

azNom :	Prénom :	Classe :
---------	----------	----------

## SNT — Fonction : appel et prototypage

Objectifs :

- Connaître l'utilité d'une fonction.
- Savoir appeler (exécuter) une fonction.
- Savoir prototyper une fonction.

### Introduction

Une **fonction** est une **portion de code** effectuant une tâche précise.

Quelle tâche précise ?

Tout dépend de la fonction !

Par exemple, la fonction `print()` permet d'afficher une valeur, c'est son rôle à elle, sa *tâche*.

### À quoi ça sert, une fonction ?

Concrètement, une fonction sert à éviter de réécrire plusieurs fois le même code : on crée (définit) une fonction une fois, puis on peut la « réutiliser » partout dans le programme en l'exécutant, autant de fois que souhaité.



Dans la vie de tous les jours, une fonction ressemble à une recette de cuisine : une fois la recette écrite, il suffit de dire « prépare un café » ou « fais des crêpes » plutôt que de redétailler chaque étape à chaque fois.

### Comment reconnaître une fonction ?

En Python, une **fonction** est un **mot** formé de lettres (et parfois de chiffres) **suivi de parenthèses**.

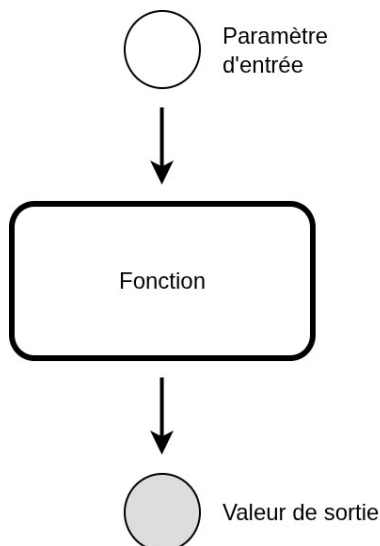
#### ► Exercice 1 — À vous de jouer !

Parmi les lignes codes Python suivants, cochez la case de celles utilisant au moins une fonction :

N°	Code Python	Fonction ? (cochez la case)
1	<code>a = 456 + b // 45</code>	<i>pas de fonction ici</i>
2	<code>for _ in range(45):</code>	× (fonction « range »)
3	<code>a = 456 + a*(b + 12)</code>	<i>pas de fonction ici</i>
4	<code>reponse = input('Saisissez votre nombre SVP ')</code>	× (fonction « input »)
5	<code>print( abs( -10 ) )</code>	× (fonctions « abs » et « print »)

## Déroulement

- En général, une fonction **reçoit** une information (un *paramètre d'entrée*).
- La fonction effectue alors un **traitement** sur cette information.
- Puis, souvent, la fonction renvoie une information en **sortie** (la *valeur de sortie*).



Dans la vie de tous les jours, une fonction s'apparente à un *mixeur de cuisine* : on y introduit des ingrédients (paramètres d'entrée), on appuie sur un bouton, et il nous rend un smoothie prêt à consommer (valeur de sortie) sans qu'on ait à comprendre tout le *mécanisme interne du mixeur*.

## Appel d'une fonction, en Python

En Python, lorsqu'une ligne de code mentionne le nom d'une fonction, on dit que Python va appeler (exécuter) la fonction.

Prenons les exemples suivants :

```
1 print("Salut, toi 🙄")
2 pi = round(3.141592654)
3 prix_ttc = ajoute_tva(368)
```

Et en voici une description, ligne par ligne :

- **Ligne 1** : on appelle la fonction `print()` en lui donnant comme paramètre d'entrée `"Salut, toi 🙄"`
- **Ligne 2** : on appelle la fonction `round()` en lui donnant comme paramètre d'entrée `3.141592654`
- **Ligne 3** : on appelle la fonction `ajoute_tva()` en lui donnant comme paramètre d'entrée `368`

## ► Exercice 2 — Enquêter

Lisez les lignes de codes ci-dessous puis déterminez le nom de la fonction et son paramètre d'entrée :

N°	Code Python	Nom de la fonction	Paramètre d'entrée
1	<code>for _ in range(10):</code>	<code>range</code>	<code>10</code>
2	<code>reponse = input('2 + 2 = ? ')</code>	<code>input</code>	<code>'2 + 2 = ? '</code>
3	<code>print("Teşekkür")</code>	<code>print</code>	<code>"Teşekkür"</code>

## Valeur de sortie d'une fonction

En général, une fonction renvoie une information **en sortie** (la *valeur de sortie*).

