



Shell - MEMO

[Important](#)

[Introduction](#)

[Condition](#)

[Boucle](#)

[Menu](#)

[Vérifier qu'un flag existe](#)

[Vérifier qu'un utilisateur existe](#)

[El Blanc](#)

[Exemple 1 - gestion des utilisateurs](#)

[Exemple 2 - Agenda](#)

Important

- Utiliser uniquement des fonctions compatible **UNIX**.
- Mettre des commentaires
 - `#!/bin/bash`
 - Nom et prénom
 - Date du jour
 - Intitulé du script (ce qu'il fait)
 - Dans le cas de passages complexes

Introduction

Condition

```
#!/bin/bash

a=1
if [ $a -eq 1 ]; then
    echo a=1
else
```

```

    echo a!=1
fi

a=1
if [ $a -eq 1 ]
then
    echo a=1
else
    echo a!=1
fi

if [ -s toto ]
then
    echo "Le fichier toto existe"
else
    echo "le fichier n'existe pas"
fi

if test -s toto
then
    echo "existe"
else
    echo "n'existe pas"
fi

```

Boucle

```

#!/bin/bash

a=10
while [ $a -gt 0 ]
do
    echo a=$a
    a=`expr $a - 1`
done

for i in 1 2 3
do
    echo i=$i
done

for j in $(seq 24)
do
    echo j=$j
done

for k in `ls`
do
    echo k=$k
done

```

Menu

```
#!/bin/bash

displayChoix()
{
    echo "Choix $1 sélectionné"
}

menuVariable=10

while [ $menuVariable -ne 0 ]; do
    echo ""
    echo -e "\e[32mMenu \e[39m"
    echo -e "  1 : \e[32mChoix 1 \e[39m"
    echo -e "  2 : \e[32mChoix 2 \e[39m"
    echo -e "  3 : \e[32mChoix 3 \e[39m"
    echo -e "  4 : \e[32mChoix 4 \e[39m"
    echo -e "  0 : \e[32mQuitter \e[39m"
    read -p "Votre choix : " menuVariable
    case $menuVariable in
        1|2|3|4) displayChoix $menuVariable;;
        0) echo "Au revoir (ノ◕〇◕)ノ ⅃"; exit;;
        *) echo "Choix introuvable"; menuVariable=10;;
    esac
done
```

Vérifier qu'un flag existe

```
#!/bin/bash

# Si pas de paramètre
if [ -z $1 ]; then
    echo -e "\e[31mERROR \e[39m: Vous devez renseigner un flag"
    exit
# Si plus de 4 paramètres
elif [ ! -z $5 ]; then
    echo -e "\e[31mERROR \e[39m: Vous devez renseigner au plus 4 flags"
    exit
fi

for arg in $*; do
    case $arg in
        "-a"|" -b"|" -c"|" -d");;
        *) echo -e "\e[31mInvalid flag\e[39m"; exit;;
    esac
done

for arg in $*; do
    case $arg in
        -a) echo "Flag a";;
```

```
-b) echo "Flag b";;  
-c) echo "Flag c";;  
-d) echo "Flag d";;  
esac  
done
```

Vérifier qu'un utilisateur existe

```
#!/bin/bash  
  
if [ -z $1 ]; then  
    echo -e "\e[31mERROR \e[39m: Vous devez renseigner un utilisateur"  
    exit  
else  
    if grep -wq ^$1 /etc/passwd; then  
        echo -e "\e[32mOK \e[39m: L'utilisateur existe"  
    else  
        echo -e "\e[31mFAILED \e[39m: L'utilisateur n'existe pas"  
    fi  
fi
```

