



Unité 2 : Débuter la programmation en Python

Compétence 3 : La boucle non bornée While

Dans cette troisième leçon de l'unité 2, vous allez découvrir comment répéter un processus ou un ensemble d'instructions en utilisant une boucle non bornée **While**

Objectifs :

- Découvrir et mettre en œuvre la boucle non bornée **While**
- Utiliser la boucle **While** dans des exemples simples.

Il est parfois utile dans un programme de répéter une ou plusieurs instructions un nombre indéfini de fois. Si le nombre de répétition du processus n'est pas connu à l'avance, on utilise une boucle non bornée **While**.

La boucle est alors parcourue un nombre de fois jusqu'à ce qu'une condition ne soit plus vérifiée. Tant que cette condition est vérifiée, la boucle continue.

La syntaxe d'une boucle **While** est la suivante :

Langage naturel

Tant que condition faire

Instructions

Langage Python

`while` condition :
Instructions

```

ÉDITEUR : BOUCLE
Fonc Ctl Ops List Type E/S Modul
1:if ..
2:if .. else ..
3:if .. elif .. else
4:for i in range(taille):
5:for i in range(début,fin):
6:for i in range(début,fin,pas):
7:for i in liste:
8:while condition:
9:elif :
0:else:
Échap

```

Il n'existe pas d'instruction de fin de boucle. C'est l'indentation, c'est-à-dire le décalage vers la droite d'une ou plusieurs lignes, qui permet de marquer la fin de la boucle.

Mise en œuvre : Exemple n°1

Vous allez créer un script permettant de bien comprendre ce qu'est une boucle ainsi qu'un processus d'itération.

Mettre en œuvre l'algorithme ci-dessous :

L'algorithme a pour objet de déterminer le plus petit entier n tel que le terme général de la suite définie par $c_0 = 3,4$ et $c_{n+1} = 0,8 c_n$ est inférieur à 1.

$n \leftarrow 0$
 $c \leftarrow 3.4$
 Tant que $c \geq 1$
 $n \leftarrow n + 1$
 $c \leftarrow 0.8c$
 Fin Tant que

- Commencer un nouveau script et le nommer « WHILE »
- L'instruction **While** est accessible en tapant sur **F1** (Fns...) puis en choisissant le menu Ctl et enfin l'instruction **8 : while condition :**
- Tant que la variable c sera strictement supérieure au seuil la variable n sera incrémentée de 1.

```

ÉDITEUR : WHILE
LIGNE DU SCRIPT 0001
def a(c,seuil):_
  n=0
  while c>=seuil:
    n=n+1
    c=0.8*c
  return n

```





- Appuyer sur **F4** pour exécuter le script et observer l'affichage

```

PYTHON SHELL
>>>
>>> # Shell Reinitialized
>>> # L'exécution de WHILE
>>> from WHILE import *
>>> a(3.4,1)
6
>>> |
Fns... | a A # |Outils|Éditer|Script

```

Conseil à l'enseignant : Il peut s'avérer nécessaire de devoir conserver les valeurs, lorsque celles-ci correspondent à un calcul qui doit être par exemple réutilisé ou simplement conservé afin d'être comparé (suite numérique par exemple). Dans ce cas l'utilisation des listes est pratique.

Appliquons nos connaissances : Les rebonds du ballon :

Une balle est lâchée d'une hauteur de 1,20 m et rebondit sur le sol des $\frac{3}{5}$ de la hauteur du rebond précédent.

Vous devez élaborer un algorithme donnant le nombre de rebonds au bout duquel la hauteur atteinte par la balle est strictement inférieure à 1 cm.

- Définir une variable H dans laquelle on stockera la hauteur du rebond exprimée en cm et R une variable pour compter les rebonds
- On devra répéter plusieurs fois l'instruction H prend la valeur $\frac{3}{5} \times H$ sans connaître à l'avance le nombre de répétitions.
- On testera donc si $H > 1$ et le traitement dans la boucle sera réalisé tant que cette condition restera vérifiée.



Algorithme

$H \leftarrow 90$

$R \leftarrow 0$

Tant que $H \geq 1$

$H \leftarrow \frac{3}{5} \times H$

$R \leftarrow R + 1$

Fin Tant que

- Créer un nouveau script et le nommer rebonds
- On propose d'utiliser une fonction. Ainsi on veillera à respecter les indentations (une pour l'instruction **While** et une seconde pour l'instruction **Return** permettant de renvoyer le contenu de la variable R)
- On peut également comme dans l'exemple n°1 créer une liste puis afficher les hauteurs successives des rebonds.

```

ÉDITEUR : REBONDS
LIGNE DE SCRIPT 0001
def rebonds():_
    H=120
    R=0
    while H>=1:
        H=H*3/5
        R=R+1
    return (R)

```





10 Minutes de Code

TI-83 PREMIUM CE & TI - PYTHON

- Appuyer sur **F3** (Outils) et choisir le menu **2 : Indent** ← afin de fermer la boucle **While**, tout en conservant l'exécution de la fonction.
- Pour exécuter le script, appuyer sur **F4** (Exec)
- Entrer le nom de la fonction « rebonds() » et valider en appuyant sur la touche
- Enrichir éventuellement le script à l'aide d'un message, par exemple :

`return` (« Nombre de rebonds = »,R)

```
PYTHON SHELL
>>>
>>> # Shell Reinitialized
>>> # L'exécution de REBONDS
>>> from REBONDS import *
>>> rebonds()
10
>>> |
Fns... | a A # |Outils|Éditer|Script
```

Un défi : calculer la distance totale parcourue par la balle jusqu'à son immobilisation. (On supposera que les rebonds de la balle sont rigoureusement situés sur la verticale)

