

Pour interpréter OCaml, le plus simple est d'utiliser l'interpréteur en ligne <https://try.ocamlpro.com>. Vous pouvez aussi utiliser un IDE avec un interpréteur sur votre ordinateur si vous préférez.

Exercice 1 :**Utilisation let... in**

La fonction `exp : float -> float`, qui est prédéfinie, correspond à l'exponentielle.

1. En utilisant la syntaxe `let ... in`, écrire une fonction `th` qui calcule la tangente hyperbolique

$th : x \mapsto \frac{e^x - e^{-x}}{e^x + e^{-x}}$ en effectuant un seul appel à la fonction `exp`.

2. Trouvez une deuxième façon de répondre à la question, en définissant une variable locale de type différent.

Exercice 2 :**Introduction au typage**

Prévoir la réponse de l'interpréteur **OCaml** à chacune des phrases suivantes (puis vérifier sur l'interpréteur) :

```

1 let a = 42
2 let a =
  let f x = 3 * x + 1 in
4   f 1
  let f x = 2 * x + 1
6 let a = f 1
  let f x = 3. * x + 1

```

Exercice 3 :**Typage de fonctions**

1. Déterminer le type des fonctions suivantes (puis vérifier sur l'interpréteur) :

```

1 let f1 (x, y) = 2 * x - 3 * y
2 let f2 x = sin x +. x
  let f3 (x, y) = if x -. 1. > y then x else y
4 let f4 = fun (x, y, z) -> (x, (y, z))

```

2. Faire de même avec les fonctions suivantes. Attention! Ici le résultat et/ou les termes de l'argument peuvent être de type fonction. Nous travaillerons plus en détail la semaine prochaine sur ces fonctions dites *curryfiées*.

```

1 let g1 (x, y) = x y
2 let g2 = fun x -> fun y -> fun z -> x
  let g3 x = fun (y, z) -> x (y z)
4 let g4 (x, y) = x (x y)

```