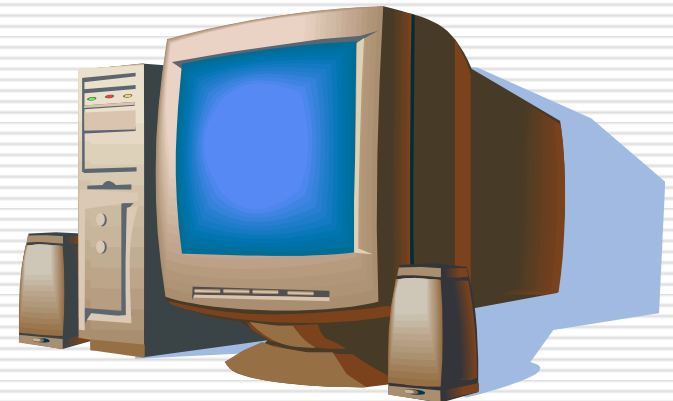
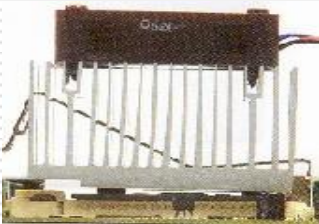
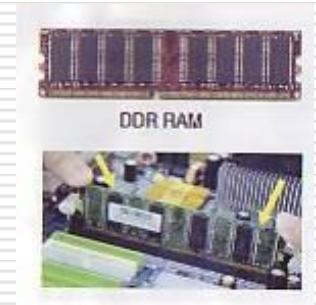
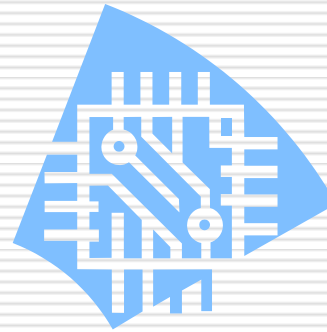
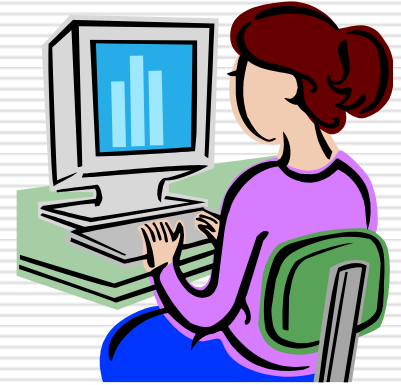


PROGRAMMATION SHELL





II. Programmation scripts shell

- Le shell est le programme qui gère l'invite de commandes. C'est le programme qui attend que vous rentriez des commandes, puis il les exécute. **C'est un interpréteur de commandes.**
- Un script shell permet d'automatiser une série d'opérations. Il se présente sous la forme d'un fichier contenant une ou plusieurs commandes.
- Il existe différents types de shell.
 - **sh** : *Bourne Shell*. L'ancêtre de tous les shells.
 - **bash** : *Bourne Again Shell*. Une amélioration du *Bourne Shell*.
 - **ksh** : *Korn Shell*.
 - **csh** : *C Shell*. Un shell utilisant une syntaxe proche du langage C.
 - **tcsh** : *Tenex C Shell*. Amélioration du *C Shell*.
 - **zsh** : *Z Shell*. Shell assez récent reprenant les meilleures idées de bash, ksh et tcsh.

■ Un exemple simple

```
echo $HOME
echo $PATH
LISTE=$1/*
for NOMFICH in $LISTE;
do
if test -d $NOMFICH
then
echo $NOMFICH
fi
done
```

Variables	NOMFICH LISTE
Variables d'environnement	HOME, PATH
Arguments	\$1
Structure de contrôle	Conditionnelle if... then...fi
Structure de contrôle	Boucle : for do done

```
linuxjojo@linuxjojo-VirtualBox:~$ ls -l Documents
total 20
-rw-rw-r-- 1 linuxjojo linuxjojo  40 oct.  6 18:05 ficheexemple
-rw-rw-r-- 1 linuxjojo linuxjojo   0 oct.  6 17:54 fichexemple
-rw-rw-r-- 1 linuxjojo linuxjojo 232 oct.  7 22:23 fichierexo2.txt
-rw-rw-r-- 1 linuxjojo linuxjojo 161 oct.  6 17:19 fichiernom.txt
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 titi
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 toto
linuxjojo@linuxjojo-VirtualBox:~$ ./script1 Documents
/home/linuxjojo
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
Documents/titi
Documents/toto
linuxjojo@linuxjojo-VirtualBox:~$
```

