

<p>2<sup>nd</sup>e</p> <p>1<sup>er</sup> trimestre</p>	<p><b>Thème : Algorithme et programmation</b></p> <p><b>TP 2 : Fonctions au sens algorithmique</b></p>	<p>Objectif : Découvrir les fonctions en langage python</p> 
--	--	---

**En mathématiques :**

$f : \mathbb{R} \rightarrow \mathbb{R}$   
 $x \mapsto 3x - 5$  est la fonction qui associe à chaque valeur réelle de la variable  $x$  un unique nombre noté  $f(x)$ , que l'on calcule en remplaçant  $x$  par sa valeur dans la formule  $3x - 5$ .

**En algorithmique :**

- Une **variable** est une information stockée temporairement dans une zone mémoire de la machine (RAM). Elle est constituée de deux éléments : son nom et sa valeur. Sa valeur peut changer au cours de l'exécution de l'algorithme.

Il existe plusieurs types de variables :

- les **chaînes de caractères** : par exemple le message "Le triangle est rectangle".
- les **entiers** : 0, 1, 2, 3..., -1, -2...
- les **flottants** : ce type de variable contient une valeur décimale, par exemple 3,14.

- Une **fonction** est une suite d'instructions portant un nom, effectuant une certaine tâche, et utilisant zéro, un ou plusieurs **arguments** ou **paramètres**.

Elle permet de découper le problème étudié en sous-problèmes et d'éviter ainsi la répétition d'instructions.

Une fois qu'elle est définie, on peut l'utiliser autant de fois que nécessaire tout au long de l'exécution de l'algorithme.

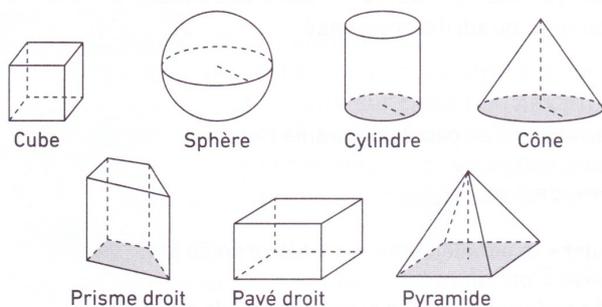
- **Syntaxe en Python :**

```
def nom_de_la_fonction(argument1, argument2, ...):
    instructions
    return(resultat1, resultat2, ...)
```

L'instruction **return** interrompt le programme dès qu'elle s'est exécutée et permet de renvoyer une ou plusieurs valeurs de types entier, décimal (dit flottant) ou chaîne de caractères.

**Exercice 1.** *extrait de Cahier d'Algo, 2<sup>nd</sup>e, collection Barbazo, édition Hachette, p76*

On donne ci-dessous des solides et des scripts de fonctions calculant des volumes. Associer chaque script à un ou plusieurs solides.



```

1 def script1(a,b,c):
  return a*b*c

2 from math import pi
  def script2(r):
    return 4/3*pi*r**3

3 from math import pi
  def script3(r,h):
    return pi*h*r**2

4 def script4(c):
  return c**3

5 def script5(B,h):
  return B*h

6 def script6(B,h):
  return 1/3*B*h

7 from math import pi
  def script7(r,h):
    return 1/3*pi*h*r**2

```

**Exercice 2.** On considère un rectangle.

1. Compléter la fonction `rectangle` qui prend pour arguments la largeur et la longueur du rectangle et renvoie son périmètre et son aire.

```
def rectangle(          ,          ):
    p =
    a =
    return( p , a )
```

2. Que renvoie l'instruction `rectangle(3,4)` ?
3. Ecrire une fonction `diagonale` qui prend pour arguments la largeur et la longueur du rectangle et renvoie la longueur de la diagonale.
4. Que renvoie l'instruction `diagonale(3,4)` ?

**Exercice 3.** On considère une fonction affine  $f$  définie sur  $\mathbb{R}$  par  $f(x) = ax + b$  où  $a$  et  $b$  sont des nombres réels.

On note  $(\mathcal{D})$  sa droite représentative dans un repère.

1. Compléter la fonction `coefficient_directeur` qui prend pour paramètres les coordonnées  $x_A$ ,  $y_A$ ,  $x_B$  et  $y_B$  de deux points  $A$  et  $B$  de la droite  $(\mathcal{D})$  et renvoie le coefficient directeur  $a$  de la droite.

```
def coefficient_directeur(          ,          ,          ,          ):
    a=
    return( a )
```

2. Compléter la fonction `ordonnee_a_lorigine` qui prend pour paramètres les coordonnées  $x_A$  et  $y_A$  d'un point  $A$  de la droite  $(\mathcal{D})$  et le coefficient directeur  $a$  et renvoie l'ordonnée à l'origine  $b$  de la droite.

```
def ordonnee_a_lorigine(          ):
    b=
    return( b )
```

3. Dans le cas où  $a$  est non nul. On note  $C$  le point d'intersection de la droite  $(\mathcal{D})$  avec l'axe des abscisses. L'ordonnée de  $C$  vaut donc 0.

Compléter la fonction `intersection` qui prend pour paramètres les coefficients  $a$  et  $b$  de la fonction affine  $f$  et renvoie l'abscisse du point  $C$ .

```
def intersection(          ,          ):
    x_C =
    return( x_C )
```