

Programmation Fonctionnelle

Listes & Exercices

Adrien Durier

26 septembre 2023

Accès aux éléments d'une liste

On peut accéder aux éléments d'une liste avec `List.hd` et `List.tl` (bibliothèque `List`)

```
# List.hd [3;6;1;2];;  
- : int = 3  
  
# List.tl [3;6;1;2];;  
- : int list = [6;1;2]
```

`List.hd` et `List.tl` échouent sur une liste vide

```
# List.hd [];;  
Exception: Failure "hd".  
  
# List.tl [];;  
Exception: Failure "tl".
```

Accès aux éléments d'une liste

On accède aux éléments d'une liste à l'aide des fonctions prédéfinies `List.hd` et `List.tl`

```
let hd l = match l with  
| [] -> failwith "hd"  
| x :: xs -> x
```

```
let tl l = match l with  
| [] -> failwith "tl"  
| x :: xs -> xs
```

Exercice : Récursion terminale

Exercice : Écrire une version récursive terminale de la fonction `somme_carre` de type `int -> int` qui renvoie la somme des carrés de 1 à `n`.

```
let somme_carre n =
```

Exercice : Récursion terminale

Exercice : Écrire une version récursive terminale de la fonction `somme_carre` de type `int -> int` qui renvoie la somme des carrés de 1 à `n`.

```
let somme_carre n =  
  let rec somme_rec acc x =  
    if x > n then acc  
    else somme_rec (x * x + acc) (x + 1)  
  in  
    somme_rec 0 1
```

Exercice : Typage

Énoncé : Les fonctions suivantes sont-elles bien typées ? Si oui, donner leur type, sinon préciser pourquoi.

```
let f1 x g z = if x then g x else g z
```

```
let f2 f g x = if f x then g x
```

```
let f3 g x y = if g x then g y
```

```
let rec f4 x y = f4 y ([]::x)
```

Exercice : Typage

Énoncé : Les fonctions suivantes sont-elles bien typées ? Si oui, donner leur type, sinon préciser pourquoi.

```
let f1 x g z = if x then g x else g z
val f1 : bool -> (bool -> 'a) -> bool -> 'a = <fun>
let f2 f g x = if f x then g x else x + g x
val f2 : (int -> bool) -> (int -> int) -> int -> int = <fun>
let f3 g x y = if g x then g y else g (x y)
Error: This expression has type 'a -> 'b but an expression was expected
let rec f4 x y = f4 y ([]::x)
val f4 : 'a list list -> 'a list list -> 'b = <fun>
```