Script shell : les variables

- Une variable est un espace de stockage pour un résultat. Elle est caractérisée par un nom.
 - Nom variable [a-zA-Z]{1} [a-zA-Z0-9]*
 - Définie dès qu'elle est utilisée
 - Son contenu est considéré comme une suite de caractères

```
linuxjojo@linuxjojo-VirtualBox:~$ TEXTE="bonjour les amis"
linuxjojo@linuxjojo-VirtualBox:~$ echo $TEXTE
bonjour les amis
linuxjojo@linuxjojo-VirtualBox:~$
```

- TEXTE est le nom de la variable.
- Elle est initialisée par TEXTE=valeur (pas d'espace au niveau du =)
- Le contenu de la variable TEXTE est \$TEXTE;

La commande « echo » écrit sur sa sortie les arguments qui lui sont fournis

Script shell : les variables d'environnements

- Un certain nombre de variables sont définies dans l'environnement du shell et sont nécessaires au fonctionnement de l'interpréteur.
- La commande `printenv` permet d'afficher ces variables

HOME	Définit le répertoire d'accueil	<code>\$>\$HOME</code> /home/linuxjojo
PATH	Liste des répertoires explorés par le shell pour trouver une commande	<code>\$>\$PATH</code> bash:/usr/local/bin:/usr/bin:/usr/sbin:
PWD	Valeur du répertoire courant	<code>\$>\$PWD</code> /home/linuxjojo/Documents
USER	Nom de l'utilisateur	<code>\$>\$USER</code> linuxjojo
SHELL	Nom du shell utilisé	<code>\$>\$SHELL</code> /bin/bash

Script shell

■ Un exemple simple

echo \$HOME

echo \$PATH

```
LISTE=$1/*
for NOMFICH in $LISTE;
do
if test -d $NOMFICH
then
echo $NOMFICH
fi
done
```

Variables	NOMFICH LISTE
Variables d'environnement	HOME, PATH
Arguments	\$1
Structure de contrôle	Conditionnelle if... then...fi
Structure de contrôle	Boucle : for do done

```
linuxjojo@linuxjojo-VirtualBox:~$ ls -l Documents
total 20
-rw-rw-r-- 1 linuxjojo linuxjojo  40 oct.  6 18:05 ficheexemple
-rw-rw-r-- 1 linuxjojo linuxjojo   0 oct.  6 17:54 ficheexemple
-rw-rw-r-- 1 linuxjojo linuxjojo 232 oct.  7 22:23 fichierexo2.txt
-rw-rw-r-- 1 linuxjojo linuxjojo 161 oct.  6 17:19 fichiernom.txt
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 titi
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 toto
linuxjojo@linuxjojo-VirtualBox:~$ ./script1 Documents
/home/linuxjojo
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
Documents/titi
Documents/toto
linuxjojo@linuxjojo-VirtualBox:~$
```

- Modifier une variable : exemple `PATH`.

```
linuxjojo@linuxjojo-VirtualBox:~/Documents$ $PATH
bash: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/u
sr/local/games: Aucun fichier ou dossier de ce type
linuxjojo@linuxjojo-VirtualBox:~/Documents$ date
mardi 13 octobre 2015, 16:31:22 (UTC+0200)
linuxjojo@linuxjojo-VirtualBox:~/Documents$ ls /bin | grep ^"date"
date
linuxjojo@linuxjojo-VirtualBox:~/Documents$
```

- La commande `date` est trouvée et s'exécute.
- Elle est en effet dans le répertoire `/bin`. Ce chemin apparait dans la variable `PATH`.

