

Multiples et nombres premiers

Compétences visées

- **chercher**, expérimenter – en particulier à l'aide d'outils logiciels ;
- **modéliser**, faire une simulation, valider ou invalider un modèle ;
- **représenter**, choisir un cadre (numérique, algébrique, géométrique...), changer de registre ;
- **calculer**, appliquer des techniques et mettre en œuvre des algorithmes.

Ces compétences sont mises en œuvre dans le cadre de l'extrait du programme de 2^{nde} GT ci-dessous :

- « Pour des entiers a et b donnés, déterminer le plus grand multiple de a inférieur ou égal à b . »
- « Déterminer si un entier naturel est premier. »

Situation déclenchante

Le crible d'Eratosthène

Ce crible, qui permet de trouver tous les entiers premiers jusqu'à un entier spécifié.

	2	3	×	5	×	7	×	×	×
11	×	13	×	×	×	17	×	19	×
×	×	23	×	×	×	×	×	29	×
31	×	×	×	×	×	37	×	×	×
41	×	43	×	×	×	47	×	×	×
×	×	53	×	×	×	×	×	59	×
61	×	×	×	×	×	67	×	×	×
71	×	73	×	×	×	×	×	79	×
×	×	83	×	×	×	×	×	89	×
×	×	×	×	×	×	97	×	×	×

Voici le principe du crible :

- 1) Écrire tous les entiers de 2 à n .
- 2) Enlever (ou barrer) les multiples de 2 sauf 2.
- 3) Récupérer le plus petit nombre non barré, c'est à dire 3, et barrer les multiples de 3 sauf 3, etc...
- 4) Test d'arrêt : On s'arrête dès qu'on a dépassé la racine carrée de n .
- 5) Les nombres restants sont les nombres premiers inférieurs ou égaux à n .

Problématique

Écrire un script qui simule le fonctionnement du crible d'Eratosthène et qui permet de trouver les nombres premiers inférieurs ou égaux à un entier spécifié.

Fiche méthode

Proposition de résolution

On crée trois fonctions dans ce script:

- Une fonction **listemulti** qui prend comme arguments deux entiers naturels a et n et qui renvoie la liste des multiples de a inférieurs ou égaux à n .
- Une fonction **derniermulti** qui prend comme arguments deux entiers naturels a et n et qui renvoie le plus grand multiple de a inférieur ou égal à n .
- Une fonction **eratosthene** qui prend comme argument un entier naturel n et qui renvoie la liste des entiers premiers inférieurs ou égaux à n en utilisant le principe du crible d'Eratosthène.

```
PYTHON SHELL
>>> listemulti(2,13)
[0, 2, 4, 6, 8, 10, 12]
>>> listemulti(3,30)
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
>>> |
```

```
PYTHON SHELL
>>> derniermulti(2,13)
12
>>> derniermulti(3,30)
30
>>> |
```

```
PYTHON SHELL
>>> eratosthene(20)
[2, 3, 5, 7, 11, 13, 17, 19]
>>> eratosthene(50)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
>>> |
```

Etapas de résolution

L'instruction **from math import *** permet d'ajouter les fonctions de la bibliothèque `math` (on aura besoin de la fonction racine carré dans le programme suivant)

Fonction **listemulti** qui prend comme arguments deux entiers naturels a et n et qui renvoie la liste des multiples de a inférieurs ou égaux à n .

L'instruction `l=[]` permet d'initialiser la liste `l` avec la liste vide. L'instruction `l.append(a*i)` permet de rajouter `a*i` à la liste `l`.

```
ÉDITEUR : MULTIPLE
LIGNE DU SCRIPT 0001
from math import *

def listemulti(a,n):
    l=[]
    i=0
    while a*i<=n:
        l.append(a*i)
        i=i+1
    return l
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus



Fiche méthode

Etapes de résolution

C'est un principe à retenir : On peut appeler une fonction (ici : la fonction **listemulti**) à l'intérieur d'une autre fonction (ici: **derniermulti**). L'utilisation successive de fonctions en python rend le script dans son ensemble plus lisible.

La fonction **derniermulti** qui prend comme arguments deux entiers naturels a et n et qui renvoie le plus grand multiple de a inférieur ou égal à n.

```
EDITEUR : MULTIPLE
LIGNE DU SCRIPT 0022
def derniermulti(a,n):
    l=listemulti(a,n)
    a=len(l)-1
    return l[a]
```

La fonction **eratosthene** qui prend comme argument un entier naturel n et qui renvoie la liste des entiers premiers inférieurs ou égaux à n.

On va créer une liste nommée multiple dans laquelle on va stocker tous les multiples des premiers entiers inférieurs ou égaux à la partie entière de \sqrt{n} . Pour cela, on va faire appel à la l'instruction **listemulti(i,n)** définie précédemment. Il faudra cependant enlever les 2 premiers éléments de la liste (instruction `del l[0]` suivi de `del l[0]`).

L'instruction `multiple=multiple+1` permet de fusionner les deux listes.

```
EDITEUR : MULTIPLE
LIGNE DU SCRIPT 0033
def eratosthene(n):
    a=floor(sqrt(n))
    multiple=[]
    premier=[]
    for i in range (2,a+1):
        l= listemulti(i,n)
        del l[0]
        del l[0]
        multiple=multiple+l
    for i in range (2,n+1):
        if i not in multiple:
            premier.append(i)
    return premier
```

Une fois la liste de multiples créée, d'après le crible d'Eratosthène, les entiers premiers sont ceux qui ne font pas partie de la liste. On détermine donc à l'aide d'une boucle les éléments non présents dans la liste et on les ajoute à la liste des nombres premiers. L'instruction `premier.append(i)` permet d'ajouter l'entier i à la liste premier.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