Prenons le cas de la fonction `int()`, une fonction qui permet de transformer un nombre à virgule (*float*<sup>1</sup>), en nombre entier (*int*) :

```
1 a = 18.459 # On stocke le flottant 18.459 dans une variable « a »
2 b = int(a) # On envoie la valeur de « a » dans la fonction « int() »
3           # et on stocke la valeur de sortie de « int() » dans une variable « b »
4 print(b)  # Affiche la valeur de « b », (en l'occurrence : « 18 »)
```

Décrivons ligne par ligne le programme ci-dessus :

- **Ligne 1** : on stocke la valeur `18.459` dans une variable `a`.
- **Ligne 2** : on appelle la fonction `int()` avec, comme *paramètre d'entrée*, la valeur stockée dans `a`. Cette fonction retourne une *valeur de sortie*, qui est ici stockée dans une variable `b`.
- **Ligne 3** : on affiche la valeur stockée dans la variable `b`.

## ► Exercice 3 — Enquêter

Déterminez les valeurs affichées dans les programmes suivants :

N°	Code Python	Valeur affichée
1	<code>a = int(10 / 2) # 5.0 dans int() devient « 5 »</code> <code>print(a) # Affiche « 5 »</code>	<code>5</code>
2	<code>blob = 45.123</code> <code>print( int( blob ) )</code>	<code>45</code>
3	<code>print( int( 18.459 ) )</code>	<code>18</code>

1 Voir le cours « Type de donnée : entier, flottant, booléen, chaîne ».

## Quelques fonctions préexistantes

Le langage Python offre un *grand nombre* de **fonctions préexistantes**. Libre à nous d'utiliser telle ou telle fonction, selon le programme que nous sommes en train de créer..

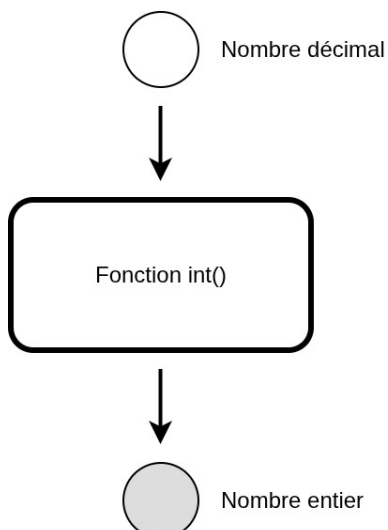
Le tableau ci-dessous tente de présenter *quelques-unes* de ces fonctions préexistantes :

Nom de la fonction	Rôle	Exemple d'utilisation	Valeur de sortie
<code>int()</code>	Convertit en nombre entier.	<code>int( 3.14 )</code>	3
<code>float()</code>	Convertit en nombre décimal.	<code>float( 3 )</code>	3.0
<code>str()</code>	Convertit en chaîne.	<code>str( 3 )</code>	'3'
<code>bool()</code>	Convertit en booléen.	<code>bool ( 1 )</code>	True
<code>len()</code>	Donne la longueur d'une chaîne.	<code>len( 'abcd' )</code>	4
<code>abs()</code>	Convertit en nombre absolue.	<code>abs( -10 )</code>	10

## Prototyper une fonction

Prototyper une fonction, c'est décrire (ou schématiser) à l'avance ce que la fonction est *censée faire*, quelle information elle reçoit et ce qu'elle renvoie, sans forcément écrire tout son code tout de suite.

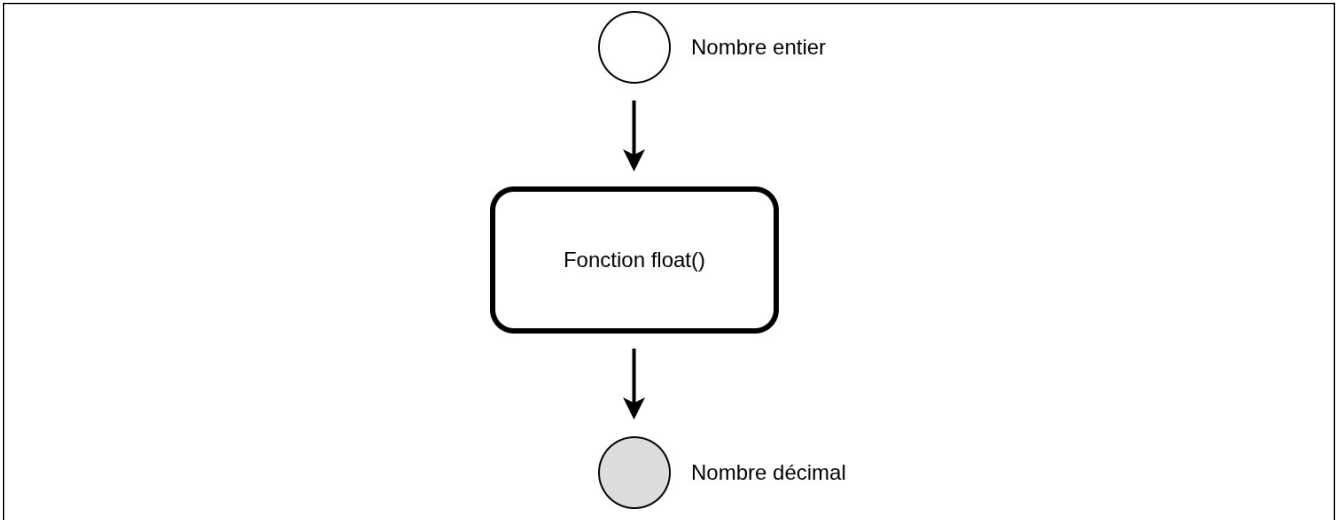
Voici par exemple à quoi pourrait ressembler le prototypage de la fonction `int()` :