■ Modifier une variable : exemple PATH.

```
linuxjojo@linuxjojo-VirtualBox:~$ pwd
/home/linuxjojo
linuxjojo@linuxjojo-VirtualBox:~$ ls Documents
exemple ficheexemple fichexemple fichierexo2.txt fichiernom.txt titi toto
linuxjojo@linuxjojo-VirtualBox:~$ $PATH
bash: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin: Aucun fichier
ou dossier de ce type
linuxjojo@linuxjojo-VirtualBox:~$ exemple
exemple : commande introuvable
linuxjojo@linuxjojo-VirtualBox:~$ Documents/exemple
bonjour les petits amis
la valeur de l'entier est 20
linuxjojo@linuxjojo-VirtualBox:~$ PATH=$PATH:$HOME:$HOME/Documents
linuxjojo@linuxjojo-VirtualBox:~$ exemple
bonjour les petits amis
la valeur de l'entier est 20
linuxjojo@linuxjojo-VirtualBox:~$ $PATH
bash: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/home/linuxjo
jo:/home/linuxjojo/Documents: Aucun fichier ou dossier de ce type
linuxjojo@linuxjojo-VirtualBox:~$
```

on modifie la variable
PATH en y ajoutant le
répertoire HOME et
HOME/Documents dans
lequel se trouve
l'exécutable exemple

Maintenant, le
programme est trouvé

Script shell : les arguments de la ligne de commandes

- Le script shell récupère les arguments de la ligne de commande dans des variables réservées appelées Paramètres Positionnels

\$#	Nombre d'arguments reçus par le script
\$0	Le nom du script lui-même
\$1 \$2 ... \$9 \${10}	\$1 est la valeur du premier argument, \$2 la valeur du second....
\$*	Liste des arguments

Script shell

■ Un exemple simple

```
echo $HOME
```

```
echo $PATH
```

```
LISTE=$1/*
```

```
for NOMFICH in $LISTE;
do
if test -d $NOMFICH
then
echo $NOMFICH
fi
done
```

Variables	NOMFICH LISTE
Variables d'environnement	HOME, PATH
Arguments	\$1
Structure de contrôle	Conditionnelle if... then...fi
Structure de contrôle	Boucle : for do done

```
linuxjojo@linuxjojo-VirtualBox:~$ ls -l Documents
total 20
-rw-rw-r-- 1 linuxjojo linuxjojo  40 oct.  6 18:05 ficheexemple
-rw-rw-r-- 1 linuxjojo linuxjojo   0 oct.  6 17:54 ficheexemple
-rw-rw-r-- 1 linuxjojo linuxjojo 232 oct.  7 22:23 fichierexo2.txt
-rw-rw-r-- 1 linuxjojo linuxjojo 161 oct.  6 17:19 fichiernom.txt
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 titi
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 toto
linuxjojo@linuxjojo-VirtualBox:~$ ./script1 Documents
/home/linuxjojo
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
Documents/titi
Documents/toto
linuxjojo@linuxjojo-VirtualBox:~$
```

\$1 = Documents

- Les structures de contrôles sont les éléments du langage qui déterminent l'ordre dans lequel les instructions sont exécutées.
- La séquence : exécution en séquence des commandes
- La conditionnelle : introduit un aiguillage, un choix entre des ensembles de commandes
- La boucle : introduit une itération, un ensemble de commandes est exécuté plusieurs fois.

- ❑ La conditionnelle est une structure de contrôle qui permet de lier l'exécution d'une séquence de commandes au résultat d'un test. Le test est appelé **condition** ou **prédicat**.

- Exemple : on doit calculer le prix final d'une commande. Si le montant de la commande est inférieur à 200 euros, les frais de port sont de 20 euros et sinon de 5 euros

```
Si (prix_final < 200)
Alors
    Prix = prix_final + 20
Sinon
    Prix = prix_final + 5
Fsi
```

- La conditionnelle est une structure de contrôle qui permet de lier l'exécution d'une séquence de commandes au résultat d'un test. Le test est appelé **condition** ou **prédicat**.

Si (**condition**)

Alors

Ensemble d'actions 1

Sinon

Ensemble d'actions 2

Fsi

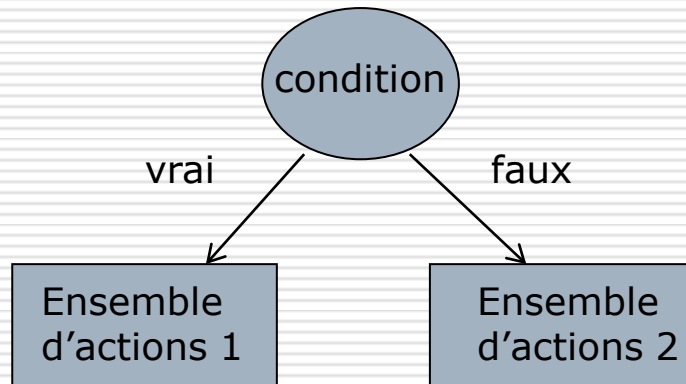
Si (**condition**)

