

## TD (feuille 2) : Les listes

**Objectifs** Manipulation des listes et définitions récursives sur les listes.

**Exercice - Définition d'une structure de données d'ensemble**

Nous voulons représenter les ensembles par des listes dans lesquelles il n'y a pas de doublons.

1. Définir la valeur `vide` de type `'a list` qui représente l'ensemble vide.
2. Définir la fonction `singleton` de type `'a -> 'a list` qui crée un ensemble à un élément.
3. Définir la fonction `cardinal` de type `'a list -> int` qui retourne le nombre d'éléments dans un ensemble.
4. Définir la fonction `appartient` de type `'a -> 'a list -> bool` qui teste l'appartenance d'un élément dans un ensemble.
5. Définir la fonction `inclusion` de type `'a list -> 'a list -> bool` qui teste si le premier ensemble est inclus dans le second.
6. Définir la fonction `egal` de type `'a list -> 'a list -> bool` qui teste l'égalité de deux ensembles.
7. Définir la fonction `ajout` de type `'a -> 'a list -> 'a list` qui ajoute un élément dans un ensemble.
8. Définir la fonction `union` de type `'a list -> 'a list -> 'a list` qui calcule l'union de deux ensembles.
9. Définir la fonction `intersection` de type `'a list -> 'a list -> 'a list` qui calcule l'intersection de deux ensembles.
10. Définir la fonction `min` de type `'a list -> 'a` qui retourne le plus petit élément d'un ensemble.
11. Définir la fonction `supprime` de type `'a -> 'a list -> 'a list` qui supprime un élément d'un ensemble.
12. Définir la fonction `difference` de type `'a list -> 'a list -> 'a list` qui supprime du premier ensemble tous les éléments qui sont inclus dans le second.
13. Définir la fonction `set_of_list` de type `'a list -> 'a list -> 'a list` qui construit un ensemble à partir d'une liste.
14. Peut-on implanter les ensembles avec des listes qui ont des doublons?
15. Peut-on implanter les ensembles avec des listes triées?