

Programmation Shell

1-Pourquoi est-il important de commencer votre script en indiquant la commande du shell ?

Car le shell courant lit cette première ligne et lance le shell correspondant pour exécuter le script. Dans le cas contraire, il considère que le langage est le sien, même si le script contient du PHP ou du Perl.

2-Un script se voit transmettre 10 paramètres. Comment récupérer le dixième le plus rapidement possible ?

- A - par \$10.
- B - en écrivant neuf « shift » et en récupérant \$1.
- C - un shift puis \$9.
- D - par : A=10 puis \${A}.

C. Un shift décale tous les arguments d'une unité, le dixième paramètre devient le neuvième, donc \$9.

3- Comment depuis votre shell pouvez-vous récupérer la valeur retournée par « exit » d'un script quelconque ?

Il s'agit du code retour de la commande, récupéré par \$?.

4- Comment récupérer, connaissant un login, son UID dans une variable de même nom ?

Par une substitution de commande : UID=\$(grep ^login /etc/passwd | cut -d: -f3)

5- Quelle ligne écrire dans un script pour sortir avec un message d'erreur et un code de retour 1 si le nombre de paramètres passés est égal à 0 ?

- A - test \$# != 0 || echo Erreur ; exit 1
- B - [\$# -eq 0] && { echo Erreur ; exit 1 ; }
- C - test \$0 -ne 0 && (echo Erreur; exit 1)
- D - [[\$# -eq 0]] && echo Erreur && exit 1

B et D sont corrects.

6-Si le fichier fic existe et est lisible, listez son contenu. Sinon, créez-le.

[[-e fic -a -r fic]] && cat fic || touch fic. Ou encore [-r fic] car de toute façon le test retourne une erreur si le fichier est absent.

7-À l'aide d'un if, s'il n'y a aucun paramètre transmis au script, mettez-en deux prépositionnés : -l et -r.

```
if [ $# -eq 0 ] ; then set '-l' '-r' ; fi
```

8- Un programme attend trois valeurs au clavier, les unes après les autres, séparées par des espaces, sur la même ligne. Quelle doit être la commande à utiliser ?

```
read a b c
```

9- Comment faire une boucle pour lister tous les paramètres ? Choisissez deux réponses.

- A - for param ; do echo \$param ; done
- B - while [\$# -ne 0]; do echo \$1; shift ; done
- C - until [\$# -ne 0]; do shift; echo \$1; done
- D - a=1; while [\$a -ne \$#]; do echo \${a}; a=\$((a+1)); done

A et B.

10- Comment sortir d'une boucle sans fin ?

Avec un break.

11- Comment programmer une fonction abs qui retourne la valeur absolue d'un nombre ?

```
abs() { v=$1 ; [ $v -lt 0 ] && v=$((0-v)) ; echo $v ; }
```