

TD 1 Programmation Fonctionnelle

Exercice 1 :

Dans l'interpréteur, tapez les lignes suivantes en observant les résultats :

```
# 12 + 56;;
# 12 + 56.5;;
# 12 +. 56.5;;
# 12. +. 56.5;;
# 'a';;
# "a";;
# ('a', "a");;
# (true, 123, 'b', 4.5, "test");;
# let x = ("toto", 1234);;
# fst x;;
# snd x;;
# let x = ("toto", 1234, true);;
# fst x;;
# snd x;;
# x = ("toto", 1234, true);;
# x = ("titi", 1234, true);;
# (function x -> function y -> x + y) 12 34;;
# (fun x -> function y -> x + y) 12;;
# let l1 = [1;2;3];;
# let l2 = -1 :: (0 :: l1);;
# l1 @ l2;;
```

Exercice 2 :

1. Définissez la fonction `poly : float * float * float -> float -> float` qui a deux arguments : un triplet de flottants (a, b, c) et un flottant x. Cette fonction doit donner le résultat de l'évaluation du polynôme du second degré $ax^2 + bx + c$.
2. Définissez ensuite les fonctions `p1 : float -> float` et `p2 : float -> float` permettant respectivement l'évaluation des polynômes $2x^2 - x + 3$ et $-x^2 + 12$. Pensez à utiliser `poly`!

Exercice 3 :

Le but de cet exercice est de vous montrer qu'il est possible de se passer d'expressions communes en programmation en utilisant des formules mathématiques.

1. Sans utiliser ni les expressions conditionnelles (if-then-else) ni le filtrage, définissez une fonction `max` :
`int * int -> int` qui calcule le maximum de deux arguments.
2. De la même manière, définissez une fonction qui calcule le minimum de deux arguments.

Exercice 4 :

Définissez une fonction qui calcule le pgcd de deux entiers.