

La boucle bornée for

Plan du T.P.

1	Les boucles	49
2	La boucle for ... in ... en Python	50
2.1	Manipuler	50
2.2	Verbaliser	50
2.3	Abstraire	50
3	Application à la résolution de problèmes	51
4	Déterminer si un nombre est premier	52

[Retour au sommaire](#)

1 Les boucles

L'informatique nous offre la possibilité de répéter des tâches de façon automatique. Nous avons déjà vu que pour éviter de répéter les mêmes instructions plusieurs fois, nous pouvions avoir recours à des fonctions ou à des boucles.

Exercice 1 :

En une phrase, comparer les effets des deux programmes suivants :

```
1 x = 0
2 x = x + 1
3 x = x + 1
4 x = x + 1
5 x = x + 1
6 x = x + 1
```

```
1 x = 0
2 while x < 5 :
3     x = x + 1
```

Deux types de situations peuvent se présenter :

- le nombre de répétitions à réaliser n'est pas connu à l'avance.
Exemple 1 : un jeu consiste à lancer un dé jusqu'à obtenir un six.
- le nombre de répétitions est connu à l'avance.
Exemple 2 : un jeu consiste à lancer 3 fois de suite un dé.

Exercice 2 :

Parmi les deux algorithmes suivants, lequel associer à l'exemple 1 ? Et à l'exemple 2 ?

```
• Tant que lancer ≠ 6 :
    lancer le dé
```

```
• Pour compteur allant de 1 à 3 :
    lancer le dé
```

Remarque :

la variable compteur ci-dessus joue, comme son nom l'indique, le rôle d'un compteur. Elle compte les "tours de boucle" réalisés.

2 La boucle for ... in ... en Python

2.1 Manipuler

En anglais, **range** signifie ampleur, gamme, envergure, étendue. Même si cette traduction n'est pas correcte, nous pourrions penser à un intervalle.

Testons dans la console les deux lignes suivantes :

```
1 >>> for compteur in range(1, 10) :  
2 ...     print(compteur)
```

Exercice 3 :

En utilisant une boucle for, écrire dans la console des instructions permettant d'afficher :

- les entiers de 1 à 10,
- les entiers de 1 à 100,
- les nombres pairs de 0 à 40.

Exercice 4 :

Écrire dans l'éditeur une fonction `jusque(n)`, prenant un nombre entier n en argument et affichant successivement tous les entiers de 0 à n .

2.2 Verbaliser

Exercice 5 :

- Par quelles valeurs passe successivement la variable `compteur` dans l'exemple ci-dessus ?
- Compléter : "pour aller de 1 à n ", nous écrivons : `range(..... ,)`

2.3 Abstraire

Syntaxe d'une boucle for :

Langage naturel	Python
Pour variable allant de 1 à n bloc d'instructions	<code>for</code> variable in <code>range(1, n+1)</code> : bloc d'instructions

Remarques :

- La variable prend successivement les valeurs de 1 à n .
- Nous remarquons que pour aller jusqu'à n , nous écrivons $n + 1$: `range(1, n+1)`.
Nous pourrions penser à un intervalle fermé à gauche, et **ouvert à droite** : $[1 ; n + 1[$.
- Les **deux points** en fin de première ligne introduisent le bloc d'instructions à répéter.
Celui-ci doit être **indenté**.
- La fin de l'indentation signale la fin du bloc d'instruction à répéter.

3 Application à la résolution de problèmes

Nous allons maintenant utiliser des boucles `for` à l'intérieur de nos fonctions. La fonction `print` que nous venons d'utiliser afin de manipuler la boucle `for` offre juste un affichage, mais ne permet pas de renvoyer un résultat que nous pourrions ré-exploiter.

