



5) Les Boucles

Les boucles (Structure Itérative) c'est un élément essentiel de la programmation et de l'algorithmie mais c'est aussi l'élément le plus compliqué en général à appréhender quand vous débutez. Dans un algorithme, utiliser une boucle permet de recommencer plusieurs fois un bloc d'instructions.

On utilisera dans un premier temps la boucle « **TANTQUE** » :

- **TANTQUE** *condition* **FAIRE**

...

FINTANTQUE

En gros tant que la condition est valide fait/répète ce qu'il y a dans le bloc. C'est pour ça qu'on appelle aussi les boucles « **structure Itérative** ». Pour mieux comprendre l'utilisation des boucles je vais vous donner deux exemples.

Imaginons je vous demande un petit exercice afficher les nombres de 1 à 10 :

```
2 DÉBUT
3   ECRIRE "Voici les chiffres de 1 à 10"
4   ECRIRE 1
5   ECRIRE 2
6   ECRIRE 3
7   ECRIRE 4
8   ECRIRE 5
9   ECRIRE 6
10  ECRIRE 7
11  ECRIRE 8
12  ECRIRE 9
13  ECRIRE 10
14 FIN
15
16
```

Console

Terminé

Voici les chiffres de 1 à 10

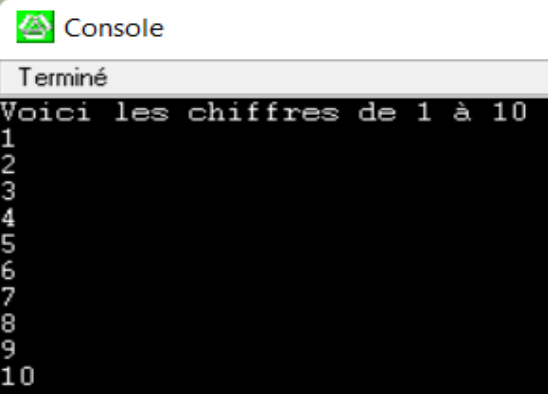
1
2
3
4
5
6
7
8
9
10

C'est simple j'ai qu'à écrire 10 fois le mot « **ECRIRE** » en changeant à chaque fois la valeur. Mais si je vous demandais les nombres de 1 à 1000, cette manière serait trop longue.

Les boucles existent pour ce genre de situation qui se répète.

Voilà comment je l'aurai fait avec une boucle :

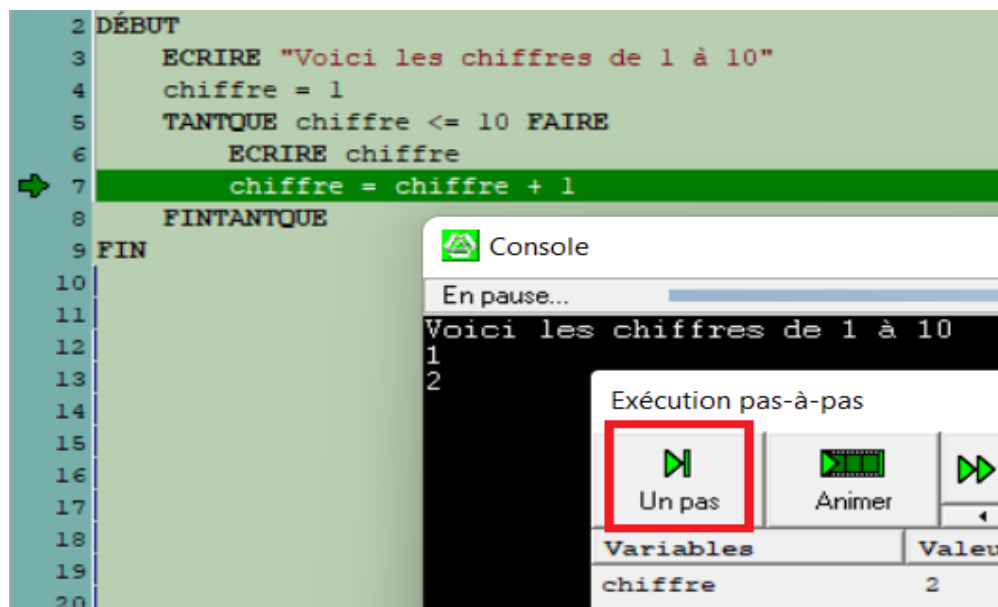
```
2 DÉBUT
3   ECRIRE "Voici les chiffres de 1 à 10"
4   chiffre = 1
5   TANTQUE chiffre <= 10 FAIRE
6     ECRIRE chiffre
7     chiffre = chiffre + 1
8   FINTANTQUE
9 FIN
```



Ici je crée une variable « chiffre » et je l'initialise avec la première valeur que je veux afficher c'est-à-dire « 1 ». Maintenant je crée ma boucle en mettant comme condition « tant que ma variable chiffre est plus petit ou égale à 10, fait ce qui suit dans le bloc ». Et dans ce bloc qu'est ce que je fais, bah j'affiche la variable chiffre ensuite je prend cette variable chiffre et je l'incrémente de 1.

