



Faculté des Sciences de Technologies

## *Rapport TD8 systeme1*

**Nom** : PIERRE

**Prénom** : Yann Lelay

**Niveau** : L3-Sciences Informatiques

## Description du TD

-Comprendre la structure de base d'un script shell., Manipuler des variables dans un script. Utiliser des conditions ( if , case ). Implémenter des boucles ( for , while ) et Automatiser des tâches simples avec des scripts shell.

## 1. Créez un script qui affiche les multiples de 5 entre 5 et 100.

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

```
# Script pour afficher les multiples de 5 entre 5 et 100
```

```
for ((i=5; i<=100; i+=5)); do
```

```
    echo $i
```

```
done
```

## 2. Créez un script avec une boucle qui affiche un menu interactif à l'utilisateur

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

```
# Script pour afficher un menu interactif
```

```
while true; do
```

```
    echo "Menu Principal"
```

```
    echo "1. Afficher la date et l'heure"
```

```
    echo "2. Afficher l'espace disque disponible"
```

```
    echo "3. Afficher les utilisateurs connectés"
```

```
    echo "4. Quitter"
```

```
    read -p "Choisissez une option [1-4]: " choix
```

```
    case $choix in
```

```
        1)
```

```
            echo "Date et Heure actuelles : $(date)"
```

```
            ;;
```

```
        2)
```

```
            echo "Espace disque disponible :"
```

```
            df -h
```

```
            ;;
```

```
        3)
```

```
            echo "Utilisateurs connectés :"
```

```
            who
```

```
            ;;
```

```
        4)
```

```
            echo "Au revoir!"
```

```
            break
```

```
            ;;
```

```
        *)
```

```
            echo "Option invalide, veuillez choisir une option entre 1 et 4."
```

```
;;
esac
echo
done
```

### 3. Créez un script qui permet d'effectuer des opérations sur les variables

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

```
# Script pour effectuer des opérations sur les variables
```

```
while true; do
    echo "Menu des opérations"
    echo "1. Addition"
    echo "2. Soustraction"
    echo "3. Multiplication"
    echo "4. Division"
    echo "5. Quitter"
    read -p "Choisissez une option [1-5]: " choix

    if [ "$choix" -eq 5 ]; then
        echo "Au revoir!"
        break
    fi

    read -p "Entrez la première variable: " var1
    read -p "Entrez la deuxième variable: " var2

    case $choix in
        1)
            result=$((var1 + var2))
            echo "Résultat de l'addition : $var1 + $var2 = $result"
            ;;
        2)
            result=$((var1 - var2))
            echo "Résultat de la soustraction : $var1 - $var2 = $result"
            ;;
        3)
            result=$((var1 * var2))
            echo "Résultat de la multiplication : $var1 * $var2 = $result"
            ;;
        4)
            if [ "$var2" -ne 0 ]; then
                result=$((var1 / var2))
            fi
        *)
            echo "Option invalide. Veuillez réessayer."
        esac
    done
```

```

        echo "Résultat de la division : $var1 / $var2 = $result"
    else
        echo "Erreur : Division par zéro n'est pas possible."
    fi
;;
*)
    echo "Option invalide, veuillez choisir une option entre 1 et 5."
;;
esac
echo
done

```

#### 4. Créez un script qui utilise une boucle until pour demander à l'utilisateur de fournir un fichier existant et un mot à rechercher.

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

```
# Script pour demander un fichier existant et un mot à rechercher
```

```
fichier=""
mot=""
```

```
# Boucle jusqu'à ce qu'un fichier existant soit fourni
until [ -f "$fichier" ]; do
    read -p "Veuillez fournir le chemin du fichier existant : " fichier
    if [ ! -f "$fichier" ]; then
        echo "Le fichier $fichier n'existe pas. Veuillez essayer de nouveau."
    fi
done
```

```
# Demander un mot à rechercher
read -p "Veuillez fournir le mot à rechercher : " mot
```

```
# Rechercher le mot dans le fichier
echo "Recherche du mot '$mot' dans le fichier '$fichier'..."
grep -n "$mot" "$fichier"
```

```
# Afficher un message si le mot n'est pas trouvé
if [ $? -ne 0 ]; then
    echo "Le mot '$mot' n'a pas été trouvé dans le fichier '$fichier'."
else
    echo "Le mot '$mot' a été trouvé dans le fichier '$fichier'."
fi
```

