

DEV1 – DEV1L – Laboratoires Python**TD 07 – Boucles While**

Nous allons à présent nous initier à l'écriture de boucles while en Python de façon à exécuter un bloc de code tant qu'une condition est satisfaite.

Table des matières

1	Boucles while	2
2	Tester les entrées de l'utilisateur	2
3	Exercices récapitulatifs	4
4	En résumé ...	5

Rappel

Vous devez absolument avoir fait et compris les td précédents *avant* d'aborder celui-ci.

1 Boucles while

Le script Python ci-dessus affiche les nombres de 1 à 100 au moyen d'une boucle `while`.

```
1
2 compteur = 1
3
4 while(compteur <= 100):
5     print(compteur)
6     compteur = compteur + 1
7
8 print("Terminé !")
```

L'instruction `while` permet donc d'exécuter un bloc de code tant qu'une condition (celle qui suit le mot clé `while`) est satisfaite.

Exercice 1 Boucles while

Modifiez le script ci-dessus de façon à ce que celui-ci :

1. demande un nombre entier n à l'utilisateur,
2. affiche tous les nombres entiers de 1 à n à l'écran (vous pouvez supposer que $n > 1$).

Exercice 2 Suites d'entiers

Écrivez un script Python qui demande à l'utilisateur un nombre entier n et affiche :

- ▷ les nombres pairs qui sont compris entre 1 et n ;
- ▷ les nombres de $-n$ à n ;
- ▷ les multiples de 5 qui sont compris entre 1 et n ;
- ▷ les multiples de n compris entre 1 et 100.

2 Tester les entrées de l'utilisateur

Une utilisation classique de la boucle `while` est d'exécuter un morceau de code tant que les données entrées par l'utilisateur ne satisfont pas une certaine condition. Voici un exemple : le script suivant calcule la racine carrée d'un nombre entré par l'utilisateur, en s'assurant auparavant que le nombre entré est bien positif.

```
1
2 import math
3
4 nombre = int(input("Veuillez entrer un nombre positif: "))
5
6 while(nombre < 0):
7     nombre = int(input("Ce nombre n'est pas positif. Veuillez réessayer: "))
8
9 print("La racine carrée de", nombre, "est", math.sqrt(nombre))
```

Exercice 3 Le nom du mois

Écrivez un script Python qui demande un nombre entier compris entre 1 et 12 à l'utilisateur et affiche le nom du mois correspondant (1 = «Janvier», 2 = «Février», ..., 12 = «Décembre»). Tant que le nombre entré n'est pas compris entre 1 et 12, le programme affiche un message d'erreur et en demande un autre à l'utilisateur.

Voici un exemple de petit jeu : on demande à l'utilisateur de deviner un nombre secret entre 1 et 10 et on boucle tant que l'utilisateur ne l'a pas trouvé.

```
1 nombre_secret = 6
2
3
4 print("L'ordinateur a choisi un nombre entre 1 et 10")
5 print(" Vous allez devoir le deviner !")
6
7 nombre_utilisateur = int(input("Entrez un nombre:"))
8
9 while(nombre_utilisateur != nombre_secret):
10     nombre_utilisateur = int(input("Raté ! Essayez encore: "))
11
12 print("Bravo !")
```

Évidemment, le script ci-dessus a peu d'intérêt vu que le nombre à deviner est toujours égal à 6. À vous de le rendre plus intéressant :

Exercice 4 Le nombre secret

Modifiez le script ci-dessus de façon à ce que l'ordinateur choisisse réellement un nombre au hasard entre 1 et 10 à deviner (Rappel : nous avons vu dans le TD03 comment générer des nombres au hasard).

