

Exercice 1 : Saisissons l'utilisateur

Q 1. Écrire un script, nommé `oui-non`, qui demande à l'utilisateur de saisir `oui` ou `non` jusqu'à ce que celui-ci ait effectué une saisie correcte.

Q 2. Écrire un script, nommé `voir-variable`, qui, indéfiniment, demande, le nom d'une variable à l'utilisateur et affiche sa valeur. Le script s'arrête uniquement si l'utilisateur entre le mot `FINI`.

Q 3. Écrire un script, nommé `compter-arguments`, qui affiche un à un ses arguments sous la forme :

```
$ compter-arguments toto bonjour titi
L'argument 1 est toto
L'argument 2 est bonjour
L'argument 3 est titi
$
```

Exercice 2 : Dessinons un arbre

Q 1. Écrire un script, nommé `arbre`, qui affiche l'arborescence dont la racine est passée en paramètre.

Q 2. Modifier la commande pour ajouter une indentation suivant la profondeur du fichier par rapport à la racine.

Q 3. En utilisant le contenu du répertoire `/proc` écrivez un script, nommé `psarbre`, qui affiche l'arborescence des processus (leur PID et leur ligne de commande) en cours d'exécution sur le système.

Exercice 3 : Un peu de recyclage avec la poubelle

Modifier la commande `jeter-fichiers` que vous avez déjà écrite afin d'ajouter quelques améliorations :

- ne pas écraser les fichiers déjà dans la poubelle, si on veut jeter un fichier de nom `fichier` et qu'un fichier de même nom se trouve déjà dans la poubelle, il faut renommer le nouveau fichier à jeter en `fichier#1`, ou `fichier#2` si le fichier `fichier#1` est déjà présent dans la poubelle et ainsi de suite ;
- transformer l'option `-r` en `-v` (v pour vider) ;
- ajouter une option `-s` (s pour sortir), `jeter-fichiers -s fichier chemin` sort le fichier `fichier` de la poubelle et le place à l'emplacement indiqué par chemin ;
- ajouter une option `-r` (r pour restaure), `jeter-fichiers -r fichier` restaure le fichier `fichier` contenu dans la poubelle à son *emplacement* initial avec son *nom* initial. Par exemple :

```
bruno@watney:~/essai/tp$ jeter-fichiers -l
toto.txt
bruno@watney:~/essai/tp$ jeter-fichiers toto.txt
bruno@watney:~/essai/tp$ jeter-fichiers -l
toto.txt toto.txt#1
bruno@watney:~/essai/tp$ cd ../test
bruno@watney:~/essai/test$ jeter-fichiers -r toto.txt#1
bruno@watney:~/essai/test$ cd ../tp
bruno@watney:~/essai/test$ ls
toto.txt
```

Si lors de la restauration, il existe déjà un fichier de même nom à l'emplacement visé, demandez confirmation avant de l'écraser.