

The REBOL Documentation Project

-- FR - Documentation REBOL - Divers --

Divers

Un nouveau datatype : Typeset !

Philippe Le Goff

Première publication : 27 avril 2006, et mis
en ligne le jeudi 27 avril 2006

Résumé :

Traduction du post 2 sur le nouveau type de données apparaissant dans Rebol 3.0

Carl Sassenrath, CTO REBOL Technologies 8-Apr-2006 20:21 GMT Article #0002

"Cet article a été modifié depuis sa première version. Le mot "typemap" a été changé en "typeset"."

Dans REBOL 3.0, un typeset est essentiellement un ensemble de datatypes.

Par exemple, lorsque vous définissez une fonction, vous spécifiez de la manière suivante les types de données (datatypes) des arguments :

```
f: func [arg1 [integer! number!] arg2 [string! url!]] ...
```

Ce qui suit chaque mot définissant un argument (arg1, arg2,...) est un bloc de datatypes.

REBOL 3.0 ajoute un nouveau type de données appelé un typeset !. Un typeset est simplement une méthode compacte, efficace de stocker des ensembles de datatypes, comme une nouvelle sorte de type de données.

Les typesets sont importants car l'interpréteur les utilise pour valider rapidement les arguments des fonctions.

Le datatype typeset ! sait comment convertir un bloc contenant des noms de datatypes comme [integer ! decimal ! money !] en une représentation interne (similaire au datatype bitset !). Il peut aussi traduire ce format interne en un bloc compréhensible, en vue d'une édition ou de modifications.

L'ajout de ce datatype typeset ! élimine le recours aux pseudo-datatypes REBOL spéciaux que sont series ! et number !. Ils sont à présent implémentés comme des typesets, mais vous pouvez continuer à les utiliser de la même manière que précédemment.

Ils peuvent être utilisés dans les spécifications d'arguments de fonction :

```
f: func [arg1 [number!] arg2 [series!]] ...
```

et aussi pour des tests de datatypes :

```
if series? value [...]
```

Notez, cependant, cette différence avec REBOL 2.* :

```
type? series!  
datatype!
```

qui devient dans REBOL 3.0 :

```
type? series!  
typeset!
```

Pour les utilisateurs avertis, REBOL 3.0 fournira une méthode pour créer et gérer vos propres typesets.

```
tmp-type: make typeset! [time! money! percent!]
```

```
f: func [arg1 [tmp-type]] [...]
```

J'en dirai plus bientôt au sujet de quelques problèmes liés aux typesets.

(Traduction : Philippe Le Goff)