

The REBOL Documentation Project

-- FR - Documentation REBOL - Divers --

Divers

Les fonctions "closure"

Philippe Le Goff

Première publication : 9 mai 2006, et mis en
ligne le mardi 9 mai 2006

Résumé :

la traduction de l'article 12 du blog sur REBOL 3.0

Closure Functions

Carl Sassenrath, CTO REBOL Technologies 24-Apr-2006 16:32 GMT Article #0012

NdT : j'ai préféré laissé le terme anglais pour éviter les méprises.

Les fonctions closure seront fournies en standard dans REBOL 3.0.

Le concept des fonctions closure n'est pas neuf (au moins 30 ans d'âge), et peut être trouvé dans la plupart des types de langages fonctionnels. Une closure est un type particulier, ou une caractéristique, d'une fonction qui conserve "vivant" l'environnement local (arguments et variables locales) d'une fonction, même après que cette fonction ait terminé son exécution.

Avantages : l'environnement de la fonction (mots et données) reste valide après que la fonction ait retournée ses valeurs. (ce qui n'est pas vrai avec les fonctions normales). C'est pratique si vous voulez retourner des mots ou d'autres données de la fonction, et que vous voulez encore accéder à la valeur des mots.

Inconvénients : leur évaluation est plus lente que les fonctions normales car elles sont allouées et liées à l'exécution. Aussi, les fonctions closures requièrent souvent plus de mémoire (et provoquent plus de recyclage), selon l'usage que vous en faites et la façon dont vous stockez leurs résultats.

En REBOL 3.0, les fonctions closures seront implémentées avec le datatype closure ! et une nouvelle fonction mezzanine appelée closure. Voici un exemple qui définit une nouvelle fonction closure appelée has-name :

```
hash-name : closure [name [string !] /local hash] [ hash : checksum/secure name return [name hash] ; note : renvoie des mots, pas aux valeurs ]
```

Lorsqu'une fonction closure est définie, un objet REBOL est créé et le corps de la fonction est lié à cet objet. Lorsque la fonction est évaluée, l'objet de la fonction d'origine est cloné et le corps de la fonction copié complètement et relié à cet objet nouvellement cloné.

Vous appelez les fonctions closure de la même manière que vous le faites pour n'importe quelle fonction :

```
result : hash-name "Bob" print second result
```

Vous pouvez tester une fonction closure avec :

```
if closure ? :hash-name [print "It is true"]
```

et, cette expression est aussi vraie :

```
if any-function ? :hash-name [print "It is true"]
```

Pour en savoir plus à propos du concept des fonctions closures, voir l'item "Function Closure" sur le site [Wikipedia](http://fr.wikipedia.org), et également les exemples et méthodes de fonctions closures REBOL fournies par Ladislav Mecir (qui travaille sur les versions courantes de REBOL) dans ses tutoriels et codes ("Bind

a Function Yourself").

(Traduction : Philippe Le Goff)