


NSIT	e07a	Révisions
Kata 1		
Tests unitaires, spécifications, listes, récursivité		1/1

**Exercice** : Trouver le plus grand élément d'une liste

**Énoncé** : Même problématique qu'à l'exercice précédent : Il faut parcourir l'ensemble de éléments de la liste.

Mais cette fois on recherche le plus grand.



```
# A vous de jouer.
def maximum(ma_liste:list)->float:
    """ Renvoie le plus grand élément de la liste ma_liste
        Précondition: ma_liste n'est pas vide
    """
    m = ma_liste[0] # Initialise la maximum au premier élément de la liste
    ...
    return m

assert maximum([1,2,5,3])==5, 'La fonction maximum ne renvoie pas le bon résultat sur une liste d\'entiers'
assert maximum([1.8,2.4,5.1,3.1])==5.1, 'La fonction maximum ne renvoie pas le bon résultat sur une \
liste de flottants'

L = [randint(1,100) for i in range(1,20)] # Tire une liste aléatoire
print(L)
maximum(L)
```

**Commentaire** :

Pour le parcours d'une liste avec un itérateur ou un indice, voir l'aide précédente.

Dans cet exercice, on initialise une variable `m` avec la première valeur de la liste.

Invariant de boucle : A tout moment dans notre parcours, cette variable va contenir le maximum de la liste parcourue.

C'est évidemment vrai pour le premier élément ... puisqu'il n'y en a qu'un seul.

Ensuite, on parcourt chaque élément et on regarde si ce nouvel élément est plus grand que `m`. Si c'est le cas, on met à jour la nouvelle valeur de `m` et on passe au suivant.