



Objectifs du module Algorithmique Appliquée

► Idée

L'idée de cette *fiche/activité* est de présenter, et ce dès le début de l'année, ce qui sera possible de faire au fur et à mesure des apprentissages sur l'algorithmique, et notamment en python.

Il n'est pas (forcément) question de comprendre dès maintenant les scripts (tant par leur contenu mathématique qu'algorithmique) suivants, mais ils serviront de référence lors des chapitres à venir !

► Cadre

On va prendre comme cadre les nombres premiers (qui seront étudiés en partie maths) et plus précisément :

- on va déterminer les diviseurs d'un entier ;
- on va tester si un entier est premier ;
- on va déterminer les nombres premiers inférieurs à un entier donné ;
- on va créer une petite Interface Homme-Machine (IHM) pour interagir avec un utilisateur.

► Algorithme 1 (fonction)

On commence donc par une fonction qui va *déterminer* les diviseurs d'un entier donné en paramètre :

Fonction renvoyant la liste des diviseurs d'un entier n

```
1 def listediviseurs(n) :
2     res = []
3     for i in range(1, n+1) :
4         if n % i == 0 :
5             res.append(i)
6     return res
```



✚ Début de la console Python ✚

```
>>> listediviseurs(14)
[1, 2, 7, 14]
>>> listediviseurs(17)
[1, 17]
>>> listediviseurs(50)
[1, 2, 5, 10, 25, 50]
```

✚ Fin de la console Python ✚

► Algorithme 2 (fonction booléenne)

On passe ensuite à une fonction booléenne qui renvoie Vrai ou Faux suivant la primalité d'un entier donné en paramètre :

Fonction booléenne testant si un entier n est premier

```
1 def estpremier(n) :
2     nbdiv = len(listediviseurs(n))
3     if nbdiv == 2 :
4         return True
5     else :
6         return False
```

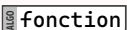


✚ Début de la console Python ✚

```
>>> estpremier(14)
False
>>> estpremier(17)
True
>>> estpremier(50)
False
```

✚ Fin de la console Python ✚

► Algorithme 3 (fonction)

C'est maintenant au tour d'une  renvoyant les nombres premiers inférieurs à un entier donné en .

Fonction renvoyant la liste des nombres premiers inférieurs à un entier n

```
1 def listenombrepremiers(n) :
2     res = []
3     for i in range(1, n+1) :
4         if estpremier(i) == True :
5             res.append(i)
6     return res
```

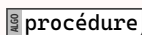


♣ Début de la console Python ♣

```
>>> listenombrepremiers(14)
[2, 3, 5, 7, 11, 13]
>>> listenombrepremiers(17)
[2, 3, 5, 7, 11, 13, 17]
>>> listenombrepremiers(50)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

♣ Fin de la console Python ♣

► Algorithme 4 (procédure)

On peut maintenant passer à la partie  qui permet une interaction (IHM) avec l'utilisateur.

Procédure d'exécution (IHM)

```
1 def exec() :
2     choix = "o"
3     while choix == "o" :
4         n = -1
5         while n <= 1 :
6             n = int(input("Saisir un entier n, supérieur à 2 : "))
7         if estpremier(n) == True :
8             print(f"{n} est premier.")
9         else :
10            print(f"{n} n'est pas premier.")
11            listeres = listenombrepremiers(n)
12            print(f"La liste des entiers premiers <= à {n} est {listeres}.")
13            print(f"Il y a donc {len(listeres)} entiers premiers <= à {n}.")
14            choix = input("Recommencer [o/n] ? ")
```

♣ Début de la console Python ♣

```
>>> exec()
Saisir un entier n, supérieur à 2 : 14
14 n'est pas premier.
La liste des entiers premiers <= à 14 est [2, 3, 5, 7, 11, 13].
Il y a donc 6 entiers premiers <= à 14.
Recommencer [o/n] ? o
Saisir un entier n, supérieur à 2 : 1
Saisir un entier n, supérieur à 2 : -3
Saisir un entier n, supérieur à 2 : 25
25 n'est pas premier.
La liste des entiers premiers <= à 25 est [2, 3, 5, 7, 11, 13, 17, 19, 23].
Il y a donc 9 entiers premiers <= à 25.
Recommencer [o/n] ? n
```

♣ Fin de la console Python ♣

