

Ex1. a) Recopier le script (**pour**) ci-dessous et teste le.

```
deg PYTHON
1 from math import *
2 for k in range(10):
3     print(k)
```

```
>>> from pour import *
0
1
2
3
4
5
6
7
8
9
```

k varie de 0 à 9

b) Modifie le script précédent pour obtenir celui-ci-dessous.

Teste le.

```
from math import *
for k in range(2,6):
```

```
>>> from pour import *
2
3
4
5
```

k varie de 2 à 5

À RETENIR

Pour k allant de 1 à n
instructions
Fin Pour

for k in range(1,n+1):
instructions

On a une répétition n fois des instructions situées entre le Pour et le Fin Pour.

Il s'agit d'une boucle bornée ; la variable k est le compteur qui prend les valeurs de 1 à n

Ex2. Écrire un script (**table5**) en Python qui affiche la table de 5.

```
>>> from table5 import *
5
10
15
20
25
30
35
40
45
```

Ex3. Écrire un script (**nbdiv**) en Python qui définit une fonction appelée nbdiv(n), de paramètres n et qui retourne le nombre de diviseurs de l'entier n.

On utilisera une boucle bornée (pour k allant de ... à ...) qui teste la divisibilité de n par k, et qui affiche en sortie le nombre de diviseurs de n avec

```
from math import *
def nbdiv(n):
    nb=0
    for k in range(1,[]):
        if n%k==[]:
            nb=[]
    return([])
```

a%b renvoie le reste de la division entière de a par b

```
>>> 12%4
0
>>> 8%3
2
```

```
deg PYTHON
>>> from nbdiv import *
>>> nbdiv(12)
6
```

Ex4. On prolonge le script précédent en créant une fonction `est_premier` de l'argument `m` non nul qui retourne si `m` est premier ou non en utilisant la fonction précédente `nbdiv`.

Teste le programme avec les entiers 6 ; 13 et $2^7 - 1$

a^p s'écrit en Python `a**p`

```
1 from math import*
2 def nbdiv(n):
3     nb=0
4     for k in range(1, n):
5         if n%k==0:
6             nb+=1
7     return(nb)
8 def est_premier(m):
9     if nbdiv(m)==1:
10        return("premier")
11    else:
12        return("non premier")
```

Ex5. script nombre premier (`premier`)

saisir n
Pour k allant de 2 à n
 Si k divise n
 alors afficher(« non premier, divisible par »,k)
 Fin Si
Fin Pour
afficher(« premier »)

```
1 from math import*
2 def premier(n):
3     for k in range(2, n):
4         if n%k==0:
5             return("non premier")
6     return("premier")
```

Ex6. script liste des diviseurs (**diviseur**)

`floor(a)` renvoie la partie entière du nombre a.

Si `floor(n/k)==n/k`

cela signifie que la division `n/k` donne un entier égal à `n/k`

On aurait pu écrire

if `n%k==0`:

floor(x) renvoie la partie entière du nombre x (au sens de plus petit entier inférieur ou égal à x)

```
>>> floor(4.1)
```

```
4
```

```
>>> floor(6.99)
```

```
6
```

```
>>> floor(-0.2)
```

```
-1
```

```
from math import *
```

```
def diviseurs(n):
```

```
    L=[]
```

```
    for k in range(1,n+1):
```

```
        if floor(n/k)==n/k:
```

```
            L.append(k)
```

```
    return(L)
```

```
deg PYTHON
>>> from diviseurs import *
>>> diviseurs(12)
[1, 2, 3, 4, 6, 12]
```

Ex7. p79_6 livre sesamath

Comprendre un algorithme en Python

fonction **f(n)**

Qu'affiche cet algorithme pour **f(196)**.

```
def f(n):
    s = 0
    n = str(n)
    L = list(n)
    for i in range(len(L)):
        s = s + int(L[i])
    return s
```

`n=str(n)` définit n comme chaîne de caractères

`L=list(n)` définit une liste

composée de chacun des chiffres de n.

`len(L)` renvoie la longueur de la liste n c'est-à-dire le nombre de caractère dans la liste

`L[i]` correspondant au terme de rang i+1 dans la liste

`int(L[i])` convertit l'élément `L[i]` en nombre entier