Nous utiliserons à cette fin des **fonctions**, qui renvoient un résultat au moyen de l'instruction `return` :

Syntaxe d'une fonction :

```
1 def nom_de_la_fonction(argument1, argument2, ...) :  
2     instructions  
3     return resultat
```

Exercice 6 : permis de conduire

Afin de passer son permis de conduire dans quelques années, une élève place la somme de 400€. Celle-ci est rémunérée à l'issue de chaque année au taux de 3% (pourcentage calculé sur le montant présent sur le compte pour l'année en cours).

Écrire une fonction `permis(n)` :

- prenant en argument un nombre d'années n ,
- et renvoyant la somme présente sur le compte au bout de n années de placement.

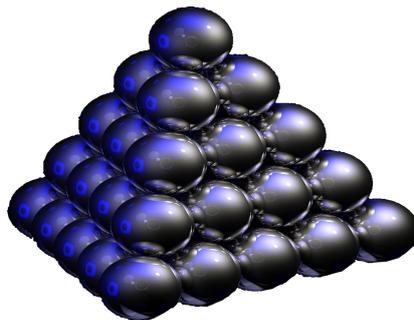
Exercice 7 : le marchand de farine

Au XVIII^{ème} siècle, un négociant en farine vivant à Sète décide de se rendre à Toulouse, pour y vendre sa farine. Il emprunte pour son périple le canal du Midi, qui relie la mer Méditerranée et la Garonne. Le canal traverse 63 écluses. A chaque écluse, le négociant doit donner 1 % de son chargement en péage royal, puis échanger 3 sacs de farine contre de la nourriture.

- Écrire une fonction `farine(sac)`, prenant en argument le nombre de sacs dont il dispose au départ de Sète, et renvoyant le nombre de sacs qu'il lui restera à son arrivée à Toulouse.
- Écrire une seconde fonction, `minimum()`, renvoyant le nombre minimum de sacs dont il faut disposer au départ pour arriver jusqu'à Sète avec au moins 500 sacs à vendre.
- N'y-a-t-il pas un anachronisme ⁽¹⁾ dans cet exercice ?

Exercice 8 : un problème d'empilement de sphères

On empile des sphères comme sur la figure ci-dessous pour former une pyramide à base carrée.



- Combien de sphères faut-il pour réaliser une pyramide à 2 étages ? 3 étages ?

(1). <https://fr.wikipedia.org/wiki/Anachronisme>

- b. Sachant qu'il faut 55 sphères pour former une pyramide à 5 étages, combien en faudra-t-il pour former une pyramide à 6 étages ?
- c. Écrire une fonction `pyramide(n)`, prenant en argument un entier naturel n , et renvoyant le nombre de sphères nécessaires pour réaliser une pyramide à n étages.
- d. En déduire le nombre d'étages de la plus haute pyramide que l'on peut réaliser avec 1000 sphères.

4 Déterminer si un nombre est premier



Définition :

Un nombre entier naturel est **premier** s'il admet exactement deux diviseurs positifs : 1 et lui-même.

Exemples :

- 1 n'est pas premier : il n'admet qu'un diviseur positif : lui-même,
- 7 est premier : il admet deux diviseurs positifs : 1 et lui-même,
- 6 n'est pas premier : il admet quatre diviseurs positifs : 1, 2, 3, 6.

Nombre	Diviseurs
6	6·3·2·1
7	7·1
8	8·4·2·1
9	9·3·1

Exercice 9 : une fonction pour déterminer si un nombre est premier

- a. Copier-coller la fonction `divisible(a, b)`, écrite dans le dernier exercice du chapitre 6.
- b. Compléter les fonctions ci-dessous :

- Fonction `divisible(a, b)` :
 Si :
 Renvoyer
- Sinon :
 Renvoyer
- Fonction `premier(n)` :
 `reponse` ← Vrai
 Pour compteur allant de ... à ... :
 Si :

 Renvoyer `reponse`

- c. Écrire une fonction `premier(n)` :
 - prenant en argument un entier naturel $n \geq 2$,
 - renvoyant :
 - `True` si n est premier,
 - `False` sinon.