

## Exercice 1 : Méthode de Newton

### Dérivées

Écrire une fonction `derive` qui prend en argument une fonction `f: float -> float` et un flottant `x` et qui renvoie la dérivée de `f` en `x` :

$$f'(x) = \frac{f(x + dx) - f(x)}{dx}$$

(On pourra prendre `dx = 1e-5`).

À l'aide de la fonction `derive`, calculer la dérivée de la fonction  $x \mapsto x^2$  en 5, puis sa dérivée seconde en 5.

### Méthode de Newton

La méthode de Newton permet de trouver la racine d'une équation  $f(x) = 0$  en partant d'une racine assez proche  $x_0$  puis en itérant :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Sous des hypothèses de convexité locale, on montre que  $(x_n)$  tend vers la racine.

On va implémenter cette méthode. Pour savoir quand s'arrêter, on regarde si  $|x_{n+1} - x_n| < \varepsilon$ , où  $\varepsilon$  est un seuil fixé. Écrire une fonction `newton` qui prend en argument un seuil `epsilon`, une fonction `f` et un flottant `x0` et qui renvoie `x` une approximation de la racine.

Tester cette fonction en calculant

- $\sqrt{2}$  via `newton 1e-6 (function x -> x *.x) 1.`
- $\pi$  via `newton 1e-6 (function x -> cos(x/.2.)) 1.`

## Exercice 2 : Tri fusion

Dans la suite on dira qu'une liste est triée si elle est triée par ordre croissant.

1. Écrire une fonction récursive `coupe` de type `'a list -> ('a list * 'a list)` qui découpe une liste en deux listes de taille égale, à un près.
2. Écrire une fonction récursive `fusion` de type `('a list * 'a list) -> 'a list` qui fusionne deux listes triées en une seule liste triée.
3. En déduire une fonction récursive `tri_fusion` de type `'a list -> 'a list` qui trie une liste donnée en argument.

## Exercice 3 : Crible d'Érathostène

On rappelle l'existence de la fonction `List.filter` qui prend en argument un filtre `f: 'a -> bool` et une liste `l` et renvoie la liste composée des éléments de `l` qui vérifient le filtre `f`.

Proposer une implémentation de cette fonction.

1. En vous aidant de `List.filter` écrire une fonction `enleve_multiple` qui prend en argument un entier `x` et une liste `l` et enlève les multiples de `x` de la liste `l`.
2. Écrire une fonction récursive `intervalle` qui prend en entrée deux entiers `m` et `n` et qui renvoie la liste des entiers de `m` à `n` inclus.
3. Le principe du crible d'Érathostène est de rayer de la liste des entiers de 2 à  $n$  les multiples stricts de 2, puis des multiples stricts du premier nombre non rayé (ici 3) et ainsi de suite, jusqu'à avoir épuisé la liste du départ. On renvoie alors la liste des nombres non rayés, qui correspond à la liste des nombres premiers de 2 à  $n$ .  
Écrire `erathostene` qui prend en argument un entier `n` et qui renvoie la liste des nombres premiers de 2 à  $n$ . (Indication : écrire une fonction récursive auxiliaire `enleve x l` qui enlève les multiples de `x` dans `l` et qui s'appelle récursivement sur le premier nombre non enlevé).