

[REBOLZONE]

(c) 2005 RebolZone - <http://rebolzone.free.fr>

Start

Exécuter un script Rebol sous Unix (Linux ...)

Le script (.r) doit avoir les droits d'exécution : faire un : `chmod +x monscript.r`

1) 1ère solution : au prompt Unix (\$) ou autre) :

\$ `rebol monscript.r` (l'évaluateur doit être dans le PATH, /usr/bin par exemple)

2) 2ème solution : au prompt Unix (\$) :

\$ `./monscript.r` (ou `monscript.r` si le PATH inclu le répertoire courant)

Dans ce cas le script rebol doit inclure en première ligne l'appel à l'évaluateur Rebol : `#!/usr/bin/rebol` (dans ce cas, l'évaluateur est sous /usr/bin).

Exemple de script :

```
#!/usr/bin/rebol
rebol [
  Sujet: "Test ..."
  Auteur: "RebolZone"
  Version: 1.0.0
]
;Début de script
print "Ok"
input
```

Pour éviter les messages (copyright ...), on peut rajouter l'option -q de l'évaluateur Rebol.

Exemple :

```
#!/usr/bin/rebol -q
rebol [
  Sujet: "Test ..."
  Auteur: "RebolZone"
  Version: 1.0.0
]
;Début de script
print "Ok"
input
```

Exécuter un script Rebol sous Windows

Dans l'explorateur de fichiers, double-cliquer sur le script (.r), celui s'exécutera automatiquement en raison de l'association des fichiers de type .r. On peut également exécuter un script Rebol à la ligne de commande Windows :

```
C> rebol monscript.r
```

(rebol.exe doit être dans le PATH)

Exécuter un script dans la console Rebol (Tout OS)

Au prompt de la console, taper l'instruction **DO** suivie du caractère % et du chemin du script.

Exemple :

```
do %/home/appli/monscript.r
```

Exécuter un script Rebol à travers le Web (Tout OS)

Une des possibilités intéressantes de Rebol (il y en a beaucoup d'autres), c'est de pouvoir exécuter des scripts situés sur un serveur distant.

Au prompt de la console, taper l'instruction **DO** suivie de l'URL du script.

Exemple :

```
do http://www.serveurxyz.com/monscript.r
```

Les options de lancement

L'évaluateur Rebol supporte plusieurs options de lancement. Ces options sont visualisables par l'instruction **USAGE** dans la console ou par **rebol -?** à la ligne de commande.

Voici ces options :

```
REBOL <options> <script> <arguments>
```

All fields are optional. Supported options are:

```
--cgi (-c)           Check for CGI input
--do expr            Evaluate expression
--link url          Connect to Link
--help (-?)        Display this usage information
--nowindow (-w)     Do not open a window
--noinstall (-i)   Do not install (View)
--quiet (-q)       Don't print banners
--reinstall (+i)   Force an install (View)
--script file      Explicitly specify script
--secure level     Set security level:
                  (allow ask throw quit)
--trace (-t)       Enable trace mode
--uninstall (-u)   Uninstall REBOL (View)
```

Other command line options:

```
+q                 Force not quiet (View)
-s                No security
+s                Full security
-- args           Provide args without script
```

Examples:

```
REBOL script.r
REBOL script.r 10:30 test@domain.dom
REBOL script.r --do "verbose: true"
REBOL --cgi -s
REBOL --cgi --secure throw --script cgi.r "debug: true"
REBOL --secure none
```

Les arguments passés à un script Rebol

Tout script Rebol peut recevoir des arguments passés en paramètre lors de son lancement.
On peut récupérer ces arguments avec la propriété args de l'objet **system/script**.
Le retour est une chaîne de caractères contenant les arguments.

Exemple :

```
rebol [  
  Sujet: "Test ..."  
  Auteur: "RebolZone"  
  Version: 1.0.0  
]  
arguments: system/script/args  
print type? arguments ; arguments est de type string  
print arg  
print "Ok"  
input
```

On peut également vérifier la présence des arguments, leur validité ...

Exemple :

```
rebol [  
  Sujet: "Test ..."  
  Auteur: "RebolZone"  
  Version: 1.0.0  
]  
arguments: system/script/args  
; Pas d'arguments !!!!  
if arguments = none  
[  
  print "Usage : monscript.r arg1 arg2"  
  print "Arguments obligatoires"  
  quit  
]  
print type? arguments  
probe arguments  
  
bArguments: to-block arguments ; conversion de la chaîne des arguments en bloc  
foreach arg bArguments ; Parcours des éléments du bloc  
[  
  print arg  
]  
  
print "Fin de script"  
input
```