

### Chapitre 3 : Les boucles

Les boucles servent à répéter une fonction un certain nombre de fois ou jusqu'à ce qu'une condition soit vraie.

#### a) Do...LpWhile

Cette boucle exécute les fonctions situées à « l'intérieur » tout pendant que la condition est vraie. Cela veut dire que dès que la condition ne sera plus vraie, le programme sortira de la boucle et continuera normalement.

Do  
(fonction 1)  
(fonction 2)  
...  
LpWhile (condition)

Do ?→A LpWhile A≠124
----------------------------

Le programme nous demandera d'entrer un nombre jusqu'à ce que celui-ci soit égal à 124. (Tout pendant que le nombre sera différent de 124 le programme retournera à la ligne qui suit le « Do »).

#### b) While...WhileEnd

Cette fonction permet de réaliser une boucle tant qu'une condition est respectée. Elle est identique à la fonction Do sauf que la condition est vérifiée en début de boucle et non en fin de boucle :

While A≠124 ?→A WhileEnd
--------------------------------

Cet exemple est identique au précédent.

#### c) For...Next

Cette boucle est la plus complexe. Elle exécute les fonctions les fonctions un certain nombre de fois.

For x-->X to x Step x  
(fonction 1)  
(fonction 2)  
...  
Next

```
For 0→C to 6 Step 2
?→A
Next
```

Ici, il exécute 3 fois la fonction car au départ  $C=0$  et à chaque passage il incrémente  $C$  de 2 ( $C=C+2$ ) jusqu'à ce que  $C$  soit égal à 6.

Ce qui équivaut à :

```
0→C
Do
C+2→C
?→A
LpWhile
```

#### **d) Lbl...Goto**

Lbl...Goto peut être considéré comme une boucle car elle permet de faire un saut à un endroit déterminé.

```
Lbl x
(fonction 1)
(fonction 2)
...
Goto x
```

$x$  correspond à un chiffre de 0 à 9 ou à une lettre de A à Z.

```
Lbl 1
?→A
Goto 1
```

Le programme nous demandera d'entrer un nombre une infinité de fois.

**Attention :** Les lettres utilisées avec les commandes Lbl...Goto non strictement rien à voir avec les variables. Cet exemple est donc impossible.

```
2→A
Lbl 2
(fonction)
Goto A
```

Il y a donc une commande qui nous permet de sortir d'une boucle : la commande « Break ».

Lorsque le programme sort d'une boucle, il va à la ligne située juste en dessous de cette boucle.

```
For 1→B to 30
?→A
If A=4
Then Break
IfEnd
Next
```

Le programme demandera d'entrer un nombre jusqu'à ce que A soit égal à 4.  
Dans cet exemple, vous avez le droit de vous tromper 30 fois.

**Attention :** « Break » ne peut pas être utilisé avec la double flèche mais seulement avec « If...Then...IfEnd ».

Il faut savoir que « Lbl...Goto » ne fait pas partie des boucles bien qu'elle peut se comporter comme une boucle, c'est pourquoi il ne faut utiliser la commande « Break » qu'avec les boucles « Do...LpWhile », « For...to...Next » et « While...WhileEnd ».

*Comme vous pouvez le constater, nous pouvons arriver à faire les mêmes choses avec des boucles différentes donc essayez de trouver celle qui convient le mieux en fonction de ce que vous voulez.*

### e) Prog...Return

La fonction « Prog » vous permet de lancer un autre programme.  
Elle vous permet donc de passer d'un programme principal à un sous-programme.  
Prog « Nom du programme »

```
Prog "New"
```

Lorsque vous êtes dans un sous-programme et que vous voulez revenir au programme principal, vous devez utiliser la commande « Return ».

```
Return
```

*Il est déconseillé d'abuser de ces fonctions.*