

# Logiciel de calcul formel

Loïc Demange

loic.demange@etud.univ-paris8.fr

12 février 2021



Objectif du TP :

- Créer une fonction qui prend une liste d'entiers positifs en entrée et retourne l'entier le plus grand de la liste.

**Indication** Indiquer le type de sortie ne suffit pas puisque ce n'est qu'une indication, il faut donc réaliser des vérifications avec la fonction **type**, en retournant un code erreur (par exemple **-1**) s'il y a un souci.

**Créer une fonction qui prend une liste d'entiers positifs en entrée et retourne l'entier le plus grand de la liste.**

```
def plusgrand(l:list) -> int:
    n = l[0]
    for i in range(1,6):
        if l[i] > n:
            n = l[i]
    return n
```

```
l = [1,2,8,4,5,6]
plusgrand(l)
```

**Créer une fonction qui prend une liste d'entiers positifs en entrée et retourne l'entier le plus grand de la liste.**

```
def plusgrand(l:list) -> int:
    n = l[0]
    for i in range(1,len(l)):
        if l[i] > n:
            n = l[i]
    return n
```

```
l = [1,2,8,4,5,6]
plusgrand(l)
```

**Créer une fonction qui prend une liste d'entiers positifs en entrée et retourne l'entier le plus grand de la liste.**

```
def plusgrand(l:list) -> int:
    if type(l) != type([]):
        return -1

    n = l[0]
    for i in range(1,len(l)):
        if l[i] > n:
            n = l[i]
    return n
```

```
l = [1,2,8,4,5,6]
plusgrand(l)
```

**Remarque** La vérification du type en Python se fait souvent à l'appel de fonction et non au sein de la fonction. Il s'agit juste d'apprendre à utiliser la méthode **type**.

**Créer une fonction qui prend une liste d'entiers positifs en entrée et retourne l'entier le plus grand de la liste.**

```
def plusgrand(l:list) -> int:
    if type(l) != type([]):
        return -1

    n = l[0]
    for i in range(1,len(l)):
        if (type(l[i]) != type(1)) or (l[i] < 0):
            return -1

        if l[i] > n:
            n = l[i]
    return n
```

```
l = [1,2,8,4,5,6]
plusgrand(l)
```

**Remarque** La vérification du type en Python se fait souvent à l'appel de fonction et non au sein de la fonction. Il s'agit juste d'apprendre à utiliser la méthode **type**.

La suite de Fibonacci est une suite définie par les relations  $F_0 = 0$ ,  $F_1 = 1$  et  $F_n = F_{n-1} + F_{n-2}$  pour  $n \geq 2$ .

Le but ici est d'implémenter sur Sage la suite de Fibonacci.

Objectif du TP :

- Créer une fonction qui prend en paramètre  $n$  l'indice de la suite, et renvoie une liste contenant les  $n$  premiers éléments de la suite.