Alors

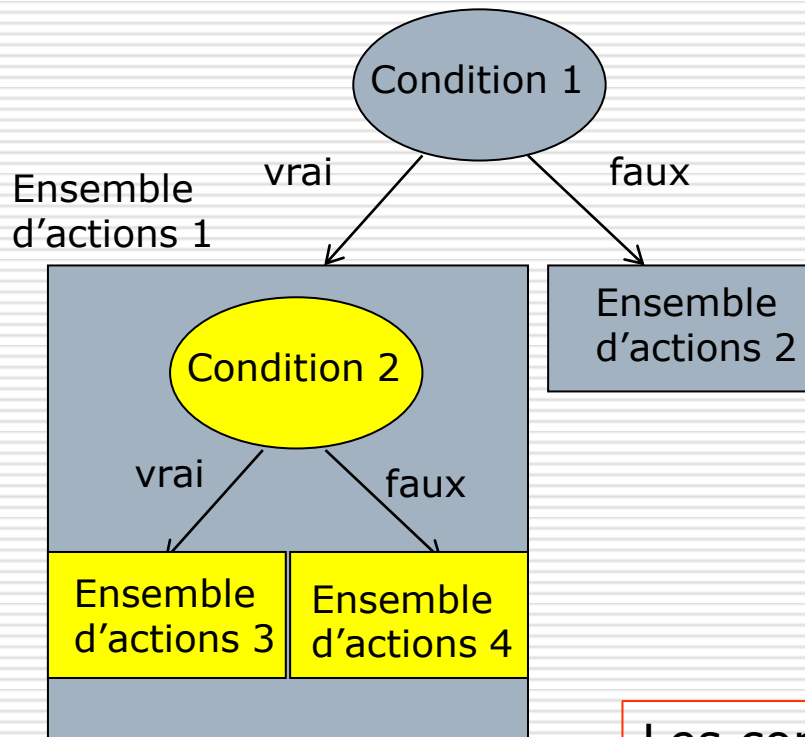
Ensemble d'actions 1

Fsi

L'ensemble d'actions 2 peut être vide



- La conditionnelle est une structure de contrôle qui permet de lier l'exécution d'une séquence de commandes au résultat d'un test. Le test est appelé **condition** ou **prédicat**.



```

Si (condition1)
Alors
    Si (condition2)
    Alors
        Ensemble d'actions 3
    Sinon
        Ensemble d'actions 4
    Fsi
Sinon
    Ensemble d'actions 2
Fsi

```

Les conditionnelles peuvent s'imbriquer

- La conditionnelle est une structure de contrôle qui permet de lier l'exécution d'une séquence de commandes au résultat d'un test. Le test est appelé **condition** ou **prédicat**.

Si (**condition**)
Alors
 Ensemble d'actions 1
Sinon
 Ensemble d'actions 2
Fsi

```
if (condition)  
then  
    Ensemble de commandes 1  
else  
    Ensemble de commandes 2  
fi
```

Syntaxe en script shell

■ Un exemple simple

```
echo $HOME
echo $PATH
LISTE=$1/*
for NOMFICH in $LISTE;
do
if test -d $NOMFICH
then
echo $NOMFICH
fi
done
```

condition

Variables	NOMFICH LISTE
Variables d'environnement	HOME, PATH
Arguments	\$1
Structure de contrôle	Conditionnelle if... then...fi
Structure de contrôle	Boucle : for do done

```
linuxjojo@linuxjojo-VirtualBox:~$ ls -l Documents
total 20
-rw-rw-r-- 1 linuxjojo linuxjojo  40 oct.  6 18:05 ficheexemple
-rw-rw-r-- 1 linuxjojo linuxjojo   0 oct.  6 17:54 fichexemple
-rw-rw-r-- 1 linuxjojo linuxjojo 232 oct.  7 22:23 fichierexo2.txt
-rw-rw-r-- 1 linuxjojo linuxjojo 161 oct.  6 17:19 fichiernom.txt
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 titi
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 toto
linuxjojo@linuxjojo-VirtualBox:~$ ./script1 Documents
/home/linuxjojo
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
Documents/titi
Documents/toto
linuxjojo@linuxjojo-VirtualBox:~$
```

- ❑ La commande `test` permet de faire des tests sur des fichiers, des chaînes de caractères et des nombres. Elle renvoie la valeur 0 (vrai) ou 1 (faux), qui peut être affiché par `$?`.

