

### ➤ Définition

Une fonction est une suite d'instruction que l'on souhaite définir en début de programme pour pouvoir l'exécuter plus tard à un ou plusieurs endroits dans un programme. La fonction renvoie éventuellement une valeur : un booléen, un nombre, une chaîne de caractères, une liste ...

```
def fct(m, n, ...):           Définition du nom de la fonction.
    """ aide """           Message affiché lorsque l'on demande l'aide sur la fonction : help(fct)
    global a, b, ...        (Optionnel) Variables globales, les modifications seront définitives !
    ...                     Les instructions exécutées à l'appel de la fonction sont dans le bloc situé
    ...                     après la tabulation
    return v                (Optionnel) Met fin à la fonction et renvoie une valeur.
```

Remarques : m, n, ... sont optionnels on peut écrire : `def fct()` :

On peut attribuer une valeur par défaut à ces paramètres avec un « = ».

Si l'on n'utilise pas l'instruction `return`, on parle habituellement de procédure plutôt que de fonction.

Attention : Par défaut les variables utilisées dans la fonction sont des variables locales, les modifications ne sont plus valables en dehors de la fonction. Si l'on veut modifier les valeurs de variable du programme principal, il faut les indiquer après l'instruction `global`.

### ➤ Récurtivité

Une fonction peut s'appeler elle-même, mais attention à bien vérifier qu'elle est bien définie et qu'elle ne va pas tourner sans jamais s'arrêter !

#### Exemple

Soit  $u$  la suite géométrique de premier terme  $u_0 = 2$  et de raison  $q = 3$ .

#### **Mode explicite** : $u_n = u_0 \times q^n$

```
def u(n):
    """ Mode explicite """
    v = 2 * (3**n)
    return v
```

#### **Mode récurrent** : $u_{n+1} = q \times u_n$

```
def u(n):
    """ Mode récurrent """
    if n == 0:
        v = 2
    else:
        v = 3 * u(n - 1)
    return v
```

### ➤ Import

On peut stocker les fonctions dans un module, puis les charger au début du programme qui va les utiliser. Il existe de nombreux modules incluant des bibliothèques de fonctions prédéfinies sur des thèmes variés. Par exemple le module "`math`" contient toutes les fonctions mathématiques utiles en T°S absentes par défaut en Python : les fonctions trigonométriques, exponentielles, logarithmes, les nombres  $e$  et  $\pi$ ...

`import math` Chaque fonction du module `math` devra être précédée du préfixe "`math.`"

Remarque : On peut changer ce préfixe en écrivant par exemple : `import math as m` (le préfixe sera "`m.`")

`from math import *` Toutes les fonctions du module `math` sont accessibles directement sans préfixe.

Attention : En cas d'import total sans préfixe, les fonctions importées risquent d'écraser des fonctions existantes portant le même nom... il est conseillé de n'importer que les fonctions utiles au programme en remplaçant "`*`" par les fonctions souhaitées, séparées par une virgule.