



Cours

Définition : On rappelle qu'un entier n est premier s'il admet exactement **deux** diviseurs. Ces diviseurs sont alors 1 et n lui-même.

Par exemple, les nombres 2, 3, 5 et 7 sont premiers.



Un peu d'histoire

La première trace incontestable de la présentation des nombres premiers remonte à l'Antiquité et se trouve dans les *Éléments* d'Euclide.

Il y donne une définition des nombres premiers, la preuve

de leur infinité, la définition du plus grand commun diviseur (pgcd) et du plus petit commun multiple (ppcm), et les algorithmes pour les déterminer, aujourd'hui appelés algorithmes d'Euclide.



Exercice 1 : Pour commencer

1. Expliquer pourquoi 789 n'est pas premier.
2. Pour déterminer les diviseurs d'un entier n , on doit tester si chacun des entiers de 1 à n divise n . Recopier et compléter le code de la fonction `diviseurs`.

```

1 def diviseurs(n:int) -> list:
2     """ Renvoie la liste des diviseurs de n. """
3     res = []
4     for ... : # i varie de 1 à n
5         if ... : # Teste si i divise n
6             res.append(...) # Ajoute i à la liste res
7     return res

```

3. Écrire une fonction `est_premier(n:int) -> boolean` qui teste si un entier n passé en paramètre est premier ou non. Par exemple, `est_premier(789)` renvoie **False**.
4. a. Que renvoie l'instruction ci-dessous ?


```
[k for k in range(1, 21) if est_premier(k)]
```
- b. Quelle instruction doit-on écrire pour connaître le nombre de nombres premiers à quatre chiffres ?
5. a. Écrire une fonction `possede_un_jumeau(n:int) -> boolean` qui vérifie si le nombre entier n est un nombre premier qui possède un jumeau. Par exemple :
 - ☐ `possede_un_jumeau(3)` renvoie **True**
 - ☐ `possede_un_jumeau(4)` renvoie **False**
 - ☐ `possede_un_jumeau(5)` renvoie **True**
- b. Quel est le plus grand nombre premier compris entre 2000 et 2999 qui possède un jumeau ?

