

Les boucles while

1 Syntaxe de la commande

Une boucle `while` permet de réaliser la répétition d'un bloc d'instructions sous réserve qu'une condition est vérifiée. La syntaxe de cette commande est la suivante :

```
1 while condition :  
2     instructions
```

Détaillons les éléments de cette syntaxe.

- *condition* : représente une expression dont l'évaluation est soit vraie (`True`) soit fausse (`False`).
- *instruction* : désigne un bloc d'instructions qui va être exécuté, de manière successive, **tant que** *condition* est vraie.

Terminaison : dans le cas d'une boucle `while`, l'exécution se termine dès que *condition* n'est plus vérifiée. Si *condition* est toujours vérifiée (comme `0==0`) le programme admet une exécution infinie. On parle de **boucle infinie**.

2 Ordre d'exécution

Afin de bien comprendre l'ordre d'exécution d'une boucle `while`, on va l'illustrer à l'aide d'un exemple consistant à trouver, dans une matrice ligne, le premier élément vérifiant une condition donnée.

Commençons par générer une matrice ligne de valeurs aléatoires i.e. une matrice ligne dont les coefficients sont choisis aléatoirement. Pour cela, on doit charger une nouvelle bibliothèque. On utilisera l'importation :

```
import numpy.random as rd
```

Une fois la bibliothèque chargée, on utilisera la commande `rd.random()`.

- ▷ Tester dans la console cette commande plusieurs fois. Qu'observez-vous ?

- ▷ Tester dans la console la commande `rd.random(10)`. Que fait-elle ?

Dans un nouveau script, taper `r=rd.random(10)`. On vient donc de créer une matrice ligne de taille 10 appelée `r` dont les coefficients sont des nombres aléatoires compris entre 0 et 1. On va alors définir une fonction qui prend en entrée un réel `t` compris entre 0 et 1 et qui renvoie la valeur du 1er coefficient de `r` plus grand que `t`.

```
1 def test_grandeur(t):  
2     i=0  
3     while r[i] < t:  
4         i=i+1  
5     return r[i]
```

- ▷ Recopier la fonction ci-dessus dans le script. Testez-là pour différentes valeurs de $t \in [0, 1]$. Que constatez-vous ?

- ▷ Tester votre fonction pour des valeurs très proches de 1. Un message d'erreur apparaît-il ?

Pour comprendre ce message d'erreur, étudions l'exécution de la boucle `while` correspondante.

- La variable `i` est initialement affectée à la valeur 0.
- Puis, on teste si les éléments successifs du vecteur `r` sont strictement plus petits qu'une certaine valeur de `t` (0.999 par exemple).
- Si aucune valeur de `r` n'est plus petite que 0.999. A la fin `i` prend la valeur $9+1=10$ et on obtient un message d'erreur car il n'y a d'éléments `r[10]` puisque `r` n'a que 10 éléments.

Il faut donc modifier la fonction de sorte à s'arrêter quand `i` prend une valeur plus grand que le nombre d'éléments de `r`. Pour ce faire, il suffit de vérifier que `i` ne dépasse pas le nombre de colonnes de la matrice ligne avant de réaliser le test `r[i] < t`.

Pour accéder au nombre de colonnes de la matrice ligne `r`, on utilise la commande `np.shape(r)[0]`.

```

1 taille_r=np.shape(r)[0]
2
3 def test_grandeur_bis(t):
4     i=0
5     while i<taille_r and r[i] < t:
6         i=i+1
7     if i>=taille_r:
8         print('Pas d'élément plus grand que', t)
9     else :
10        return r[i]
```

- ▷ Recopier cette fonction dans le script et testez-la.

La modification a bien empêché `i` de prendre des valeurs plus grandes que le nombre d'éléments de `r`. Ceci est notamment dû au mécanisme d'évaluation de l'opérateur `and`. À chaque tour de boucle, la condition `i < taille_r and r[i] < t` est testée. L'évaluation de ce test se déroule comme suit.

Tout d'abord, Python teste si `i < taille_r`.

1. Si c'est le cas, `r[i] < t` est alors testé.
 - (a) Si c'est le cas, la condition est vérifiée et les instructions de la boucle sont exécutées.
 - (b) Sinon, la condition n'est pas réalisée et on sort de la boucle.
2. Sinon, la condition n'est pas réalisée et on sort de la boucle.

3 A vous de jouer

Exercice 3.1 On considère la suite définie par $u_0 = 4$ et $\forall n \in \mathbb{N}, u_{n+1} = u_n^2 + 1$.

1. Ecrire une fonction qui prend en entrée un réel x et qui renvoie la valeur du premier terme de la suite qui soit supérieur ou égal à x .
2. Modifier votre fonction pour qu'elle renvoie également l'indice de ce terme.

Exercice 3.2 On considère la suite définie par $u_0 = \frac{3}{4}$ et $\forall n \in \mathbb{N}, u_{n+1} = u_n - u_n^2$. On admet que la suite $(u_n)_n$ est décroissante et converge vers 0. Ecrire une fonction, prenant en entrée une valeur `eps` et qui permet de déterminer le premier entier n pour lequel $u_n < \text{eps}$.