

# TRAVAUX PRATIQUES - TP LA PROGRAMMATION AVEC LE SHELL/UNIX

## Modalités de livraison

Ce TP fait l'objet d'une notation. Vous travaillerez seul et vous devrez rendre un compte-rendu. Les TP sont à remettre **avant**

**le mercredi 22 novembre 2023** sur **EDUNAO**. Vous devez envoyer un fichier au format **PDF** qui devrait être nommé en suivant la convention suivante : **NOM-TP2.pdf**.

## Rappel

Un script shell est un fichier texte possédant l'extension **.sh**, comme **toto.sh**.

Un tel fichier peut être exécuté en commençant son code par une ligne précisant l'interpréteur de commande à utiliser :

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

Le fichier est ensuite rendu exécutable par la commande :

```
$ chmod +x toto.sh
```

et puis exécuté par :

```
$ ./toto.sh
```

## Exercice 1 : Sauvegarde automatique de fichiers

Le but de cet exercice consiste à définir un script permettant de sauvegarder tous les fichiers du répertoire courant dans un autre répertoire avec, en plus, une indication du jour où la sauvegarde est effectuée.

- Ecrire un script `bash` copiant tous les fichiers du répertoire courant dans le sous-répertoire `OLD` en ajoutant au nom des fichiers la date du jour au format `"#année-mois-jour"`. Il faudra, au préalable, s'assurer de l'existence du répertoire `OLD` et le créer s'il n'existe pas. On vérifiera alors dans ce cas que la création du répertoire s'est bien passée.
- Par exemple, si le répertoire courant comporte un fichier `fich.txt` et que le shell est lancé aujourd'hui, ce programme devra recopier le fichier `fich.txt` vers le fichier `OLD/fich#2020-11-04.txt`.

## Exercice 2 : Gestion d'une Corbeil de fichiers

- Écrire un script `jeter.sh` qui permet de manipuler une poubelle à fichier (un répertoire nommé `poubelle` situé à votre racine). La commande `$ ./jeter.sh fichier1 fichier2 ...` est utilisée pour déplacer les fichiers

- considérés vers la poubelle ;
- La commande accepte trois options :
    1. **-l** pour lister le contenu de la poubelle ;
    2. **-s fichier chemin** pour sortir le fichier de la poubelle et le placer à l'emplacement chemin ;
    3. **-v** pour vider la poubelle ;
  - Si la poubelle n'existe pas, elle est créée à l'appel de la commande.

## Exercice 3 : Compilation de fichiers Java

### Remarque

Pour réaliser cet exercice, vous devez installer **Java** sur votre machine. Vous devez également télécharger les codes **Java** à partir de [ce lien](#) pour tester votre script.

- Définir un script shell **javac\_test.sh** qui attend un argument et teste si cet argument est un fichier **Java** compilable sans erreurs.
  - Si l'argument est un fichier **Java** qui se compile sans aucune erreur, il affiche sur la sortie standard **file.java : compilé**, génère un fichier **file.class**, et termine avec le code de retour **0**.
  - Si l'argument est un fichier **Java** qui ne se compile pas, il affiche sur la sortie standard **file.java : non compilé** et termine avec le code de retour **1**.
  - Dans tous les autres cas, il affiche **Entree incorrecte** sur la sortie d'erreur standard et termine avec le code de retour **2**.

### Remarque

Pour compiler un fichier **Java**, il faut exécuter la commande :

```
$ javac fichier.java
```

- Définir un script shell **java\_compilation.sh** qui prend comme argument un répertoire **rep** qui contient des fichiers (**Java** ou autre), parcourt le contenu du répertoire, exécute **javac\_test.sh**, et sauvegarde les fichiers **.class** générés dans un répertoire **rep/bin**
- Définir un script shell **java\_execution.sh** qui prend comme argument un répertoire **rep**, teste si le répertoire **rep/bin** existe et exécute les fichiers **.class** générés par le script de la question précédente.

### Remarque

Pour exécuter un fichier **Java**, il faut exécuter la commande :

```
$ java fichier # sans l'extension
```

### Remarque

les fichiers **Java** exécutables contiennent une méthode avec la signature :

```
public static void main
```