog REBOL 3.0.

Le concept des Set-Functions

Carl Sassenrath, CTO REBOL Technologies 2-May-2006 21:02 GMT Article #0019

Vous savez ce qu'est un set-word :

```
word : 10
```

C'est la syntaxe utilisée par REBOL pour définir la valeur d'un mot.

Une set-function est une fonction qui est appelée lorsque cette action de définition se produit. Une set-function dans l'exemple précédent aurait permis de contrôler le range, le datatype, de conserver la trace d'autres détails.

Le concept d'une set-function n'est pas nouveau. En fait, c'est l'un des éléments essentiels de l'un des aïeuls de REBOL : le langage Self (voir Self : The Power of Simplicity, l'un de mes livres préférés.)

Il y a longtemps, dans les premiers jours de création de Self, alors que j'étais chez Apple, je trainais avec David Ungar (concepteur de Self, à Standford actuellement) et je parlais de ce langage de communication. Je trouvais intéressant que David veuille retirer un état de Self en supprimant l'affectation. (L'affectation est l'une des cause d'ennuis classique dans les langages purement fonctionnels.) Il accomplissait toujours cela en utilisant des fonctions qui à la fois définissaient et récupéraient des valeurs au sein d'un contexte (objet).

L'idée me tentait toujours, mais je sentais que ce n'était pas exactement réalisable. L'état est un macro-composant par nature (je n'ai pas dit "composant" ici, parce que je ne veux pas entrer dans une discussion sans fin). La suppression de l'état est une recherche mathématique, mais ce n'est pas nécessairement pratique. Il y a de nombreux aspects utiles à un état. Un journal est très utile, mais mathématiquement sans intérêt fonctionnel.

Lorsque vint le temps de reprendre cette approche dans REBOL (il y plus de 10 ans), j'ai décidé de conserver cela pratique. Cependant, en secret, j'ai toujours voulu conserver la méthode set-function, car elle a beaucoup de valeur. Une set-function peut précisément contrôler comment un mot est défini et ce qu'il définit.

Je devrais mentionner, cependant, qu'il n'est pas facile d'ajouter les set-functions. Si c'était vraiment facile, nous les aurions ajoutées. Le problème réside dans le fait que les set-functions ajoute une nouvelle branche à l'arbre de la sémantique du langage REBOL.

Par exemple, comment sont définies les set-fonctions et comment sont-elles réflectives ?

Vous pouvez faire une sorte de set-function avec le code qui suit :

```
obj : make object ! [   value : 0 ; internal   set-value : func [new] [value : new]  ]
```

Et vous écririez :

```
obj/set-value 10
```

Mais ce que vous voulez réellement écrire, c'est :

obj/value : 10

à l'intérieur de la fonction pour définir cette valeur.

Mais là, la question se pose, comment faire référence à la set-function elle-même ?

OK, voilà quelque chose à réfléchir. Je ne suis pas sûr qu'il y ait une bonne réponse, ou aussi si nous avons besoin d'une. J'avais déjà parlé de cela autrefois, et sans suggérer que ceci soit ajouté à REBOL 3.0. (voir l'article précédent sur les extensions du langage.)

Je suppose que je suis en train de vous faire réfléchir à cela. Merci de me signaler vos commentaires.

(Traduction : Philippe Le Goff)