

# Correction de la partie de Pierre Pompidor d'Utilisation des Systèmes informatiques HAI103I

Quel est le métacaractère de Bash qui permet la transmission à une commande le résultat d'une autre commande?

Donnez le nom de la commande qui filtre les lignes d'un fichier par rapport à un motif donné.

```
grep  
ou egrep
```

Donnez la commande qui recherche à partir du dossier courant tous les fichiers dont les noms se terminent par ".sh"

```
find . -name "*.sh"  
ou  
ls -R | grep "\.sh$"
```

Donnez le symbole qui dans une expression régulière désigne un caractère de valeur quelconque.

```
.
```

Donnez l'expression régulière qui vérifie qu'une chaîne de caractères ne contient que des chiffres.

```
^[0-9]+$  
ou ^[01234789]+$  
ou ^\d+$  
ou ^[:,digit:]+$
```

(le + exige d'utiliser egrep, il peut être remplacé par \* mais cela serait moins précis car les lignes vides seraient acceptées!  
Par ailleurs toutes les expressions régulières précédentes peuvent être entourées par des guillemets)

Par quel mot désigne-t-on le typage des langages de scripts Bash et Python?

```
dynamique (j'ai aussi accepté "faible")
```

Par quel mot, désigne-t-on les programmes qui exécutent les intructions des scripts Bash et Python?

```
interpréteur
```

En Bash, quel est le mot réservé qui termine un bloc d'instructions créé par une alternative (un test)?

```
fi
```

En Bash, comment est désignée la liste des paramètres (deux solutions sont possibles)?  ou

Donnez les instructions Bash qui créent une variable nommée nom initialisée avec la valeur "test", puis affichent sa valeur dans le terminal.

```
nom=test  
echo $nom
```

(j'ai bien sûr accepté nom="test")

Donnez l'instruction Bash qui crée une variable nommée liste qui contient les noms de tous les éléments du dossier dans lequel s'exécute le script.

```
liste=**  
ou liste=$(ls)  
ou liste='ls'
```

Donnez l'instruction Bash qui incrémente la variable i (c'est à dire lui rajoute 1).

```
i=$((i+1))  
ou let i=i+1
```

Donnez le code Bash qui teste si un des paramètres du script commence par un tiret (et correspondrait donc à une chaîne d'options).

```
for p in $@  
do  
  if [[ $p == -* ]]  
  then  
    ...  
  fi  
done
```

Donnez le code Python qui crée un dictionnaire nommé dico vide.

```
dico = {}
```

En Python, soit une liste nommée `liste` contenant des entiers positifs (par exemple `[2, 23, 23, 7, 7, 7, 4, 9, 9, 7]`). Donnez le code qui crée une nouvelle liste nommée `nliste` similaire à la précédente mais qui dans le cas d'une séquence de valeurs identiques, ne conserve qu'une seule occurrence de celles-ci (pour notre exemple, le résultat serait `[2, 23, 7, 4, 9, 7]`). Une bonne idée serait d'utiliser une variable mémoire...

```
nliste = []
mem = -1 # ou 0, ou None, ou n'importe quoi sauf un entier positif
for n in liste :
    if n != mem :
        nliste.append(n)
        mem = n
```

Il y a bien sûr d'autres possibilités, mais les deux suivantes un peu plus courtes fonctionnent imparfaitement. Celle-ci car le dernier 7 dans l'exemple est omis (en fait tous les entiers qui réapparaissent en discontinuité) :

```
nliste = []
for n in liste :
    if n not in nliste :
        nliste.append(n)
```

celle-là car la liste `nliste` est initialement vide et donc la fameuse erreur "index out of range " survient :

```
nliste = []
for n in liste :
    if n != nliste[-1] :
        nliste.append(n)
```