

TP3 Python : Les boucles « tant que »

1) Les boucles « Tant que »

Les boucles sont indispensables quand on veut répéter plusieurs fois des instructions.

Dans ce diaporama, nous ne verrons que les boucles « Tant Que ».

En anglais, « Tant Que » se dit **while**.

La syntaxe de la boucle « Tant que » :

```
while condition :  
    instruction1  
    instruction2  
    etc.  
suite du programme
```

Ne pas oublier les tabulations !

Les **conditions** s'écrivent : *variable* *symbole* *variable ou nombre ou texte ou calcul*

où *symbole* est un des symboles suivants :

< > <= >= == != in

Exercice N°1 : La punition

Bart doit régulièrement recopier un texte un certain nombre de fois.

Le texte change chaque semaine (oui, il est collé toutes les semaines...) et le nombre de fois qu'il doit le recopier change aussi.

Écrivez un programme qui demanderait à Bart la phrase à recopier, le nombre de fois qu'il doit la recopier et qui fait la punition à sa place...

Testez ce programme bien sûr.

Exercice N°2 : Code secret

Samuel a oublié son code pin. Il a droit à 5 essais, après quoi son téléphone sera bloqué !

Le code oublié est 1234 (dur à oublier pourtant...).

Écrivez et testez un programme qui demandera à Samuel son code et bloquera son téléphone au bout de cinq erreurs (votre programme affichera simplement « Téléphone bloqué » dans ce cas).

- Exemple 1 : la machine doit afficher le texte « Bonjour » cinquante fois.

```
i=1
while i<=50 :
    print("Bonjour") .....
```

Un problème apparaît : la boucle ne s'arrête jamais (on parle de *boucle infinie*) car i reste égal à 1 donc la condition $i \leq 50$ est toujours vraie.

Il faut donc augmenter la valeur de i (on dit *incrémenter* i).

```
i=1
while i<=50 :
    print("Bonjour")
    i=i+1 .....
```

- Exemple 2 : la machine doit afficher tous les entiers de 1 à 100.

```
i=1
while i<=100 :
    print(i)
    i=i+1
```

- Exemple 3 : la machine doit afficher les carrés de tous les entiers de 1 à 100.

```
i=1
while i<=100 :
    print("Le carré de ",i," est ",i**2)
    i=i+1
```

2) Les autres types de boucles

- Une remarque pour finir : il existe deux autres types d'instructions de boucle :
 - Répéter jusqu'à
 - Pour.....allant de.....à

Lorsque l'on souhaite répéter un nombre **donné** de fois la même instruction ou le même bloc d'instructions, la commande `for` est la plus appropriée.

Admettons que l'on veuille afficher 5 fois le mot *blabla*. Voici ce que l'on peut faire.

```
>>> for i in range(5): # pour i allant de 0 à 4
...     print("blabla")
...
blabla
blabla
blabla
blabla
blabla
```

Bon, arrêtons là le blabla et regardons un exemple plus mathématique. Si on veut afficher les carrés des entiers de 1 à 7 :

```
>>> for i in range(1, 8): # pour i allant de 1 à 7
...     print(i**2)
...
1
4
9
16
25
36
49
```

- Par exemple, pour la boucle « Pour » :


```
for i in range(1,101) :
    print("Le carré de ",i," est ",i**2)
```

La boucle « Tant Que » est plus universelle que la boucle « Pour » car on n'a pas besoin de savoir au départ combien de répétitions la machine va faire.

Puis terminons sur un exemple classique qui est le calcul de la somme des premiers entiers. Disons ici que l'on s'arrête à 30. Autrement dit, on veut calculer

$$1 + 2 + 3 + \dots + 30$$

que l'on note aussi

$$\sum_{i=1}^{30} i$$

```
>>> S = 0
>>> for i in range(1, 31):    # pour i allant de 1 à 30
...     S = S + i
... print(S)
...
465
```

Exercice N°3 : A la recherche de la formule magique

Certains racontent que le jeune Carl Friedrich Gauss avait un jour été puni et devait calculer la somme de tous les entiers de 1 à 100 pendant sa récréation.

Le jeune homme, un peu plus doué en maths que la moyenne, a trouvé une formule et terminé le calcul rapidement, ce qui lui a permis de profiter de sa récréation.

1°) Écrivez un programme qui calcule la somme des entiers de 1 à n où n est choisi par l'utilisateur.

2°) Testez ce programme avec différentes valeurs de n et essayez de trouver la formule qu'aurait (re)trouvé Gauss.

Exercice N°4 : Un jeu

Écrivez et testez un programme qui :

- choisit un nombre entier au hasard ;
- demande au joueur de le trouver ;
- le joueur a le droit à autant d'essai qu'il le souhaite ;
- la machine dit à chaque essai si le nombre mystère est plus petit ou plus grand que le nombre choisi par le joueur.

On peut à la fin attribuer un score au joueur en fonction du nombre d'essai qu'il a réalisé.

Exercice N°5 : Problème du duc de Toscane

Le duc Cosme II de Médicis, protecteur de Galilée lui a un jour posé le problème suivant :

« J'ai remarqué que quand je lance trois dés, j'obtiens plus souvent la somme 10 que la somme 9, alors qu'il y a six façons d'obtenir l'un ou l'autre !».

1°) Quelles sont les six façons d'obtenir un 9 ? les six façons d'obtenir un 10 ?

2°) Écrivez un programme qui permet de vérifier l'affirmation du duc de Toscane.