Rappelez vous qu'on pouvait exécuter le pseudo-code « **pas par pas** », c'est ici avec les boucles qu'on l'utilisera le plus, pour bien voir ce qu'il s'y passe.

```
2 DÉBUT
3   ECRIRE "Voici les chiffres de 1 à 10"
4   chiffre = 1
5   TANTQUE chiffre <= 10 FAIRE
6     ECRIRE chiffre
7     chiffre = chiffre + 1
8   FINTANTQUE
9 FIN
```

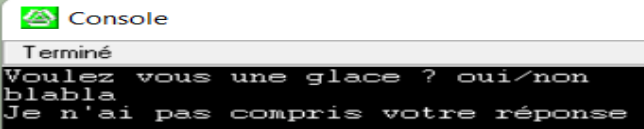


Donc si je reprends, au début notre chiffre vaut 1, le tant que vérifie est ce que « 1 » est plus petit ou égal à 10, si oui, il entre dans la boucle et affiche 1, puis il incrémente de 1, donc chiffre passe à « 2 ». Il retourne sur le tant que et vérifie si « 2 » est plus petit ou égal à 10, si oui, il entre dans la boucle et affiche 2, puis incrémente de 1, donc chiffre passe à « 3 » etc. Jusqu'à quand le chiffre est égal à 10, il check si « 10 » est plus petit ou égal à 10, si oui, il affiche 10 puis incrémente chiffre de 1, donc « 11 ». Il va ensuite checker si 11 est plus petit ou égal à 11, NON donc il sort de la boucle et le programme se termine.

Voici le deuxième exemple d'utilité.

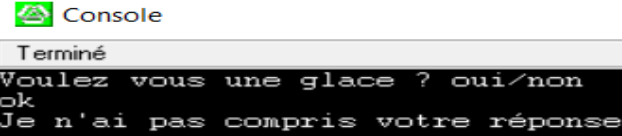
Imaginons je crée un algorithme qui vous sert des glaces. Il demandera à l'utilisateur si il veut une glace. L'utilisateur répondra oui ou non, je vais aussi gérer le cas où l'utilisateur introduit autre chose. Ça donnera quelque chose comme ça :

```
2 DÉBUT
3   ECRIRE "Voulez vous une glace ? oui/non"
4   LIRE reponse
5   SI reponse != "oui" ET reponse != "non" ALORS
6     ECRIRE "Je n'ai pas compris votre réponse"
7   FINSI
8 FIN
```



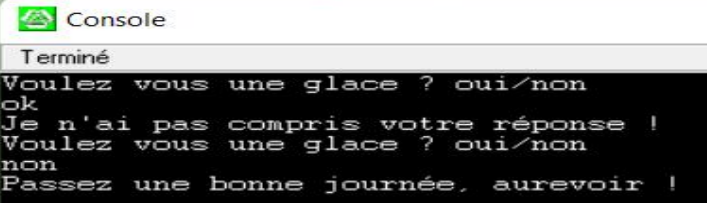
La on gère du coup quand une personne introduit autre chose que ce qui est demandé, par contre le programme s'arrête. Donc même si je complète mon algorithme :

```
2 DÉBUT
3   ECRIRE "Voulez vous une glace ? oui/non"
4   LIRE reponse
5   SI reponse != "oui" ET reponse != "non" ALORS
6     ECRIRE "Je n'ai pas compris votre réponse"
7   SINON
8     SI reponse = "oui" ALORS
9       ECRIRE "Voici votre glace, enjoy ;)"
10    SINON
11      ECRIRE "Passez une bonne journée, aurevoir"
12    FINSI
13  FINSI
14 FIN
```



Ici mon algorithme si je tape oui ou non ça fonctionnera logiquement, par contre quand je tape autre chose il me dit « Je n'ai pas compris votre réponse » et puis s'arrête. Dans la logique on voudrait qu'il redemande si on veut une glace ou pas. C'est là que les boucles aussi sont utiles :

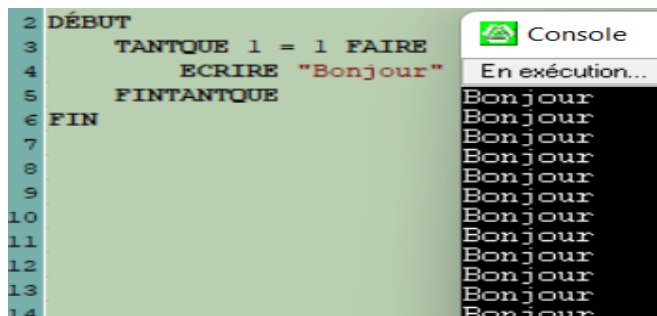
```
2 DÉBUT
3   ECRIRE "Voulez vous une glace ? oui/non"
4   LIRE reponse
5   TANTQUE reponse != "oui" ET reponse != "non" FAIRE
6     ECRIRE "Je n'ai pas compris votre réponse !"
7     ECRIRE "Voulez vous une glace ? oui/non"
8     LIRE reponse
9   FINTANTQUE
10  SI reponse = "oui" ALORS
11    ECRIRE "Voici votre glace, enjoy ;)"
12  SINON
13    ECRIRE "Passez une bonne journée, aurevoir !"
14  FINSI
15 FIN
```



Voilà ici tant qu'il n'aura pas oui ou non il continuera de poser la question.