Exercice 5 Le nombre secret (suite)

Modifiez votre script sur le nombre secret de façon à compter et afficher le nombre d'essais que l'utilisateur a du faire pour réussir à trouver le nombre. Voici un exemple d'exécution possible du script (l'utilisateur trouve le nombre en 5 essais) :

```
L'ordinateur a choisi un nombre entre 1 et 10
Vous allez devoir le deviner !
Entrez un nombre: 1
Raté ! Essayez encore: 9
Raté ! Essayez encore: 5
Raté ! Essayez encore: 7
Raté ! Essayez encore: 8
Bravo ! Vous avez trouvé le nombre secret en 5 essais.
```

Voici encore un exemple de boucle dont la condition dépend des entrées au clavier : on lit des nombres positifs au clavier tant que l'utilisateur ne rentre pas une valeur négative. Concrètement, le nombre -1 est utilisé comme *valeur sentinelle* : on continue de boucler tant que le programme ne détecte pas de valeur inférieure ou égale à -1 .

Le corps de la boucle peut alors traiter le nombre entré au clavier (le traitement dépend de ce qu'on souhaite faire de la suite de nombres en question).

```

1 # Canevas d'un script lisant une suite de nombres au clavier
2 nb_valeurs = 0
3
4 valeur = int(input("Entrez une suite de nombre entier positifs, terminez par -1: "))
5
6 while(valeur > -1):
7     nb_valeurs = nb_valeurs + 1
8     # Faire quelque chose avec le contenu de la variable "valeur" ici
9     valeur = int(input("Entrez une autre valeur (ou -1 pour terminer): "))
10
11 print("Vous avez entré",nb_valeurs,"valeurs")

```

Exercice 6 La somme

Adaptez le canevas ci-dessus de façon à lire une suite de nombre terminée par -1 et à afficher la somme à l'écran. Par exemple, si l'utilisateur entre : 2 6 3 1 2 -1 le script affiche La somme vaut 14.

Exercice 7 La moyenne

Modifiez le script précédent de façon à ce que le programme affiche la moyenne (et pas la somme) de tous les nombres entrés. De plus, si l'utilisateur ne rentre aucun nombre positif, le programme affiche un message d'erreur.

3 Exercices récapitulatifs

Exercice 8 Les nombres au carré

Écrivez un script qui demande un nombre positif n à l'utilisateur et qui affiche tous les nombres au carré plus petit ou égal à n .

Exemple : Si l'utilisateur entre 30, le script affiche : 1 ($= 1 * 1$), 4 ($= 2 * 2$), 9 ($= 3 * 3$), 16 ($= 4 * 4$) et 25 ($= 5 * 5$).

Exercice 9 Le mot

Écrivez un script qui demande un mot à l'utilisateur. Un mot est une chaîne de caractère ne contenant aucun espace. Tant que la chaîne entrée par l'utilisateur contient un espace, le script affiche un message d'erreur et demande un autre mot à l'utilisateur. (Relisez le TD4 pour savoir comment tester qu'une chaîne contient un espace).

Exercice 10 Le nombre secret (suite et fin)

Modifiez à nouveau votre script sur le nombre secret de façon à aider le joueur : en fonction du nombre erroné entré par l'utilisateur, le programme affiche «trop petit» ou «trop grand».

Voici un exemple d'exécution possible du script :

```

L'ordinateur a choisi un nombre entre 1 et 10
  Vous allez devoir le deviner !
Entrez un nombre: 1
Trop petit ! Essayez encore: 9
Trop grand ! Essayez encore: 6
Trop grand ! Essayez encore: 5
Bravo ! Vous avez trouvé le nombre secret en 4 essais.

```

Exercice 11 Maximum et minimum

Écrivez un script qui demande à l'utilisateur une série de nombres positifs qui se termine par -1 (valeur sentinelle). Le programme affiche le maximum et le minimum de la série.

Par exemple, si l'utilisateur entre : 5 9 2 5 12 7 4 3 -1
le programme affiche

maximum : 12

minimum: 2

4 En résumé ...

Principaux points de matière du TD

Voici les principaux points abordés lors de ce TD. Vous devez absolument être à l'aise avec ceux-ci avant d'aborder la prochaine séance d'exercice.

1. L'instruction `while` en Python.
2. L'utilisation de la boucle `while` pour vérifier les entrées de l'utilisateur.
3. Mettre en pratique la boucle `while` pour écrire des scripts résolvant des problèmes divers.