AGUILAR_Flavien

i1-tp-script_1 : Script

1 Affichage:

1. 1 Utiliser la commande echo dans un script pour afficher «bonjour»

1 - Pour afficher bonjour « retour_a_la_ligne » tout le monde, nous allons utiliser la commande :

```
test@debian:~$ echo -e "Bonjour \ntout le monde"
Bonjour
tout le monde
```

2 - Pour afficher le même texte qu'à la question 1 le tout en 3 commandes et à l'aide d'un script, nous allons utiliser les commandes suivantes :

```
test@debian:~$ vim monscript.sh
test@debian:~$ chmod +x monscript.sh
test@debian:~$ ./monscript.sh
Bonjour
tout le monde
```

Le script « monscript.sh » contient les lignes suivantes :

```
#!/bin/bash
echo -e "Bonjour \n tout le monde"
```

3 - Pour afficher « Bonjour tout le monde » à l'aide d'un script et de deux commandes, nous allons utiliser les commandes :

```
test@debian:~$ vim monscript.sh
test@debian:~$ ./monscript.sh
Bonjour tout le monde
```

Le script « monscript.sh » contiendra alors les lignes :

```
#!/bin/bash
echo "Bonjour tout le monde"
```

2 Variables :

1 - Grace à la commande set, j'ai pu définir les deux variables correspondant au nom de l'utilisateur et au nom de la machine :

```
USER=test    -> Nom d utilisateur
HOSTNAME=debian
```

2 - Le script permettant d'afficher « je suis l'utilisateur x sur la machine y » est :

```
#!/bin/bash
echo "Je suis l'utilisateur $USER de la machine $HOSTNAME"
```

Nous pouvons vérifier son bon fonctionnement :

```
test@debian:~$ ./monscript.sh
Je suis l'utilisateur test de la machine debian
```

3 - Pour définir des variables et afficher « l'utilisateur x s'appelle y z », nous allons utiliser les commandes suivantes :

```
test@debian:~$ export nom=AGUILAR
test@debian:~$ export prenom=Flavien
test@debian:~$ echo "L'utilisateur $USER s'appelle $nom $prenom"
L'utilisateur test s'appelle AGUILAR Flavien
```

4 - Pour affecter la commande ls à une variable, nous allons utiliser les commades suivantes :

```
test@debian:~$ export list=$(ls)
test@debian:~$ echo $list
monscript.sh
```

5 - Pour demander à l'utilisateur de saisir son nom et son prénom afin de l'afficher, nous allons utiliser le script suivant :

```
#!/bin/bash

read -p "Veuillez saisir votre nom : " nom
read -p "Veuillez saisir votre prénom : "prenom

echo "Vous vous appellez $nom $prenom"
```

```
test@debian:~$ ./monscript.sh
Veuillez saisir votre nom : aguilar
Veuillez saisir votre nom : flavien
Vous vous appellez aguilar flavien
```

3 Analyse de la ligne de commande

1 - Pour afficher l'ensemble de la ligne de commande d'un script, nous allons utiliser la variable $* :

```
#!/bin/bash
echo $*
```

```
test@debian:~$ ./monscript.sh 1 2 3 1 2 1 3
1 2 3 1 2 1 3
```

2 - Pour afficher seulement la commande, nous allons utiliser la variable $0 :

```
#!/bin/bash
echo $0
```

```
test@debian:~$ ./monscript.sh 1 2 3 1 2 1 3
./monscript.sh
```

3 - Pour créer un groupe dont le nom sera donné en argument et dont nous allons affecter tous les droits à « others », nous allons utiliser le script suivant :

```
#!/bin/bash
mkdir $1
```

```
3 chmod 007 $1
```

4 - La variable permettant d'afficher le statut de l'exécution d'une commande est : $? , nous allons modifier notre script afin d'afficher ce retour entre chaque commande :

```
1 #!/bin/bash
2 mkdir $1
3 echo $?
4 chmod 007 $1
5 echo $?
```

```
1 test@debian:~$ ./monscript.sh test
2 0
3 0
```

5 - Pour créer un répertoire dans un répertoire ou nous avons les droits, nous allons utiliser le script suviant :

```
1 #!/bin/bash
2 mkdir ~/$1
```

6 - Pour créer un répertoire dans un répertoire ou nous n'avons pas les droits nous allons utiliser le script :

```
1 #!/bin/bash
2 mkdir /root/flavien
```

7 - Pour créer une arborescence à 2 niveaux, nous allons utiliser le script :

```
1 #!/bin/bash
2 mkdir niveau1
3 mkdir niveau1/niveau2/
```

8 - Lorsque nous voulons écrire dans un répertoire ou nous avons les droits, le retour de la variable $? sera 0, ce qui indique que la commande a bien pu s'exécuter sans erreur, en revanche,lorsque nous avons pas les droits sur le répertoire, la variable $? nous retourne un code d'erreur qui nous indique que la commande n'a pas pu s'exécuter normalement.

4 Utilisation du shebang

1 - Le shebang va permettre de communiquer au système que le fichier n'est pas un binaire mais un script. Il précise aussi avec quel interpréteur nous allons exécuter le script.

2 - Pour exéctuter un script à l'aide du shebang, nous allons devoir disposer des droits d'exécutions sur le fichier.

3 - La ligne à rajouter tout en hautd d'un script pour le rendre executable est :

```
1 #!/bin/bash
```

4 - Pour inclure le dossier bin situé dans notre répertoire personnel à la variable $PATH, nous allons utiliser la commande :

```
1 test@debian:/$ export PATH=$PATH:/home/test/bin/
```

5 - Cette méthode va nous permettre d'exécuter notre script depuis n'importe quel répertoire.