

TP2

Pour effectuer le TP, créer un répertoire (`mkdir`) et se placer dedans.

1 Environnement

Question 1 : Écrire un script shell `hello.sh` affichant la chaîne « hello ! ».

Question 2 : Essayer de lancer ce script en tapant la commande `hello.sh`. Que constatez-vous ?

Ajouter les droits d'exécution à votre script à l'aide de la commande `chmod u+x hello.sh`.

Question 3 : Essayer de nouveau la commande de la question précédente. Observez-vous des changements ?

Question 4 : Utiliser maintenant la variable `USER` pour personnaliser le message.

Les fichiers `/etc/profile` et `~/.bash_profile` contiennent les configurations des variables (entre autres choses) prises en compte au login. Dans le cas d'un shell lancé directement par l'utilisateur (dit *interactif*), le fichier `~/.bashrc` est également pris en compte.

Question 5 : Cherchez toutes les modifications de la variable `PATH`.

2 Arguments

Question 6 : Écrivez un script shell basique `prod.sh` qui effectue la multiplication des deux entiers passés en argument. Un exemple d'exécution est donné ci-dessous.

```
$ ./prod 4 6
24
$
```

Question 7 : Enrichissez votre script pour afficher (sur la sortie d'erreur) un message indiquant le fonctionnement si le nombre d'arguments est incorrect. Faites également en sorte que la valeur de retour indique une erreur.

Question 8 : Écrivez également la fonction précédente en python.

Question 9 : Écrivez en shell un script `table.sh` qui affiche la table de multiplication d'un nombre passé en argument. Le résultat pourra s'afficher de la façon suivante :

```
$ table.sh 4
4*0=0
4*1=4
4*2=8
4*3=12
<...>
4*10=40
$
```

3 En shell, je programme

Question 10 : Écrivez un script `rev.sh` qui transforme chaque ligne lue sur l'entrée standard en prenant les deux premiers termes et en les mettant à la fin de la ligne précédés d'une virgule. Par exemple, on aura l'exécution suivante :

```
$ rev.sh
je suis content
content, je suis
je crois que le script marche
que le script marche, je crois
...
```

Question 11 : Modifiez votre script pour que celui-ci s'arrête si le premier mot d'une ligne est **quit**.

Question 12 : Enchaîner alors votre script deux fois consécutives à l'aide d'un tube. Que se passe-t-il sur la suite d'entrées suivante :

- le résultat est assez incompréhensible
- la commande `quit` sert à fermer le programme
- il faut toujours réfléchir

Question 13 : Expliquez le comportement observé précédemment.