<code>if test -d \$NOMFICH</code>	Si le fichier <code>\$NOMFICH</code> est un
<code>then</code>	répertoire
<code>echo \$NOMFICH</code>	alors
<code>fi</code>	afficher le nom du fichier
	sinon rien
	fsi

□ La commande `test`

■ Sur les fichiers

<code>test -d nomfic</code>	vrai si le fichier est un répertoire
<code>test -a nomfic</code>	vrai si le fichier existe
<code>test -f nomfic</code>	vrai si le fichier est un fichier ordinaire
<code>test -r nomfic</code>	vrai si le fichier est accessible en lecture
<code>test -w nomfic</code>	vrai si le fichier est accessible en écriture
<code>test -x nomfic</code>	vrai si le fichier est accessible en exécution

```
linuxjojo@linuxjojo-VirtualBox:~$ ls -l exemple
-rwxrwxr-x 1 linuxjojo linuxjojo 8616 sept. 19 19:02 exemple
linuxjojo@linuxjojo-VirtualBox:~$ test -x exemple
linuxjojo@linuxjojo-VirtualBox:~$ echo $?
0
linuxjojo@linuxjojo-VirtualBox:~$ ls -l fichtrace
-rw-rw-r-- 1 linuxjojo linuxjojo 53 sept. 19 19:14 fichtrace
linuxjojo@linuxjojo-VirtualBox:~$ test -x fichtrace
linuxjojo@linuxjojo-VirtualBox:~$ echo $?
1
linuxjojo@linuxjojo-VirtualBox:~$
```

□ La commande test

■ Sur les chaines de caractères

test --z ch1	vrai si la chaine ch1 est de longueur 0
test --n ch1	vrai si la chaine ch1 n'est pas de longueur 0
test ch1 = ch2	vrai si les deux chaines sont égales
test ch1 != ch2	vrai si si les deux chaines sont différentes

```
linuxjojo@linuxjojo-VirtualBox:~$ ch1=bonjour
linuxjojo@linuxjojo-VirtualBox:~$ ch2=aurevoir
linuxjojo@linuxjojo-VirtualBox:~$ test ch1 = ch2
linuxjojo@linuxjojo-VirtualBox:~$ echo $?
1
linuxjojo@linuxjojo-VirtualBox:~$ test ch1 != ch2
linuxjojo@linuxjojo-VirtualBox:~$ echo $?
0
linuxjojo@linuxjojo-VirtualBox:~$ ch2=bonjour
linuxjojo@linuxjojo-VirtualBox:~$ test ch1 = ch2
linuxjojo@linuxjojo-VirtualBox:~$ echo $?
1
linuxjojo@linuxjojo-VirtualBox:~$
```

□ La commande `test`

■ Sur les nombres

<code>test nb1 -eq nb2</code>	vrai si nb1 est égal à nb2
<code>test nb1 -ne nb2</code>	vrai si nb1 est différent de nb2
<code>test nb1 -lt nb2</code>	vrai si nb1 est strictement inférieur à nb2
<code>test nb1 -le nb2</code>	vrai si nb1 est inférieur ou égal à nb2
<code>test nb1 -gt nb2</code>	vrai si nb1 est strictement supérieur à nb2
<code>test nb1 -ge nb2</code>	vrai si nb1 est supérieur ou égal à nb2

□ La commande test

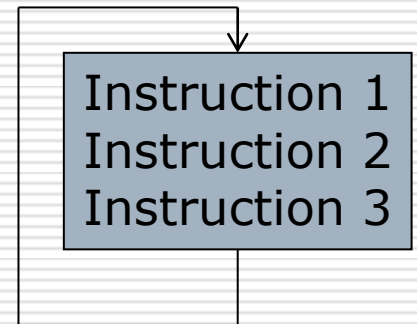
■ Sur les nombres

```
if test $1 -eq $2
then
    echo "les deux nombres sont égaux"
fi
if test $1 -ne $2
then
    echo "les deux nombres sont différents"
    if test $1 -lt $2
    then
        echo "nb1 plus petit que nb2"
    else
        echo "nb1 plus grand que nb2"
    fi
fi
```

```
linuxjojo@linuxjojo-VirtualBox:~$ ./testnombre 5 5
les deux nombres sont égaux
linuxjojo@linuxjojo-VirtualBox:~$ ./testnombre 9 7
les deux nombres sont différents
nb1 plus grand que nb2
linuxjojo@linuxjojo-VirtualBox:~$ ./testnombre 7 9
les deux nombres sont différents
nb1 plus petit que nb2
```

- Une **boucle** est une structure de contrôle itérative qui permet de répéter un traitement un certain nombre de fois.

