

# A<sub>4</sub>(Algorithmique) : Fonctions.

```
1 def nom_de_la_fonction(paramètre1 , paramètre2 , ....):
2     instruction1
3     instruction2
4     ....
5     return résultat_retourné_par_la_fonction
```

**Definition 0.1** Pour des raisons pratiques( par exemple, éviter de répéter des instructions), il peut-être utile de définir une **fonction** c'est à dire un bloc d'instructions qui ne sera exécuté que s'il est appelé.

Une fonction possède généralement des paramètres et retourne une **valeur de retour** (mais pas systématiquement, par exemple si elle réalise simplement un affichage). En informatique quand une fonction ne retourne pas de valeur, on dit que c'est une **procédure**.



## Exercice 0.1

```
1 def fonction1(a):
2     y=a**2+a
3     return y
4
5 def fonction2(b):
6     z=2b+1
7     return ma_fonction(z)
```

- Donner la valeur de **fonction1(10)**; **fonction1(5)**; **fonction1(-10)**.
- Donner la valeur de **fonction2(10)**; **fonction2(5)**; **fonction2(-10)**.
- Donner en fonction de  $a$  l'expression algébrique de **fonction2(fonction1(a))**.
- Donner en fonction de  $a$  l'expression algébrique de **fonction1(fonction2(a))**.



## Exercice 0.2

```
1 def calcul_de_IMC(masse , taille):
2     IMC=masse/(taille*taille)
3     return IMC
4
5 IMC_de_jean=calcul_de_IMC(60,1.60)
6 massePaul=85
7 taillePaul=1.90
8 print(IMC_de_jean)
9 print(Calcul_de_IMC(massePaul, taillePaul))
```

- ➡ Quel est le nom de la fonction de ce programme ?
- ➡ Combien possède t-elle de paramètres ?
- ➡ Le programme imprime deux nombres lequel ?