

Aide mémoire Test en BASH

variables shell

Variables prédéfinies:

- **\$#** contient le nombre de paramètres passés;
- **\$0** contient le nom du script lui-même;
- **\$1 à \$9** contient les neufs premiers paramètres;
s'il y en a plus de neuf, il faut utiliser la commande shift qui décale sur la droite les arguments :
\$1 reçoit \$2, ..., \$9 reçoit le dixième argument;
- **\$*** Contient les n paramètres passés
- **\$?** Valeur de status de la dernière commande utilisée
- **\$\$** Valeur du numéro de processus courant

Entrées / sorties

Sorties écran

`echo` permet d'afficher une chaîne de caractères ou le contenu de variables sur la sortie standard.

```
unNom="bidochon"  
echo "La variable unNom vaut $unNom"
```

Entrées / sorties

Entrées clavier

`read` permet de lire une ligne de caractères à partir de l'entrée standard et place les différents mots dans les variables transmises en argument.

```
echo "Saisissez votre login et votre mot de passe: "  
read leLogin mdp  
echo $leLogin $mdp
```

Tests

La commande **test** permet de tester l'existence et le status d'un fichier ou d'une variable.

On peut également utiliser **[]**.

Test de fichier

[-r fichier]	Vrai si le fichier existe et est lisible
test -r fichier	Vrai si le fichier existe et est lisible
test -w fichier	idem en écriture
test -x fichier	idem en exécution
test -s fichier	Vrai si la taille du fichier est supérieure à 0
test -d fichier	Vrai si le fichier est un répertoire
test -f fichier	Vrai si le fichier est un fichier ordinaire

Tests

Test de chaînes de caractères

`test -z s1`

Vrai si la chaîne s1 est de taille nulle

`test -n s1`

Vrai si la chaîne s1 est de taille non nulle

`test s1 = s2`

Vrai si s1 est identique à s2

`test s1 != s2`

Vrai si s1 est différent de s2

`test s1`

Vrai si s1 n'est pas la chaîne nulle

ATTENTION à bien mettre les espaces autour de = ou !=

Tests

Test de valeurs

e1 -eq e2	Vrai si $e1 = e2$
e1 -ne e2	Vrai si $e1 \neq e2$
e1 -gt e2	Vrai si $e1 > e2$
e1 -ge e2	Vrai si $e1 \geq e2$
e1 -lt e2	Vrai si $e1 < e2$
e1 -le e2	Vrai si $e1 \leq e2$

Tests

Ces tests peuvent être combinés

- ! négation logique
- a et logique (équivalent au && en langage C)
- o ou logique (équivalent au || en langage C)

Ex:

test -r fic1 -a -s fic2

→ vrai si fic1 est un fichier accessible en lecture et si fic2 n'est pas un fichier vide.

Tests

if-then-else

```
if test de condition
then
    liste de commandes
else
    liste de commandes
fi
```

Tests

Exemple 1

```
#!/bin/sh  
if test ! -s $HOME/listeDeMesAmies  
then  
    echo "tu ne possedes pas d amies"  
fi
```

Tests

if-then-else

Exemple 2 (nom du script : equal.sh)

```
#!/bin/sh
if test $1 -eq $2
then
    echo $1 et $2 sont identiques
else
    echo $1 et $2 sont differents
fi
```

./equal 1 2 → 1 et 2 sont differents

./equal 2 2 → 2 et 2 sont identiques

Tests

Le branchement multiple case

```
case mot in
  cas1) liste de commandes ;;
  cas2) liste de commandes ;;
  ...
esac
```

Exemple:

```
case $reponse in
  "oui") echo "votre reponse est oui" ;;
  "non") echo "votre reponse est non" ;;
  *) echo "c'est quoi cette reponse?" ;;
esac
```