Grâce à ce schéma, nous voyons que la fonction se **nomme** « int », qu'elle prend comme **paramètre d'entrée** un nombre décimal, et qu'elle retourne comme **valeur de sortie** un nombre entier.

## ► Exercice 4 — Enquêter

À vous de jouer, en vous aidant du tableau des *fonctions préexistantes* (voir page précédente), dessinez le prototypage de la fonction `float()` :



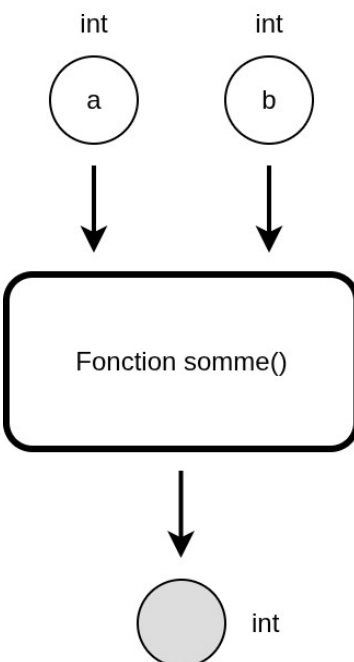
## Plusieurs paramètres d'entrée

Il est tout à fait possible de créer une fonction ayant **plusieurs** paramètres d'entrée.

Voici un exemple de description de fonction :

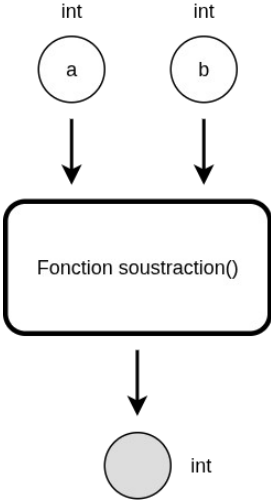
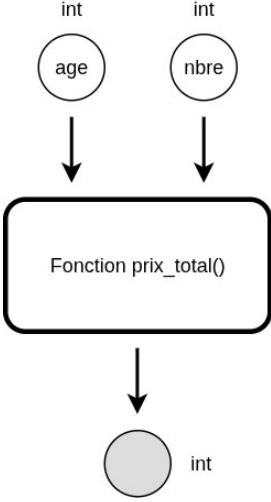
Soit somme le nom de la fonction qui prend comme paramètres deux entiers `a` et `b`. Cette fonction additionne les valeurs de `a` et `b`. Sa valeur de sortie est le résultat entier de cette opération.

Et voici à présent le prototypage réalisé à partir de cette description :



► **Exercice 5** — Enquêter

Réalisez le prototypage de fonctions, à partir de leurs descriptions :

N°	Description	Prototypage
1	<p>Soit <u>soustraction</u> le nom de la fonction qui prend comme paramètres deux entiers <code>a</code> et <code>b</code>. Cette fonction soustrait la valeur de <code>a</code> par celle de <code>b</code>. Sa valeur de sortie est le résultat entier de cette opération.</p>	
2	<p>Soit <u>prix_total</u> le nom de la fonction qui prend comme paramètres les entiers <code>age</code> et <code>nbre</code>. Cette fonction calcule le prix total à payer, selon l'âge et le nombre de place désiré. Sa valeur de sortie est le résultat entier de cette opération.</p>	
3	<p>Soit <u>aire_rectangle</u> le nom de la fonction qui prend comme paramètres les flottants <code>longueur</code> et <code>largeur</code>. Cette fonction calcule l'aire d'un rectangle, selon les valeurs de <code>longueur</code> et <code>largeur</code>. Sa valeur de sortie est le résultat flottant de cette opération.</p>	