

Programmation fonctionnelle

Premiers pas avec OCaml

Notes

Pour cette séance de TD, pas de cours préalable. À vous de consulter la documentation et de prendre les initiatives nécessaires. Vous aurez besoin d'employer l'environnement d'OCaml (dont l'icône est probablement sur le bureau)

Vous pourrez trouver toute la documentation de OCaml sur le site <http://www.ocaml.org>.

En particulier, vous êtes invités à consulter :

- le manuel officiel d'OCaml, en anglais, qui détaille tous les aspects du langage, <http://caml.inria.fr/pub/docs/manual-ocaml/index.html> (regardez le chapitre 1)
- un livre en français sur le développement d'applications avec OCaml, <http://www.univ-valenciennes.fr/ROI/poirriez/ensuvhc/ocamluvhc/DA-OCAML/index.html> (regardez le chapitre 2).

Expériences

Exercice 1. Quelle est la différence entre les deux valeurs suivantes

```
print_newline
et
print_newline ()
?
```

Quel est le type de ces deux valeurs ?

Exercice 2. * (Chez vous) Téléchargez et installez

- OCaml pour Windows (version MinGW)
- la documentation d'OCaml au format `html`
- l'éditeur Scintilla Scite.

Exercice 3. * (Chez vous) Copiez, compilez et exécutez le programme suivant :

```
print_endline "Bonjour, le monde !";;
```

Exercice 4. Que déclare l'instruction suivante ?

```
let succi x = x + 1;;
```

Exercice 5. Que déclare l'instruction suivante ?

```
let succv = function x -> x + 1;;
```

Exercice 6. Que déclare l'instruction suivante ?

```
let succf x = x +. 1.;;
```

Exercice 7. Définissez une fonction `f` qui accepte deux arguments entiers et renvoie leur somme. Quel est le type de `f` ?

Exercice 8. Que déclare l'instruction suivante ?

```
let rec fac n =
  if n = 0 then
    1
  else
    n * ( fac ( n - 1 ) );;
```

À quoi sert le mot `rec` ? Que se passerait-il si vous l'omettiez ?

Exercice 9. * Considérons la fonction arithmétique g définie sur \mathbb{N} par $\forall n \geq 0, g(n) = \sum_{k=0}^n k$.

Nous allons implanter la fonction g en OCaml sous la forme d'une fonction `g`.

Quel sera le type de `g` ?

Définissez `g`. Vous pourrez vous inspirer de la fonction définie dans l'exercice 8.

Exercice 10. * Que déclare l'instruction suivante ?

```
let rec is_odd n =
  if n = 0 then
    true
  else
    not (is_odd (n - 1));;
```

Note: On aurait pu écrire cela de manière plus concise sous la forme

```
let rec is_odd n = (n = 0) || not (is_odd (n - 1));;
```

Cette deuxième formulation peut presque se traduire directement en une définition mathématique.