

Cours: Les fonctions

I) Definition

Une fonction est une série d'instructions (un algorithme) isolée du reste du programme et qui peut être appelée par son nom à n'importe quel endroit du programme et autant de fois que l'on veut.

Deux avantages :

- Eviter les répétitions et gagner ainsi de la place mémoire
- "Agencer" un programme et le clarifier :

Si vous voulez créer un jeu, vous ne pouvez pas travailler efficacement sur le même code à 10, il vous faudra répartir les tâches :

- L'un créera les fonctions de déplacement
- L'un créera une fonction son_defaite
- L'un créera les fonctions d'interactions avec le décor
-

II) Structure

Toutes les fonctions sont déclarées en début de programme avec la syntaxe suivante :

```
def nom_fonction(argument1, argument2):  
    "explication de la fonction"  
    instruction 1  
    instruction 2  
    instruction 3  
  
    return(valeur)
```

Des exemples :

Exemple 1

```
def trait():  
    "trace un trait"  
    print("-----")
```

Cette fonction ne prend aucun argument, lorsqu'elle est appelée elle tire un trait. Elle ne retourne pas de valeur.

```
>>> trait()  
-----
```

Exemple 2

```
def trait(nombre):  
    "trace un trait de nombre '-'"  
    for i in range(nombre):  
        print("-", end="")
```

Cette fonction prend un argument, lorsqu'elle est appelée elle tire un trait composé de nombre '-'. Elle ne retourne pas de valeur.

```
>>> trait(5)  
-----  
>>> trait(15)  
-----
```

Exemple 3

```
def puissance(a,b):  
    "calculer a^b"  
    p=1  
    for i in range(b):  
        p=a*p  
    return(p)
```

Quel est le type de a et b ?

Cette fonction calcule le nombre a^b . Elle prend deux arguments et retourne une valeur.

```
>>> puissance(2,5)  
32
```

Attention à ce que renvoie la fonction :

a=trait(5) n'est pas une instruction valide car trait(5) ne renvoie rien et ne peut être stockée dans une variable

b=puissance(2,7) fonctionne très bien, la fonction renvoie un nombre que l'on stocke dans la variable b.