# TD de programmation fonctionnelle et logique
# TD 1 : définitions, filtrage

## 1 Retour sur les listes

```
# let trois_entiers = [1;2;3];;
 val trois_entiers : int list = [1; 2; 3]
# let reste = function []->failwith "liste vide !" | _::r ->r;;
 val reste : 'a list -> 'a list = <fun>
# 4::(reste(reste (0::trois_entiers)));;
 - : int list = [4; 2; 3]
# trois_entiers;;
 - : int list = [1; 2; 3]
```

## 2 Définitions

1. Définissez une variable `a` égale à `3*b` où `b` vaut 7. Attention, nous ne voulons pas que `b` soit définie globalement !
2. Définissez une fonction qui à l'entier `x` associe `x+a` où `a` vaut localement 3.
3. Définissez une fonction `f` qui à l'entier `x` associe l'entier `2*x+1`.
4. Définissez une fonction `g` qui à l'entier `x` associe $(2*x+1)^2$.
5. Même question que précédemment, mais en ne calculant qu'une fois la valeur de `2*x+1` (et sans utiliser de fonction "carré").
6. Même problème qu'à la question 4, mais en utilisant une fonction locale pour calculer le produit `2*x+1`.
7. Même problème qu'à la question précédente, mais en n'effectuant qu'un seul appel à la fonction locale.

## 3 Filtrage

**Attention** : tous les filtrages doivent être exhaustifs !

1. Nous voulons trois versions d'une fonction qui prend une paire d'entiers en argument et qui renvoie la somme des éléments de la paire :
   (a) une version simple (sans filtrage) ;
   (b) une version avec filtrage par fonction anonyme ;
   (c) une version avec filtrage explicite (et prenant en entrée un unique argument `p`).
2. Écrivez une fonction qui prend en entrée une liste et qui renvoie la somme des deux premiers entiers la constituant. Proposez plusieurs solutions (avec et sans filtrage (explicite ou non)).
3. Écrivez deux versions d'une fonction qui prend en entrée les coordonnées de deux points (sous forme de paire de réels) et qui renvoie le carré de la longueur du segment correspondant, une version simple et une avec un filtrage explicite (la fonction prend alors en entré deux arguments `a` et `b`).

4. Écrivez une fonction qui prend en entrée une liste de paires et qui renvoie la somme des éléments de son premier élément (sic).
5. Écrivez une fonction qui prend en argument une paire de listes et qui renvoie la somme du premier élément de la première liste avec le premier élément de la deuxième.
6. Même question que précédemment, mais sans utiliser le symbole « _ ».
7. Écrivez une fonction qui prend en entrée une paire et une liste et qui fait la somme des premiers éléments de ses deux arguments.
8. Écrivez une fonction qui prend en entrée une liste de trois entiers, notée `[a;b;c]` et qui renvoie la liste `[c;b;a;b;c]`. Cette fonction devra utiliser un synonyme.