

CHAPITRE 4 – FONCTIONS D'ORDRE SUPÉRIEUR

DÉFINITION

- Une fonction d'ordre supérieur est une fonction qui prend en paramètre des fonctions
- Certaines fonctions d'ordre supérieur sont très utilisées pour travailler sur les listes
 - Elles permettent d'automatiser certains traitements, et donc de rendre le code plus concis et plus lisible
 - Ces fonctions sont : `List.map`, `List.filter` et `List.fold_left`

LIST.MAP

- La fonction `List.map` prend en paramètre une fonction `f` et une liste `[x1;x2;...;xn]`, et retourne la liste `[f(x1);f(x2);...;f(xn)]`
 - Cette fonction applique donc le même traitement à tous les éléments d'une liste
 - Elle retourne une liste de la même taille que la liste initiale
 - Exemple d'utilisation : retourner la liste des prénoms de tous les utilisateurs d'une liste, enlever un point à toutes les notes d'une liste, mettre au carré tous les entiers positifs d'une liste (en laissant les entiers négatifs tels quels), etc.

LIST.FILTER

- La fonction `List.filter` prend en paramètre une fonction `p` et une liste `l`, et retourne la liste des éléments de la liste `l` pour lesquels la fonction `p` est vraie
 - Elle retourne une liste généralement plus petite que la liste initiale
 - Les éléments de la liste retournée n'ont pas été modifiés (mais certains ont disparu)
 - Exemple d'utilisation : retourner la liste des personnes majeures, retourner la liste des étudiants ayant validé l'UE, etc.

LIST.FOLD_LEFT

- La fonction `List.fold_left` prend en paramètre une fonction, une valeur de base `b`, et une liste `[x1;x2;...;xn]`, et elle « replie » la liste autour d'une valeur unique
 - La valeur retournée est $f(\dots f(f(b, x1), x2) \dots, xn)$
- Exemple : calcul de la somme
 - Si on veut calculer la somme $x1 + x2 + \dots + xn$, on va utiliser le cas de base 0 et la fonction `+`
 - On va donc écrire « `let add x y = x+y;;` » et « `let somme l = List.fold add 0 l;;` »
- Beaucoup de fonctions peuvent s'écrire avec `List.fold_left`, car le traitement effectué est en fait très général