5. Créez un script qui demande un mot à l'utilisateur jusqu'à ce qu'il tape "Au revoir!".

```
#!/bin/bash

# Script qui demande un mot à l'utilisateur jusqu'à ce qu'il tape "Au revoir!"

mot=""

until [ "$mot" == "Au revoir!" ]; do
    read -p "Veuillez entrer un mot (tapez 'Au revoir!' pour quitter) : " mot
    if [ "$mot" != "Au revoir!" ]; then
        echo "Vous avez entré : $mot"
    fi
done

echo "Au revoir!"
```

6. Créez un script qui génère la table de multiplication ou division pour un nombre donné.

```
#!/bin/bash

# Script pour générer une table de multiplication ou de division pour un nombre donné

read -p "Choisissez l'opération (1 pour multiplication, 2 pour division) : " operation
read -p "Entrez le nombre pour lequel vous voulez générer la table : " nombre

if [ "$operation" -eq 1 ]; then
    echo "Table de multiplication pour $nombre"
    for i in {1..10}; do
        result=$((nombre * i))
        echo "$nombre x $i = $result"
    done
elif [ "$operation" -eq 2 ]; then
    echo "Table de division pour $nombre"
    for i in {1..10}; do
        if [ $i -ne 0 ]; then
            result=$(echo "scale=2; $nombre / $i" | bc)
            echo "$nombre / $i = $result"
        fi
    done
else
    echo "Opération invalide. Veuillez choisir 1 pour multiplication ou 2 pour division."
fi
```

## 7. Créez un Script pour Générer un Récit

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

```
# Script pour générer un récit interactif
```

```
# Demander les informations à l'utilisateur
```

```
read -p "Entrez un nom : " nom
```

```
read -p "Entrez un lieu : " lieu
```

```
read -p "Entrez un objet : " objet
```

```
read -p "Entrez une action (verbe) : " action
```

```
# Générer le récit
```

```
echo ""
```

```
echo "Voici votre récit :"
```

```
echo "Il était une fois, dans un lieu nommé $lieu, un personnage nommé $nom."
```

```
echo "Un jour, $nom a trouvé un $objet mystérieux."
```

```
echo "Avec beaucoup de courage, $nom a décidé de $action le $objet."
```

```
echo "Cette aventure a changé la vie de $nom pour toujours."
```

```
echo ""
```

```
# Fin du script
```

```
echo "Merci d'avoir créé ce récit avec nous !"
```

