

La gestion des paramètres

1. Rappels

Les paramètres spéciaux

Ce sont en fait là aussi des variables réservées qui permettent pour certaines d'effectuer des traitements sur les paramètres eux-mêmes.

Ces paramètres sont les suivants :

\$0	Contient le nom du script tel qu'il a été invoqué
\$*	L'ensemble des paramètres sous la forme d'un seul argument
@	L'ensemble des arguments, un argument par paramètre
\$#	Le nombre de paramètres passés au script
\$?	Le code retour de la dernière commande
\$\$	Le PID du shell qui exécute le script
#!	Le PID du dernier processus lancé en arrière-plan

2. La commande set

Il est possible d'affecter directement des paramètres au shell grâce à la commande "set".

Une simple commande tel que :

```
set param1 param2 param3
```

initialisera automatiquement les paramètres positionnels "\$1,\$2,\$3" avec les valeurs "param1,param2,param3", effaçant de ce fait les anciennes valeurs si toutefois elles existaient. Les paramètres spéciaux "#,* et @" sont automatiquement mis à jours en conséquence.

Cette fonctionnalité peut s'avérer utile dans le traitement de fichiers ligne par ligne afin d'isoler chaque mot (champ), et d'en formater la sortie.

```
$ IFS=" "; set $(grep $USER /etc/passwd)
$ echo -e "Login :\t$1\nNom :\t$5\nID :\t$3\nGroup :\t$4\nShell :\t$7"
```

3. La commande shift

La commande interne "shift" permet quant à elle de décaler les paramètres.

La valeur du 1er paramètre (\$1) est remplacée par la valeur du 2nd paramètre (\$2), celle du 2nd paramètre (\$2) par celle du 3ème paramètre (\$3), etc...

On peut indiquer en argument (shift n) le nombre de pas (position) dont il faut décaler les paramètres.

```
#!/bin/bash

while [ $# -ne 0 ]
do
    echo $1
    shift
done
```

4. La commande getopt

La commande getopt permet de parser les paramètres comme étant des options.

getopts « optstring » name

optstring contient les lettres options à reconnaître. Si une lettre est suivi par « : », alors l'option nécessite un argument qui doit être séparé par un espace.

A chaque fois que getopt est invoquée, \$name contient le nom de l'option.

Si une option requiert un argument, alors celui-ci est placé dans OPTARG.

```
#!/bin/bash

OPTIND=1

outputfile=""
verbose=0

function show_help
{
    echo "Usage : "
    echo "    ${0#*/} -h : print this help"
    echo "    ${0#*/} -v : verbose "
    echo "    ${0#*/} -f : output file"
}

while getopts "h?vf:" opt; do
    case $opt in
        h|\?) show_help
                exit 0 ;;
        v) verbose=1 ;;
        f) output_file=$OPTARG ;;
        esac
done

echo "verbose=$verbose, output_file=$output_file, leftovers: $@"
```