

Interprétation et compilation

Chapitre 1 Introduction à OCaml



Pablo Rauzy <pr@up8.edu>
`pablo.rauzy.name/teaching/ic`

Introduction à OCaml

- ▶ Créé en 1996 faisant suite à Caml Light (début 90) et Caml (1985).
- ▶ Langage fonctionnel se voulant pragmatique (permet l'impératif et l'objet).
- ▶ Typage statique fort et inférence de type, types algébriques.
- ▶ Dispose d'un interpréteur (`ocaml`), d'un compilateur bytecode (`ocamlc`), et d'un compilateur natif (`ocamlopt`).

Jouons un peu avec le langage

- ▶ Vous savez déjà programmer et avez déjà eu un cours de programmation fonctionnelle.
- ▶ Voyons ensemble la syntaxe et les spécificités d'OCaml :
 - valeurs et types de bases ;
 - expressions, variables, et inférence de type ;
 - fonctions et curryfication ;
 - listes et fonctions récursives ;
 - filtrage (*pattern matching*) ;
 - types algébriques (ADT) et filtrage ;
 - types rékursifs et paramétrés ;
 - filtrage avec garde ;
 - inférence de type et filtrages non-exhaustifs ;
 - enregistrements et déclarations directs dans les ADT ;
 - modules “fichiers”, modules “déclarés”, et foncteurs.

- ▶ Site et documentation : <https://ocaml.org/>
 - bibliothèque standard (et les alternatives Batteries et Core).
- ▶ Autres sources de documentation :
 - OCamlverse : <https://ocamlverse.net/>
 - *OCaml Programming: Correct + Efficient + Beautiful* : <https://cs3110.github.io/textbook/cover.html>
- ▶ Gestionnaire de paquets `opam` : <https://opam.ocaml.org/>.
- ▶ Moteur de production `ocamlbuild` ou `dune`.
- ▶ REPL `utop`.
- ▶ IDE dans Emacs avec `tuareg` et `merlin` (alternative : serveur LSP).
- ▶ Autres outils : `ocamldebug`, `ocamlprof`, `ocamldoc`, `ocamldep`, ...

- ▶ On utilise la bibliothèque standard de OCaml, les utilitaires `ocamllex` et `ocamlyacc`, `utop`, et `dune`.
- ▶ Pour commencer, installez `opam` :
 - `sudo apt install opam`
 - `opam init`
- ▶ Ensuite, la dernière version de OCaml (`ocamllex` et `ocamlyacc` seront automatiquement installés) :
 - `opam switch create 5.4.0`
- ▶ Puis, le REPL `utop` (`dune` sera automatiquement installé) :
 - `opam install utop`
- ▶ Enfin, optionnel mais conseillé, `ocaml-lsp-server` :
 - `opam install ocaml-lsp-server`

Un “vrai” projet OCaml

- ▶ Étudions ensemble un projet petit mais complet écrit en OCaml.
- ▶ Il s’agit un compilateur pour un langage très simple d’expression booléenne.
- ▶ Il compile vers un petite machine virtuelle simple.
- ▶ Au passage, il s’agit d’un exemple simplifié du type de chose que l’on va apprendre à faire dans ce cours.
- ▶ <https://code.up8.edu/-/snippets/3>