- Il existe différents types de boucle.
 - Boucle while ou Tant que
 - Boucle For ou Pour



- Il existe différents types de boucle.

```
while (condition)
{
/* Instructions à répéter */
}
```

Boucle tant que (while) : les instructions sont itérées tant que la condition est vraie

Exemples : parcours d'une liste de fichiers ou d'un tableau de 10 entiers

```
tab [1..10] of integer;
i = 1;
While (i < 11)
{
    afficher (tab[i]);
    i = i + 1;
}
```


- Il existe différents types de boucle.

For (liste de valeurs val1,
val2, ... val n))

```
{  
/* Instructions à répéter */  
}
```

Boucle for ou pour : les instructions sont itérées en traitant la liste de valeurs fournies :

**val 1 à itération 1
val 2 à itération 2
val n à itération n**

Exemples : parcours d'une liste de fichiers ou d'un tableau de 10 entiers

```
tab [1..10] of integer;  
;  
For (i de 1 à 10)  
{  
    afficher (tab[i];  
}
```

□ Boucle While en shell

```
while (condition)
{
/* Instructions à répéter */
}
```

```
numero=1
while test $numero -ne 51
do
# la commande "touch" permet de créer un
fichier vide :

touch fichier"$numero"

# cette commande ajoute 1 à la variable
"numero" :

numero=$(( $numero + 1 ))
done
```

```
while (condition)
do
commande1
commande2
...
commande n
done
```

Script shell : Les structures de contrôle

□ Boucle For en shell

```
For (liste de valeurs val1, val2, ...  
val n))  
{  
/* Instructions à répéter */  
}
```

```
echo $HOME  
echo $PATH  
LISTE=$1/*  
for NOMFICH in $LISTE;  
do  
if test -d $NOMFICH  
then  
echo $NOMFICH  
fi  
done
```

```
for (var) in val1 val 2 val n  
do  
commande1  
commande2  
...  
commande n  
done
```

■ Un exemple simple

```
echo $HOME
echo $PATH
LISTE=$1/*
for NOMFICH in $LISTE;
do
if test -d $NOMFICH
then
echo $NOMFICH
fi
done
```

Variables	NOMFICH LISTE
Variables d'environnement	HOME, PATH
Arguments	\$1
Structure de contrôle	Conditionnelle if... then...fi
Structure de contrôle	Boucle : for do done

```
linuxjojo@linuxjojo-VirtualBox:~$ ls -l Documents
total 20
-rw-rw-r-- 1 linuxjojo linuxjojo  40 oct.  6 18:05 ficheexemple
-rw-rw-r-- 1 linuxjojo linuxjojo   0 oct.  6 17:54 fichexemple
-rw-rw-r-- 1 linuxjojo linuxjojo 232 oct.  7 22:23 fichierexo2.txt
-rw-rw-r-- 1 linuxjojo linuxjojo 161 oct.  6 17:19 fichiernom.txt
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 titi
drwxrwxr-x 2 linuxjojo linuxjojo 4096 oct. 11 22:22 toto
linuxjojo@linuxjojo-VirtualBox:~$ ./script1 Documents
/home/linuxjojo
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
Documents/titi
Documents/toto
linuxjojo@linuxjojo-VirtualBox:~$
```

Script shell : Calcul arithmétique

```
numero=$((numero + 1))
```

Les calculs arithmétiques sont représentés par l'opérateur `$((opérations))`. Les 4 opérateurs de base sont `+` `-` `*` `/`.

```
linuxjojo@linuxjojo-VirtualBox:~/Documents$ echo $((2*(4+(10/2)) - 1))  
17  
linuxjojo@linuxjojo-VirtualBox:~/Documents$ echo $((7/3))  
2
```