

FICHE 4: LES FONCTIONS

On peut simplifier l'écriture des programmes en utilisant des **fonctions**, sur le modèle des fonctions numériques étudiées en mathématiques.

Définition

Une fonction est une succession d'instructions, utilisant un ou plusieurs arguments, regroupées dans un bloc auquel on donne un nom, celui de la fonction, et qui retourne un résultat à l'aide de l'instruction « return ».

La syntaxe Python pour la définition d'une fonction est la suivante :

```
def nom_fonction(liste de paramètres):  
    bloc d'instructions  
    return résultat
```

Corps de la fonction

Comme les instructions **if**, **for** et **while**, l'instruction **def** est une instruction composée. La ligne contenant cette instruction se termine obligatoirement par un deux-points **:**, qui introduit un bloc d'instructions qui est précisé grâce à **l'indentation**. Ce bloc d'instructions constitue le **corps de la fonction**.

En général les fonctions ont un ou des paramètres mais il existe des fonctions sans paramètres.

Exemple

```
def somme():  
    S = 1  
    For i in range(1,21):  
        S = S + (1/3)**i  
    Return S
```

Exemple de Fonctions avec un ou des paramètres

```
def f(x):  
    if x <= 2 :  
        y=3*x-1  
    else :  
        y= -2*x+4  
    return y
```

Un commerçant solde des articles qui ont un prix p de $t\%$ et veut obtenir le prix après rabais. Donner une fonction qui permet de le calculer.

Algorithme	Python
Fonction solde(p,t) $p \leftarrow p - p * t / 100$ retourne(p)	def solde(p,t): p= p-p*t/100 return p

FICHE 5: Le Langage Python

Les bibliothèques de Python

► Certaines fonctions spécifiques au langage Python sont rangées dans des bibliothèques. Pour pouvoir les utiliser, on peut importer entièrement la bibliothèque ou seulement la ou les fonction(s) souhaitée(s).

- L'étoile * permet d'importer toutes les fonctions d'une bibliothèque.

| Exemple : `from math import *` importe toutes les fonctions de la bibliothèque `math`.

- On peut importer d'une bibliothèque seulement les fonctions dont on a besoin.

| Exemple : `from math import sqrt` importe la fonction racine carrée de la bibliothèque `math`.

► Lorsque le nom de la bibliothèque est très long, on peut lui définir un alias en ajoutant `as` suivi de quelques lettres formant l'alias. On peut ensuite utiliser toutes les fonctions de la bibliothèque en faisant précéder leur nom de l'alias.

| Exemple : `import matplotlib.pyplot as plt`

L'instruction `plt.plot(x,y)` permet d'importer la fonction `plot` de la bibliothèque `matplotlib.pyplot` en utilisant l'alias `plt`.

Type de bibliothèque	Nom de la bibliothèque	Syntaxe d'importation
Principales fonctions mathématiques	<code>math</code>	<code>from math import *</code>
Fonctions des probabilités	<code>random</code>	<code>from random import *</code>
Fonctions graphiques	<code>matplotlib.pyplot</code> <code>pylab</code> <code>numpy</code>	<code>import matplotlib.pyplot as plt</code> <code>import pylab as pb</code> <code>import numpy as np</code>

Certains outils de Python sont rangés dans des modules (ou bibliothèques).

Quand on veut utiliser un tel outil, on doit l'importer depuis le module.

On peut importer l'outil seul ou l'ensemble des outils du module.

Racine carrée d'un nombre : `sqrt()`

Exemple :

Calcul de $\sqrt{42,25}$ qui donne 6,5.

```
>>> from math import sqrt
>>> sqrt(42.25)
6.5
```

Calculer avec π : `pi`

Exemple :

Calcul de $3 \times \pi$.

```
>>> from math import pi
>>> 3*pi
9.42477796076938
```

Prendre la partie entière : `floor`

Exemple :

Calcul de la partie entière de $\frac{25}{3}$.

```
>>> from math import floor
>>> floor(25/3)
8
```

Pour obtenir le contenu du module `math`, exécuter les instructions ci-contre :

Par exemple, `cos`, `sin` et `tan` sont aussi à importer depuis le module `math`.

