



CS230

TP 2 Programmation Shell

Ecriture de scripts simples

A rendre dans une semaine à partir d'aujourd'hui :

- Une archive en format compressé (zip, tar.gz, etc), à déposer sur chamillo (partie "Travaux").
- L'archive doit comportée :
 - un compte rendu explicatif en format PDF contenant les noms des binomes, les réponses aux différentes questions, les éventuelles remarques et détails des problèmes rencontrés, les scripts shell et les résultats d'exécutions commentés.
 - les scripts développés

Table des matières

I	Écriture et exécution de scripts	2
II	Les variables	2
III	Manipulation de chaînes de caractères	3
IV	Expressions arithmétiques et exécution conditionnelle	3

Première partie . Écriture et exécution de scripts

Exercice 1 : Ecrire un programme shell `ex1.sh` qui affiche le nom de connexion de l'utilisateur et le chemin absolu de son répertoire courant de la manière suivante :

```
$ ./ex1.sh
Ceci est mon deuxieme TP de shell CS230.
Mon nom de connexion est : debbabi
Mon repertoire courant est : /home/debbabi/tp2
```

Remarque : vous devez accorder le droit d'exécution à votre script.

Deuxième partie . Les variables

Exercice 2 :

- Dans un shell ouvert, créez trois variables `VAR1`, `VAR2` et `VAR3`; initialisez ces variables avec les valeurs "Quatorze", "14" et "Happy Valentine Day".
- Afficher les valeurs des trois variables.
- Ces variables sont-elles locales au shell courant ou globales?
- Supprimez la variable `VAR3`.
- Pouvez-vous voir ces variables depuis un autre terminal? comment faire pour que ces variables soient visibles à tous les utilisateurs?
- Ajoutez le script développé dans l'exercice 1 (`ex1.sh`) dans le `PATH` afin que vous puissiez l'appeler depuis n'importe quel dossier dans le système et depuis n'importe quelle terminal.
- Est ce que ce changement est pérenne? (c-à-d auriez-vous toujours le chemin de ce script dans le `PATH` à chaque redémarrage de l'ordinateur?). Qu'est ce qu'il faut faire dans le cas contraire?

Exercice 3 : Ecrire un programme shell `ex3.sh` qui affiche le message "Entrez un mot : ", lit le mot saisi par l'utilisateur puis affiche ce mot deux fois sur la même ligne.

```
$ ./ex3.sh
Entrez un mot : toto
toto toto
```

Exercice 4 : Ecrire un programme shell `ex4.sh` qui demande à l'utilisateur de saisir une suite de mots constituée d'au moins trois mots et qui affiche sur la même ligne le premier et le troisième mot saisis.

```
$ ./ex4.sh
Entrez une suite de mots : un petit coucou de Rodez
un coucou
```

Exercice 5 : Écrire un script `ex5.sh` qui affiche le nombre de ses arguments et les valeurs des deux premiers arguments.

```
$ ./ex5.sh un deux trois quatre cinq
Il y a 5 arguments.
1) un
2) deux
```

Exercice 6 : En utilisant la variable d'environnement `IFS`, la commande interne `set` et la commande externe `grep`; écrire un script `ex6.sh` qui permet d'afficher les informations qui concerne un utilisateur se trouvant sur le fichier système `/etc/passwd`.

```
$ ./ex6.sh debbabi
Login : debbabi
Nom   : Debbabi
ID    : 1000
Group : 1000
Home  : /home/debbabi
Shell : /bin/bash
```

Troisième partie . Manipulation de chaînes de caractères

Exercice 7 : Soit le texte suivant :

```
Albert Einstein a dit : "L'imagination est plus importante que le savoir."
```

Ecrire un script `ex7.sh` qui permet de lire ce texte et de le ré-afficher comme suit (format JSON¹) :

```
{
  Personnage : "Albert Einstein",
  Citation   : "L'imagination est plus importante que le savoir."
}
```

Quatrième partie . Expressions arithmétiques et exécution conditionnelle

Exercice 8 : Soit le script `ex8.sh` suivant :

```
#!/bin/bash
var1=7
# affectation simple
var1=$((var1+10))
echo $var1
```

- Expliquez pourquoi ce script affiche “7+10” sur l’écran au lieu de “17”.
- Corrigez ce script pour que le résultat de l’évaluation de l’expression arithmétique soit correct (proposez trois solutions différentes).

Exercice 9 : Ecrire un script `ex9.sh` qui prend deux entiers positives en arguments et affiche le max et le min. Vérifiez que le nombre de paramètres est correct. On traite tous les cas d’erreur et les nombres négatifs.

Exercice 10 : Ecrire un script `ex10.sh` qui donne la valeur d’un nombre (donné en argument) selon les cas suivants :

- a) nombre = nombre si nombre est pair,
 - b) nombre = 3 * nombre + 1 si nombre est impair.
- Traitez les cas d’erreurs y compris un nombre de valeur supérieure à 9999999.

Exercice 11 : Écrire un script `ex11.sh` qui calcule la somme de deux nombres (réels) passés comme arguments. Les nombres réels utilisent un point ou une virgule comme séparateurs. Par exemple 5.6 ou 5,6

Exercice 12 : Écrire un script `ex12.sh` qui prend en argument le nom d’un fichier et affiche sur la sortie standard les lignes de ce fichier comprises entre deux bornes (données comme argument aussi).

1. JSON (*JavaScript Object Notation*) est un format de données textuelles, générique, dérivé de la notation des objets du langage ECMAScript (JavaScript)

```
$ ./ex12.sh --min 7 --max 10 --fichier monFichier.txt  
ligne7  
ligne8  
ligne9  
ligne10
```

Indications :

- L'ordre des arguments n'est pas obligatoire (on peut lancer ce script comme suit par exemple : `./ex12.sh --fichier monFichier.txt --max 10 --min 7`)
- la commande `head -nombre` fournit les **nombre** premières lignes d'un fichier (exemple : `head -7 ref_fichier` fournit les 7 premières lignes du fichier `ref_fichier`)
- la commande `tail +nombre` fournit les dernières lignes d'un fichier en commençant par la ligne de numéro **nombre**. (exemple : `tail +7 ref_fichier` fournit les dernières lignes du fichier `ref_fichier` en commençant par la ligne 7)
- il faut traiter tous les erreurs liés aux paramètres.