

ACTIONS AVANCÉES SUR LES VARIABLES

Supprimer une variable

- La commande **unset** permet de supprimer la variable à condition que celle-ci ne soit pas protégée par une instruction **readonly**.
- Attention, il ne s'agit pas de supprimer la valeur associée à la variable, mais bien de supprimer la variable elle-même (en libérant l'espace mémoire lui étant affecté).

```
debian:/home/jordan# VARIABLE1="Jordan"  
debian:/home/jordan# echo $VARIABLE1  
Jordan
```

```
debian:/home/jordan# unset VARIABLE1  
debian:/home/jordan# echo $VARIABLE1
```

Utilisation de l'export

- Pour qu'un processus "père" puisse faire connaître une variable à son processus "fils", il doit l'exporter avec la commande **export**.
- Script : `script_export.sh`

```
#!/bin/bash  
echo "La variable prenom a pour valeur : $PRENOM"
```

```
debian:/home/jordan# PRENOM="Jordan"  
debian:/home/jordan# ./script_export.sh  
La variable prenom a pour valeur
```

```
debian:/home/jordan# export PRENOM  
debian:/home/jordan# ./script_export.sh  
La variable prenom a pour valeur Jordan
```

Préciser le type de variable

- En Scripting Shell, comme dans la plupart des autres langages, il est possible de spécifier le type de variable que l'on déclare grâce à la commande **typeset** ou **declare**
- `typeset -a NOM_DE_LA_VARIABLE` : tableau
- `typeset -i NOM_DE_LA_VARIABLE` : entier
- `typeset -r NOM_DE_LA_VARIABLE` : readonly
- `typeset -x NOM_DE_LA_VARIABLE` : export

Les variables prédéfinies

- **\$HOME** : Répertoire personnel de l'utilisateur
- **\$PWD** : Répertoire courant
- **\$LOGNAME** : Nom de l'utilisateur
- **\$PS1** : Prompt principal
- **\$REPLY** : Chaîne saisie par l'utilisateur si la commande **read** n'a pas été utilisée avec un argument.
- **\$RANDOM** : nombre aléatoire compris entre 0 et 32767
- **\$\$** : numéro du processus actuel

Utilisation de la commande set et des IFS

- Lorsque des instructions avec la commande **set** sont effectuées, le Shell parvient à isoler et stocker les différentes valeurs dans les variables de positionnement.
- C'est la variable **IFS** qui contient les caractères devant être utilisés pour la séparation des différentes valeurs (par défaut il s'agit de l'espace)

```
#!/bin/bash
CHAINE="Bonjour:je:suis:Jordan"
IFS=:
Set $CHAINE
echo $2
```

```
./script.sh
je
```

Arithmétique sur les variables

- La commande **expr** réalise les opérations mathématiques demandées :
 - + : addition
 - - : soustraction
 - * : multiplication
 - / : division euclidienne
 - % : modulo
- Syntaxe :

```
i=5
j=`expr $i + 4`

echo $j
9
```

Découper des lignes avec cut

- La commande **cut** permet de sélectionner un champ ou un caractère précis dans une ligne. Elle prend différentes options :
 - -f : découpe la ligne suivant le caractère tabulation
 - -d : spécifie un autre caractère de séparation
 - -s : ignore les lignes ne contenant pas de caractère de séparation

Affichage du début de chaque ligne jusqu'à trouver un ":"
`cat /etc/passwd | cut -d: -f1`

Affichage des deux premiers champs de chaque ligne en utilisant : comme délimiteur
`cat /etc/passwd | cut -d: -f1-2`