El Blanc

Exemple 1 - gestion des utilisateurs

```
#!/bin/bash  
  
# Robin Bidanchon  
# 2021-02-22  
# Ce script permet de manipuler les utilisateurs d'un systeme linux  
  
# Récupère la liste des utilisateurs  
UsersList()  
{  
    usersList=`eval "cut -d: -f1 /etc/passwd"`  
    echo $usersList  
}  
  
# Retourne le nombre d'utilisateurs  
NumberOfUsers()  
{  
    nbUsersValue=`eval "cat /etc/passwd | wc -l"`  
    echo "Il y a $nbUsersValue utilisateurs"  
}  
  
# Vérifie si un utilisateur peut se logger  
UserCanLogin()
```

```

{
    if grep -wq ^$1 /etc/passwd; then
        if grep -w ^$1 /etc/passwd | grep -wq nologin; then
            echo "Cet utilisateur ne peut pas se connecter"
        else
            echo "Cet utilisateur peut se connecter"
        fi
    else
        echo "Cet utilisateur n'existe pas"
    fi
fi
}

# Vérifie que le UID de l'utilisateur Root est dupliqué
RootUidDuplication()
{
    nbRootUid=`eval "cat /etc/passwd | grep ':0:' | wc -l"`
    if [ $nbRootUid -ne 1 ]; then
        echo "Il y a une duplication de l'utilisateur root (uid: 0)"
    else
        echo "Il y n'y a pas de duplication de l'utilisateur root"
    fi
}

menuVariable=10

while [ $menuVariable -ne 0 ]; do
    echo ""
    echo "Menu"
    echo "  1 : Lister les utilisateurs"
    echo "  2 : Nombre d'utilisateurs"
    echo "  3 : Un utilisateur peut se logger"
    echo "  4 : Duplication de l'utilisateur Root"
    echo "  0 : Quitter"
    read -p "Votre choix : " menuVariable
    case $menuVariable in
        1) UsersList;;
        2) NumberOfUsers;;
        3)
            read -p "Nom de l'utilisateur : " username
            UserCanLogin $username
            ;;
        4) RootUidDuplication;;
        0) echo "Au revoir (J'°□°) J' _ L L"; exit;;
        *) echo "Choix introuvable"; menuVariable=10;;
    esac
done

```

Exemple 2 - Agenda

```

#!/bin/bash

# Robin Bidanchon
# 2021-02-22
# Ce script permet de manipuler un agenda

```

```

# Création d'un nouveau rendez-vous
PrendreRDV()
{
    read -p "Date (jjmmaaaa) : " date
    read -p "Heure : " heure
    read -p "Objet : " objet
    echo "$date,$heure,$objet" >> data.txt
}

ConsulterRDV()
{
    if [ -z "cat data.txt" ]; then
        echo "Vous n'avez pas de rendez-vous"
    else
        read -p "Date (jjmmaaaa) : " date
        rendezVous=`eval "grep -w $date data.txt | wc -l"`
        if [ $rendezVous -eq 0 ]; then
            echo "Vous n'avez pas de rendez-vous ce jour-ci"
        else
            grep -w $date data.txt
        fi
    fi
}

SupprimerRDV() {
    cat data.txt
    read -p "Entrez la date des rendez-vous à supprimer (jjmmaaaa) : " dateRdv
    rendezVousDate=`eval "grep -w $dateRdv data.txt | wc -l"`
    if [ $rendezVousDate -eq 0 ]; then
        echo "Vous n'avez pas de rendez-vous ce jour-ci"
    else
        grep -v $dateRdv data.txt > data.tmp
        mv data.tmp data.txt
    fi
}

menuVariable=10

dateJour=`date +%d%m%Y`

if [ $# -eq 1 ] && [ $1 = "-d" ]; then
    for i in `cut -d, -f1 data.txt`; do
        if [ $i -lt $dateJour ]; then
            grep -v $i data.txt > data.tmp
            mv data.tmp data.txt
        fi
    done
fi

while [ $menuVariable -ne 0 ]; do
    echo ""
    echo "Menu"
    echo " 1 : Prendre un rendez-vous"
    echo " 2 : Consulter les rendez-vous par date"
    echo " 3 : Supprimer un rendez-vous"
    echo " 0 : Quitter"
    read -p "Votre choix : " menuVariable

```

```
case $menuVariable in
  1) PrendreRDV;;
  2) ConsulterRDV;;
  3) SupprimerRDV;;
  0) echo "Au revoir (J'°□°) J'°_°_°"; exit;;
  *) echo "Choix introuvable"; menuVariable=10;;
esac
done
```