

# Programmation fonctionnelle - TD2

Licence 3

Stefano Guerrini

5 février 2010

1. Donner le type des expressions suivantes :
  - (a) `fun x -> x`
  - (b) `fun x y -> x y`
  - (c) `fun (x,y) -> (y,x)`
  - (d) `fun x y -> x (x y)`
  - (e) `fun n x y -> x (n x y)`
2. Donner des exemples de fonctions avec les types suivants :
  - (a) `'a -> 'b`
  - (b) `('a -> 'b) -> 'a`
  - (c) `'a -> 'b -> 'a * 'b`
  - (d) `('a * 'b -> 'c) -> 'a -> 'b -> 'c`
  - (e) `('a -> 'b -> 'c) -> 'a * 'b -> 'c`
  - (f) `('a -> 'b -> 'c) -> ('a -> 'b) -> 'a -> 'c`
3. Donner la déclaration des type énumérés suivantes et des fonctions après spécifiées.
  - (a) Donner la déclaration du types `semaine` pour le noms des jour de la semaine et écrire une fonction qui prend comme argument un jour de la semaine et renvoie le numéro du jour de la semaine (Dimanche = 1, Lundi = 2, etc.).
  - (b) Donner une déclaration du type `mois` pour le noms des mois et écrire une fonction qui prend comme argument un mois et renvoie le numéro des jours du mois.
4. Écrire les suivantes fonctions sur les listes.
  - (a) Donner la une fonction qui calcul la somme des élément dans une liste. Quelle est le type de la fonction que vous avez écrit ?
  - (b) Donner une fonction qui calcule la somme des carrés des éléments dans une liste.
  - (c) Donner une fonction qui calcul le produit de la somme des éléments en position impair dans une liste et de la somme des éléments en position pair (Aide : les éléments en position pair dans `n::1` sont les éléments en position impair dans `1`; les éléments en position impair sont `n` et les éléments en position pair dans `1`).
  - (d) Donner une fonction de type `int list -> int * int` qui calcule la somme des éléments en position impair dans une liste et le produit des éléments en position pair (Aide : donner au même temps, par récursion mutuelle, la définition de la fonction qui calcule la somme des éléments en position pair et le produit des éléments en position impair).