Lister les diviseurs d'un nombre

Niveau 1 : sans liste

Nous allons écrire un programme, sous la forme d'une fonction `diviseurs(n)`, qui va afficher à l'écran les diviseurs positifs d'un nombre naturel entré par l'utilisateur.

1. Comment savoir si un entier est un diviseur d'un autre entier ?

   Un entier b est le diviseur d'un entier a si, dans la division euclidienne de a par b, le reste est égal à 0.

2. Quels nombres doit-on tester pour espérer obtenir la liste complète des diviseurs d'un nombre ?

   On teste tous les entiers naturels de 1 jusqu'à n.

3. Quelle instruction permet de passer en revue une série de nombres entiers dont on connaît le début et la fin ? Quelle ligne de code utilisera-t-on ?

   Nous allons utiliser une boucle `for` avec la ligne `for i in range(1,n+1):`.

4. Le symbole % permet d'obtenir le reste de la division euclidienne d'un nombre par un autre. A l'aide des réponses précédentes, compléter le script suivant :

```
from math import *
def diviseurs(n):
  for i in range( ... , ...):
    if n%i==0:
       print( ... )
```

   Pour écrire plus vite, on peut retrouver dans la boîte à outils, accessible avec la touche représentant une boîte, les instructions basiques (boucles, définition d'une fonction, etc.) et dans le menu variable (VAR), toutes les variables déjà définies dans le script. Cela évite d'avoir à tout réécrire lettre par lettre !

```
from math import *
def diviseurs(n):
  for i in range(1,n+1):
    if n%i==0:
       print(i)
```

Niveau 2 : en utilisant les listes

Notion de liste

En Python, on distingue plusieurs types de variables dont les principaux sont : nombres entiers, nombres décimaux et chaînes de caractères (texte). Un autre type de ces variables est la liste qui, comme son nom l'indique, permet de stocker une liste de… variables. En effet, une liste peut comprendre des nombres, des chaînes de caractères, d'autres listes, et même tout cela à la fois.
Une liste se définit à l'aide de crochets et ses éléments sont séparés par des virgules :

```
zoo = ["lion","tigre","ours"]
```

```
temperatures = [27.4,27.7,24.5,21]
```

Nous allons nous intéresser au cas le plus simple puisque nous listons des entiers.

Lister les diviseurs d'un nombre

Dans notre cas, nous allons construire la liste pas à pas puisque son contenu va dépendre du nombre n qui sera choisi au lancement du programme.
Néanmoins, même si cette liste n'existe pas encore, il est nécessaire de la déclarer dès le début de notre fonction.
Nous allons donc créer un liste vide appelée div avec la ligne :

```
div = []
```

Pour ajouter un nombre k à notre liste, on utilisera `div.append(k)` et dans notre cas, le nombre ajouté doit être un diviseur du nombre n choisi au départ.
On propose de modifier légèrement la fonction précédente, en complétant le script suivant :

```
from math import *
def diviseurs(n):
  div= ...
  for i in range( ... , ...):
    if n%i==0:
       div.append( ... )
  return div
```

```
from math import *
def diviseurs(n):
  div=[]
  for i in range(1,n+1):
```

```
    if n%i==0:
        div.append(i)
  return div
```

Niveau expert : listes en compréhension

Il est aussi possible de définir une liste en indiquant entre crochets toutes les conditions nécessaires à sa génération. Par exemple, pour générer la liste des multiples de 7 compris entre 100 et 200, on pourra utiliser :

```
return [i for i in range(100,201) if i%7==0]
```

Essayez d'écrire une fonction `diviseurs_short(n)` qui permet de lister les diviseurs de n en seulement deux lignes !

```
def diviseurs_short(n):
  return [i for i in range(1,n+1) if n%i==0]
```