Ici on voit toute l'importance d'une boucle. Imaginez un programme, ou même un site web ou on vous demande de vous connectez, vous faites une fausse manœuvre, il vous met que vous avez fait une erreur et puis il s'arrête. Ca n'a pas de sens, pareil vos cartes bancaire, imaginez si on ne vous laissait pas 3 tentatives pour le code PIN, et que ça s'arrêtait à la moindre erreur et ne rendait pas votre carte. Ces 3 tentatives c'est bien une boucle avec un compteur qui compte le nombre de fois qu'il passe par la boucle. On va essayer de le faire dans les exercices.

Attention les boucles c'est bien mais il faut bien faire attention à ne pas faire de boucle infini. Par exemple si j'écris :

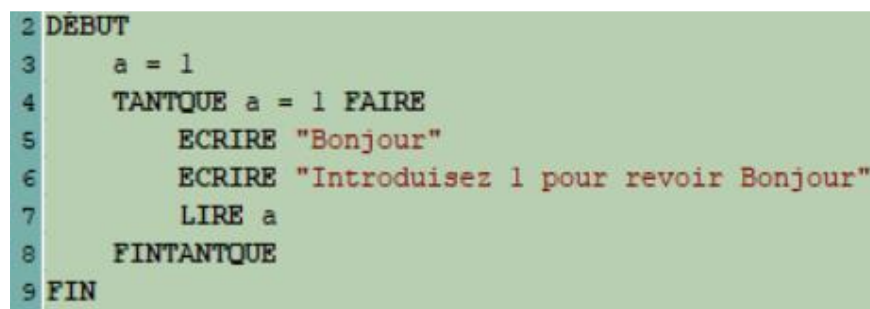


```
2 DÉBUT
3   TANTQUE 1 = 1 FAIRE
4       ECRIRE "Bonjour"
5   FINTANTQUE
6 FIN
```

The console window on the right shows the text "En exécution..." and a list of "Bonjour" messages, indicating that the program is running an infinite loop.

Ici vous le voyez bien, ma condition dans mon tant que ne changera jamais, 1 sera toujours égal à 1 du coup à chaque fois il rentrera dedans et il écrira « Bonjour ». Ca fera planter le programme.

Ne jamais oubliez de faire en sorte que votre condition puisse changer un moment donné, par exemple :



```
2 DÉBUT
3   a = 1
4   TANTQUE a = 1 FAIRE
5       ECRIRE "Bonjour"
6       ECRIRE "Introduisez 1 pour revoir Bonjour"
7       LIRE a
8   FINTANTQUE
9 FIN
```

On règle le problème de boucle infini parce qu'on a bien une condition susceptible de changer un moment donné en fonction de ce qu'on tapera au clavier dans ce cas-ci.

Si vous vous rappelez l'exemple avec les chiffres de 1 à 10, il y avait bien un changement, au fur et à mesure j'incrémentais de 1.

Je sais que ça peut paraître difficile pour certains, mais c'est avec de l'entraînement que vous y verrez plus clair. N'hésitez pas à vous entrainer en refaisant les exercices ou même les exemples que je donne chez vous.

La logique, il n'y a rien de magique, que de la pratique.

(Exos)

- 1) Créez moi un algorithme qui affiche de 0 à – 100 compris.
- 2) Créez moi un algorithme qui m’affiche **tous les nombres pairs** de 0 à 200 non compris tout deux.
- 3) Créez moi un algorithme qui m’affiche tous les nombres impairs de 0 à 100 compris.
- 4) Créez moi un algorithme qui me demande si j’ai faim, si oui il me propose du pain, si non il me dit bonne journée. Par contre si on met autre chose que oui ou non il me reposera la question de savoir si j’ai faim.
- 5) Créez un algorithme qui fait la Somme des nombres au clavier ca s’arrête dès que c’est plus de 1000.
- 6) (Difficile) Essayer d’écrire un programme qui demande d’introduire un chiffre et qui me sort les 10 premiers chiffre de la table de multiplication :

```
Veillez introduire un chiffre
6
Voici les 10 premiers multiples de 6
6 x 0 = 0
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
```

- 7) (Très Difficile) Essayez d’écrire un algorithme qui demande d’introduire un code PIN et qui vous laisse 3 tentatives pour l’introduire. Après 3 tentatives, il bloque votre carte si vous trouvé pas votre code (que vous aurez déclaré par exemple codePin = 1234) :

```
Terminé
Veillez introduire votre code PIN
1425
Veillez introduire votre code PIN
1254
Veillez introduire votre code PIN
1236
Vous avez raté 3 fois votre code PIN, carte bloquée !
```