

## Cours : les boucles

### La boucle énumérative

La boucle énumérative explore une énumération de valeurs texte avec une variable et exécute une section de script.

```
for x in un deux trois quatre
do
    echo le nombre est $x
done
```

produit en sortie :

```
le nombre est un
le nombre est deux
le nombre est trois
le nombre est quatre
```

Que s'est-il réellement passé ? le script a déclaré une nouvelle variable d'environnement `x` introduite par la boucle `for`. Cette variable explore la "collection" de valeurs obtenues par découpage lexical de la séquence derrière le mot clef `"in"`. On note alors que si ces valeurs sont des chaînes de caractères plus complexes, il convient de les délimiter correctement :

```
for x in 'le premier' 'le deuxième' 'le troisième' 'le quatrième'
do
    echo ce script est $x
done
```

N'importe quelle liste de valeurs textuelles peut être explorée par une syntaxe `for`, et en particulier des listes de noms de fichiers :

```
for f in /etc/r*
do
    if [ -d "$f" ]
    then
        echo $f est un répertoire
    else
        echo $f est un fichier
    fi
done
```

**ATTENTION :** On remarque ici que l'interprétation de `/etc/r*` comme un appel à la lecture du système de fichiers est totalement implicite. Ce n'est pas trivial. Ceci montre une orientation certaine de `bash` à être un langage de script orienté "système".

Le phénomène qui produit la liste déterminée par le motif variable est appelé l'**expansion de chaîne**.

Les variables texte s'expansent dans le système de fichier sur la base :

- d'un chemin absolu (commence par /)
- d'un chemin relatif (tout autre début)

## Etendre la liste de valeurs

Il est possible d'étendre la liste de valeurs en présentant à l'instruction for plusieurs variables à expander.

```
for f in /etc/r* /etc/e* $@
do
  if [ -d "$f" ]
  then
    echo $f est un répertoire
  else
    echo $f est un fichier
  fi
done
```

Cette nouvelle formule constitue la liste des fichiers commençant par "r" ou par "e" dans /etc et y ajoute les chemins donnés en ligne de commande.

## La boucle indéterminée

Lorsqu'on dit "indéterminée", on parle du nombre d'itérations et non pas de la condition d'arrêt. La condition pour laquelle s'arrête la boucle est (heureusement) connue, mais on ne sait pas au bout de combien d'itérations elle se réalise.

La boucle indéterminée est semblable à la boucle for, mais en remplaçant le mot clef par while.

```
x=10
while (( $x > 0 ))
do
  echo $x
  (( x=$x - 1 )) # le marquage arithmétique est indispensable ici
done
```

affichera :

```
10
9
8
7
6
5
4
3
2
1
```