
TD 9 (programmation) - Fonctions en paramètres, `map` et `filter`.

Remarques :

1. Certains exercices nécessitent de déterminer le type d'une variable. Pour ce faire, on utilise la syntaxe `type(var) is float`, qui est ici évaluée à `True` si `var` est une variable réelle (remplacez bien entendu `float` par le type qu'il vous faut).
2. Quand c'est possible, essayez d'utiliser `map` et `filter` plutôt que des compréhensions.
3. Si vous ne voyez pas comment vous passer des compréhensions, construisez-les avec des parenthèses (par exemple `(x**2 for ...)`).

Exercice 1

- (a) Écrivez une fonction `temps_exec(fonction, parametre)` qui renvoie le temps que prend la fonction passée en paramètre pour s'exécuter avec le paramètre donné.
- (b) Écrivez une fonction `liste_temps_exec(fonction, parametres)` d'une ligne qui renvoie la liste des temps d'exécutions de la fonction donnée pour chaque élément de la liste `parametres`.
- (c) Écrivez une fonction `liste_temps_exec_moyen(fonction, parametres, p)` d'une ligne qui renvoie la liste des temps d'exécutions moyens de la fonction donnée en paramètre pour chaque valeur de la liste `parametres`. Les moyennes sont calculées sur `p` exécutions pour chaque valeur.

Exercice 2

Écrivez une fonction d'une ligne renvoyant la somme des éléments impairs d'une liste d'entiers.

Exercice 3

Écrivez une fonction qui prend une liste d'entiers en argument et renvoie, en une ligne, `True` s'il existe un ensemble de cases consécutives dans la liste dont la somme vaut 0, et `False` sinon. Par exemple :

```
>>> contient_somme_nulle([5, 3, -1, -2, 6])
True
>>> contient_somme_nulle([5, 3])
False
>>> contient_somme_nulle([1, 2, -2])
True
```

Vous pouvez utiliser la fonction `sum(L)`, qui renvoie la somme des éléments d'un itérable `L`.

Exercice 4

Écrivez une fonction `separer(liste)`, qui renvoie un tuple de listes contenant respectivement les entiers, les réels et les chaînes de la liste passée en paramètre. Par exemple :

```
>>> separer(["bla", 1, "deux", 3.14, 42])
([1, 42], [3.14], ["bla", "deux"])
```

Exercice 5

Écrivez une fonction d'une ligne renvoyant la concaténation des lettres minuscules d'un texte. Par exemple :

```
>>> catmin("AaBBbCCcd")
"abcd"
```

Exercice 6

Écrivez une fonction `modifier(liste, f)`, qui remplace chaque élément de la liste par son image par la fonction `f`. Les modifications doivent se faire en place.

Exercice 7

Soit `lst` une liste contenant des entiers et des chaînes.

- (a) Utilisez `lst.sort(...)` avec le paramètre `key` pour écrire une fonction plaçant toutes les chaînes de `lst` au début de `lst`.
- (b) Écrivez une fonction qui renvoie la position du premier entier d'une liste (`None` si la liste n'en contient pas).
- (c) Combinez ces deux parties pour écrire une fonction qui trie `lst` en mettant les chaînes dans l'ordre lexicographique au début de `lst`, puis les entiers dans l'ordre croissant. Par exemple, pour `lst = [3, "b", "a", 1, 2]`, la fonction devra transformer `lst` en `["a", "b", 1, 2, 3]`.

Exercice 8

Écrivez une fonction renvoyant le nombre de nombres premiers dans une liste donnée. Vous pouvez supposer l'existence d'une fonction `est_premier(n)`, renvoyant `True` si `n` est premier et `False` sinon.