

ALGORITHMIQUE

C – Les fonctions

1. Définition d'une fonction

- ▶ Une fonction, en programmation, est un sous-programme qu'on peut utiliser à volonté dans le programme principal.
- ▶ Écrire des fonctions permet d'organiser et de simplifier les programmes.
- ▶ Une fonction a un nom, peut prendre des valeurs en entrée données sous la forme de variables appelées arguments ou paramètres, et peut renvoyer un ou plusieurs résultats. En langage Python, elle est structurée de la manière suivante :

```
1 def nomDeLaFonction(a, b, ...):
2     instruction1
3     instruction2
4     ...
5     return resultat
```

Mot-clé def Nom de la fonction Paramètres

Indentation Mot-clé return pour renvoyer un résultat

- ▶ Les instructions contenues dans la fonction sont décalées vers la droite. Ce décalage, appelé indentation, peut être réalisé en créant 2 ou 4 espaces ou en utilisant la touche tabulation .

Exemple :

On considère la fonction suivante :

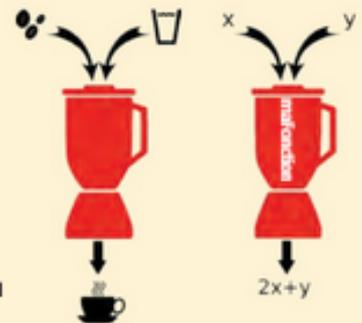
```
1 def maFonction(x, y):
2     resultat=2*x+y
3     return resultat
```

Le nom de la fonction est `maFonction`, elle a deux paramètres en entrée x et y , et elle renvoie le nombre $2 \times x + y$.

Remarques :

- il faut ajouter deux points « : » au bout de la ligne de définition de la fonction ;
- pour renvoyer la valeur de la variable resultat, on peut écrire `return resultat` ou `return (resultat)` ;
- il est également possible de renvoyer plusieurs valeurs en les séparant par des virgules.

```
1 def doubleTriple(x):
2     return 2*x, 3*x
```



Remarque importante: en informatique, l'**indentation** consiste en l'ajout de tabulations ou d'espaces dans un fichier, pour une meilleure lecture et compréhension du code.

L'**indentation** est synonyme de décalage. ... L'une des particularités de **Python** est son utilisation de l' **indentation** pour mettre en évidence les blocs de code.

Lien : mathssa.fr/erreurindentation (les 5 premières minutes)

2.Appel d'une fonction

► Pour appeler (c'est-à-dire utiliser) une fonction, il faut écrire son nom avec les valeurs d'entrée, appelées paramètres, entre parenthèses et dans le bon ordre. Ainsi, pour appeler la fonction `maFonction` pour les valeurs d'entrée (paramètres) $x = 2$ et $y = 3$, il faut écrire l'instruction suivante sous la définition de la fonction :

```
a=maFonction(2,3)
```

► Utiliser des fonctions permet également de modifier facilement un programme. Ainsi, si le calcul n'est plus $2 \times x + y$ mais $5 \times x + y$, il suffira de le changer une seule fois dans la fonction. La ligne 2 de `maFonction` sera alors :

```
2 resultat=5*x+y
```

PROPRIÉTÉS • Une fonction ne renvoie qu'un seul résultat (ce peut être un couple de nombres).

• Une fonction peut n'avoir aucun argument. Par exemple, la fonction `imp()` ci-contre peut être utilisée dans un programme où on résout des équations.

```
def imp():  
    return("impossible")
```

• Une fonction peut être appelée dans un autre programme : il suffit pour cela de l'insérer dans une instruction en saisissant son nom et les valeurs des arguments. La valeur appelée peut être stockée dans une variable.

EXEMPLE : La fonction `f` est appelée dans le programme de la fonction `rep`. Ainsi, `rep(5)` renvoie 'lalalala'.

Si `C` est une chaîne de caractères, `5*C` est la chaîne `C+C+C+C+C`.

```
def f():  
    return("la")
```

```
1 def rep(n):  
2     s=n*f()  
3     return(s)
```

Vidéo : lienmini.fr/10557-04 (de 1mn15s à 3mns50s) et mathssa.fr/fonctionpython (5mns20s)

Qcm : lienmini.fr/10557-05 et lienmini.fr/10557-06