8. Créez un Script Collecter les Informations d'étudiant.

```
done
```

```
# Demander l'email et vérifier qu'il contient le caractère @
```

```
while true; do
```

```
read -p "Entrez votre adresse email : " email
```

```
if [[ "$email" =~ "@" ]]; then
```

```
break
```

```
else
```

```
echo "L'adresse email doit contenir le caractère @. Essayez encore."
```

```
fi
```

```
done
```

```
# Choisir le sexe
```

```
echo "Choisissez votre sexe :"
```

```
echo "1. Homme"
```

```
echo "2. Femme"
```

```
echo "3. Autre"
```

```
while true; do
```

```
read -p "Votre choix (1/2/3) : " sexe
```

```
case $sexe in
```



```
1) sexe="Homme"; break ;;
2) sexe="Femme"; break ;;
3) sexe="Autre"; break ;;
*) echo "Choix invalide. Veuillez entrer 1, 2 ou 3." ;;

esac

done

# Afficher les informations

echo ""

echo "Informations collectées :"

echo "Nom complet : $nom_complet"

echo "Date de naissance : $date_naissance"

echo "Âge : $age ans"

echo "Adresse : $adresse"

echo "Numéro de téléphone : $tel"

echo "Email : $email"

echo "Sexe : $sexe"

#!/bin/bash

# Script pour collecter les informations des étudiants

# Fichier de sortie pour stocker les informations
```

```
fichier_etudiants="informations_etudiants.txt"
```

```
# Fonction pour ajouter les informations d'un étudiant au fichier
```

```
ajouter_etudiant() {
```

```
    echo "Nom complet: $nom_complet" >> "$fichier_etudiants"
```

```
    echo "Date de naissance: $date_naissance" >> "$fichier_etudiants"
```

```
    echo "Âge: $age ans" >> "$fichier_etudiants"
```

```
    echo "Adresse: $adresse" >> "$fichier_etudiants"
```

```
    echo "Numéro de téléphone: $tel" >> "$fichier_etudiants"
```

```
    echo "Email: $email" >> "$fichier_etudiants"
```

```
    echo "Sexe: $sexe" >> "$fichier_etudiants"
```

```
    echo "-----" >> "$fichier_etudiants"
```

```
}
```

```
# Boucle pour collecter les informations des étudiants
```

```
while true; do
```

```
    read -p "Entrez le nom complet de l'étudiant (ou tapez 'quitter' pour terminer) : " nom_complet
```

```
    if [ "$nom_complet" == "quitter" ]; then
```

```
        echo "Fin de la collecte des informations."
```

```
        break
```

```
    fi
```

```
    read -p "Entrez la date de naissance (JJ/MM/AAAA) : " date_naissance
```

```
    read -p "Entrez l'âge de l'étudiant : " age
```

```
    read -p "Entrez l'adresse : " adresse
```

```
    read -p "Entrez le numéro de téléphone : " tel
```

```
# Demander l'email et vérifier qu'il contient le caractère @
```

```
while true; do
```

```
read -p "Entrez votre adresse email : " email
if [[ "$email" =~ "@" ]]; then
    break
else
    echo "L'adresse email doit contenir le caractère @. Essayez encore."
fi
done
```

```
# Choisir le sexe
echo "Choisissez votre sexe :"
echo "1. Homme"
echo "2. Femme"
echo "3. Autre"
while true; do
    read -p "Votre choix (1/2/3) : " sexe
    case $sexe in
        1) sexe="Homme"; break ;;
        2) sexe="Femme"; break ;;
        3) sexe="Autre"; break ;;
        *) echo "Choix invalide. Veuillez entrer 1, 2 ou 3." ;;
    esac
done
```

```
# Ajouter les informations de l'étudiant au fichier
ajouter_etudiant
```

```
# Afficher les informations collectées
echo ""
echo "Informations collectées :"
```

```

echo "Nom complet : $nom_complet"
echo "Date de naissance : $date_naissance"
echo "Âge : $age ans"
echo "Adresse : $adresse"
echo "Numéro de téléphone : $tel"
echo "Email : $email"
echo "Sexe : $sexe"
echo ""
done

# Fin du script

echo "Toutes les informations ont été collectées et enregistrées dans $fichier_etudiants."

```

9. Créez un script qui demande à l'utilisateur de fournir le nom du dossier, demande à l'utilisateur combien de fichiers texte il souhaite créer, crée les fichiers texte et les ajoute au dossier, liste tous les fichiers créés.

```

#!/bin/bash

# Demande à l'utilisateur de fournir le nom du dossier
read -p "Entrez le nom du dossier : " dossier

# Crée le dossier s'il n'existe pas déjà
if [ ! -d "$dossier" ]; then
    mkdir -p "$dossier"
    echo "Le dossier '$dossier' a été créé."
else
    echo "Le dossier '$dossier' existe déjà."
fi

# Demande à l'utilisateur combien de fichiers texte il souhaite créer
read -p "Combien de fichiers texte souhaitez-vous créer ? : " nombre_fichiers

# Boucle pour créer les fichiers texte

```

```
for ((i=1; i<=nombre_fichiers; i++)); do
    fichier="$dossier/fichier_$i.txt"
    touch "$fichier"
    echo "Fichier $fichier créé."
done

# Liste tous les fichiers créés
echo ""
echo "Liste des fichiers créés dans le dossier '$dossier' :"
ls -l "$dossier"
```

conclusion :

Par ce TD des notions de scripts aborder , j'arrive par créer des fichiers et dossiers , comprende le script