.... NOTE

On peut aussi importer tous les outils du module `math` en une seule fois par l'instruction **`from math import *`** :

Exemple

```
>>> from math import *
>>> sqrt(9*pi)
5.317361552716548
>>> floor(9*pi)
5
```

```
>>> import math
>>> dir(math)
```

Exemple d'utilisation de la bibliothèque dans un programme :

```
from math import sqrt
def racine(x) :
    if x >=0:
        y=sqrt( x )
    else y= "non défini"
    return(y)
```

`sqrt` est la fonction racine carrée

Programmation avec le langage Python

Il existe plusieurs logiciels permettant de programmer en langage Python. Quand on ouvre un logiciel, deux fenêtres apparaissent à l'écran : la console et l'éditeur.

Créer un nouveau programme	On saisit le programme dans l'éditeur On va à la ligne après chaque instruction, mais on peut taper plusieurs instructions sur la même ligne en les séparant par un point-virgule.	
Saisir A	$A=input("A=")$ si A est une chaîne de caractères $A=float(input("A="))$ si A est un flottant $A=int(input("A="))$ si A est un entier On peut utiliser la fonction demande de la bibliothèque lycée	
Afficher A	print(A)	
Afficher un texte, par exemple « oui »	print("oui")	
Affecter à B la valeur de A	$B=A$	
Écrire des commentaires	On écrit les commentaires sur une ligne, précédés de #	
Tester si $A = B$ / si $A \neq B$	$A==B$	$A!=B$
Tester si $A \geq B$ / si $A \leq B$	$A>=B$	$A<=B$
Si {condition C } Alors {instructions A } Sinon {instructions B } Fin Si	if {condition C : {instructions A } else : {instructions B }	Il n'y a pas d'instruction de fin : c'est l'indentation (le décalage vers la droite) qui indique les instructions faisant partie de la structure conditionnelle.
Pour i variant de 1 à n faire {instructions} Fin Pour	for i in range(1, $n+1$) : {instructions}	<ul style="list-style-type: none"> L'instruction for i in range(n), fait parcourir à la variable i tous les entiers de 0 à $n - 1$. L'instruction for i in range(n_0, n), fait parcourir à la variable i tous les entiers de n_0 à $n - 1$. L'instruction for i in range(n_0, n, p), fait parcourir à la variable i les entiers de n_0 jusqu'à l'entier immédiatement inférieur ou égal à $n - 1$ avec un pas de p. Comme pour if, on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.
Tant que {condition C } faire {instructions} Fin Tant que	while {condition C : {instructions}	Comme pour if et for, on utilise l'indentation pour indiquer les instructions faisant partie de la boucle.
Définition d'une fonction f	def f(a,b,c,\dots) : $y= \dots$ return(y)	
Racine carrée de x	sqrt(x) avec la bibliothèque lycée ou le module math	
x à la puissance n	$x**n$	
π	pi avec la bibliothèque lycée ou le module math	
Reste de la division de a par b	$a\%b$ ou reste(a,b) avec la bibliothèque lycée	
Nombre décimal aléatoire compris entre a et b	uniform(a,b) avec la bibliothèque lycée ou le module random	
Entier aléatoire compris entre a et b	randint(a,b) avec la bibliothèque lycée ou le module random	
Longueur d'une chaîne de caractères C	len(C)	
Extraire un caractère d'une chaîne de caractères C	$C[k]$ renvoie le $(k + 1)$ -ième élément de C	
Concaténation de deux chaînes de caractères	"abcd" + "efg" donne "abcdefg"	
Exécuter un programme	Cliquer sur 	