UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Outils Info. - TD : Feuille 1

```
/home
+-- bob
    +-- outils-info
        +-- corbeille
        +-- tp0
        |   +-- compte-rendu.txt
        |   +-- cr.txt
        |   +-- ex1
        |   |   +-- info.txt
        |   |   +-- src
        |   +-- ex2
        |   |   +-- cr.txt
        |   |   +-- info2.txt
        |   |   +-- info.txt
        |   |   +-- src
        |   |       +-- calcul.py
        |   +-- ex3
        |       +-- src
        |           +-- erreur.py
        |           +-- simple.py
        +-- tp1
        |   +-- cr.txt
        |   +-- ex1
        |   |   +-- prog.bash
        |   +-- ex2
        |   |   +-- test.bash
        |   +-- ex3
        |       +-- commentaires.txt
        |       +-- solution.bash
        +-- tp2
            +-- ex1
            +-- ex2
            +-- ex3
```

Exercice 1 – commandes simples, chemins

Pour cet exercice, les questions sont indépendantes et on suppose : 1) l'arborescence donnée plus haut, et 2) que `pwd` renvoie `/home/bob/outils-info/tp0`, 3) que chaque « façon » ne doit contenir *qu'une commande*.

Q1) Donnez une façon d'afficher le contenu du fichier `compte-rendu.txt`.

Q2) Donnez 2 façons d'afficher le contenu du fichier `info2.txt`.

Q3) Donnez 3 façons de se déplacer dans le *dossier maison* (dossier `bob`).

Q4) Donnez 2 façons d'avoir la liste des fichiers contenus dans `ex3` de `tp1`

Q5) Qu'affiche la commande `pwd` ?

Q6) Que fait `echo salut bob` ?

Q7) Que fait `echo salut    bob` ?

Q8) Que fait `echo *` ?

Q9) Que fait `echo ../*/ex1/*.txt` ?

Q10) Que ferait `rm -r ~` ? (NB : ne pas tester cette commande)

Exercice 2 – successions de commandes, chemins

Pour cet exercice, les questions sont indépendantes et on suppose : 1) l'arborescence donnée plus haut, et 2) que `pwd` renvoie `/home/bob`, 3) que chaque « façon » peut contenir *plusieurs commandes.*

Q11) Donnez 2 façons de renommer le fichier `erreur.py` en `ok.py`.

Q12) Donnez 4 façons de déplacer le fichier `erreur.py` dans le dossier `corbeille`. Par exemple, 1) sans se déplacer, 2) en se déplaçant dans `outils-info`, 3) en se déplaçant dans `corbeille`, et 4) en se déplaçant dans `src`.

Q13) Donnez 2 façons de faire une copie de `erreur.py` appelée `ancien-erreur.py`.

Exercice 3 – script avec paramètres

Q14) Écrire un script `sauvegarde.bash` qui quand on l'exécute avec `bash sauvergarde.bash tp0` crée une copie de `tp0` (son paramètre) appelée `sauve-tp0`.

Q15) Écrire un script `sauve-tous.bash` de façon à ce que quand on l'exécute avec `bash ../sauve-tous.bash py`, il crée un dossier `sauve-py` et copie tous les fichiers ayant l'extension `py` dans ce dossier.

Q16) Où le script doit il être sauvé (donnez son chemin complet) pour que la commande bash de la question précédente fonctionne ?

Q17) Écrire un script `sauve-tous-depuis.bash` de façon à ce que quand on l'exécute avec `bash sauve-tous-depuis.bash txt tp0/ex2` depuis le dossier `outils-info`, il crée un dossier `sauve-txt` (dans `tp0/ex2`), y copie tous les fichiers ayant l'extension `txt` et y crée un fichier `DATE` (vide, sa date de création répresentera la date de la sauvegarde).

Q18) Selon vous que se passe-t-il si l'on appelle 2 fois de suite `sauve-tous-depuis.bash` avec exactement les mêmes paramètres ?

(optionnel) Exercice 4 – conversion script Bash vers programme Python

Q19) Écrire l'équivalent Python 3 du script bash ci-dessous

```
a=toto
b=tutu
echo "${a} et ${b}"
```

Q20) Écrire l'équivalent Python 3 du script bash ci-dessous

```
echo "#####"
echo "Bonjour ${1}"
echo "#####"
echo "Nous sommes le ${2}"
```

Q21) Écrire l'équivalent Python 3 du script bash ci-dessous

```
a=2
b=3
c=$1
echo ${a}+${b}-${c}+${2}
```