

PROGRAMMATION FONCTIONNELLE 2

MANIPULATION DE FONCTIONS

L'objectif de la séance est de manipuler des fonctions. Notamment, nous allons passer pas mal de temps à construire des fonctions à partir d'autres fonctions.

Récapitulons

OCaml est un langage de programmation fonctionnelle français, gratuit et disponible sur le site <http://www.ocaml.org>. Vous pouvez trouver de la documentation sur ce même site, en anglais. Vous pouvez aussi consulter en ligne le livre *Développement d'applications avec Objective Caml*, disponible à l'adresse <http://www.pps.jussieu.fr/Livres/ora/DA-OCAML/index.html>.

Vous pouvez éditer le code OCaml par exemple à l'aide de l'éditeur de texte Scite, gratuit et disponible à l'adresse <http://www.scintilla.org/SciTE.html>.

Jusqu'à présent, vous avez manipulé le *toplevel* d'OCaml : vous tapez des commandes, OCaml répond.

Vous avez vu comment manipuler les entiers, les booléens, les nombres à virgule et les fonctions, comment appliquer une fonction à une valeur et comment donner un nom à une valeur. Vous avez aussi vu comment définir une fonction récursive – et donc comment faire une boucle. En cas d'hésitation, n'hésitez pas à vous référer au chapitre 2 de *Développement d'applications avec Objective Caml*.

Mise en jambe

Exercice 1. Définissez la fonction identité sur les entiers.

Exemple :

```
# identite 5;;  
- : int = 5
```

Quel est son type ?

Exercice 2. Que fait l'instruction suivante ?

```
let modulo p q =  
  let r = p mod q in  
  if r >= 0 then  
    r  
  else  
    r+q;;
```

Exercice 3. Définissez la fonction caractéristique de l'ensemble $\{n \in \mathbf{N} / n \text{ divisible par } 3\}$ dans `int`, c'est-à-dire la fonction qui, à un nombre positif et divisible par 3, associe `true` et à tout autre nombre associe `false`.

Exemple :

```
# divisible_par_trois 7;;  
- : bool = false  
# divisible_par_trois 90;;  
- : bool = true
```

Exercice 4. Définissez de même la fonction caractéristique de l'ensemble \mathbf{N} dans `int`. Quel est son type ?

Fonctions sur des fonctions

Rappel Avec OCaml, une fonction n'est jamais qu'une valeur comme une autre.

Exemple :

```
# function x -> x+1;;  
- : int -> int = <fun>
```

est la fonction qui à tout entier x associe $x + 1$.

Exercice 5. Que signifie le type $(\text{int} \rightarrow \text{bool}) \rightarrow (\text{int} \rightarrow \text{bool})$?

Exercice 6. Que signifie le type $(\text{int} \rightarrow \text{bool}) \rightarrow (\text{int} \rightarrow \text{bool}) \rightarrow (\text{int} \rightarrow \text{bool})$?

Exercice 7. Définissez `complement_dans_int : ('a -> bool) -> ('a -> bool)`, une fonction qui, à la fonction caractéristique d'un ensemble A dans E , associe la fonction caractéristique de \bar{A} dans l'ensemble des entiers.

Exemple :

```
# let non_divisible_par_trois = complement_dans_int divisible_par_trois;;  
val non_divisible_par_trois : int -> bool = <fun>  
# non_divisible_par_trois 7;;  
- : bool = true  
# non_divisible_par_trois 90;;  
- : bool = false
```

Vous pourrez vous servir de la fonction `not`, dont je vous laisse déterminer le rôle.

Exercice 8. Définissez `union : ('a -> bool) -> ('a -> bool) -> ('a -> bool)`, une fonction qui, partant de la fonction caractéristique d'un ensemble A dans l'ensemble des entiers et de la fonction caractéristique d'un ensemble B dans l'ensemble des entiers, construit la fonction caractéristique de $A \cup B$ dans l'ensemble des entiers.

Exemple :

```
# let divisible_par_trois_ou_pas = union non_divisible_par_trois  
divisible_par_trois;;  
val divisible_par_trois_ou_pas : int -> bool = <fun>  
# divisible_par_trois_ou_pas 6;;  
- : bool = true  
# divisible_par_trois_ou_pas 5;;  
- : bool = true
```

Vous pourrez vous servir de l'opérateur `||`, dont je vous laisse déterminer le rôle.

Exercice 9. * Définissez de même `singleton`.

Exemple :

```
# let singleton_3 = singleton 3;;  
- : int -> bool = <fun>  
# singleton_3 5;;  
- : bool = false  
# singleton_3 3;;  
- : bool = true
```

Quel est le type de la fonction `singleton` ?

Exercice 10. * Définissez de même l'intersection. Quel est son type ?