

# Quel Caml pour l'option informatique ?

Yann Salmon

Séminaire Algorithmique et Programmation  
C.I.R.M., Luminy, Marseille  
3 mai 2016

# Quel Caml pour l'option informatique ?

Yann Salmon

Séminaire Algorithmique et Programmation  
C.I.R.M., Luminy, Marseille  
3 mai 2016

# Quel Caml pour l'option informatique ?

Yann Salmon

Séminaire Algorithmique et Programmation  
C.I.R.M., Luminy, Marseille  
3 mai 2016

# Origines de Caml Light et OCaml

- Années 1980** Categorical Abstract Machine Language  
Compilation vers du Lisp interprété dans une machine virtuelle.  
Lent
- 1990** Caml Light  
Implémentation basée sur un noyau écrit en C  
Rapide et léger.
- 1996** Objective Caml  
Basé au départ sur Caml Light, ajoute des objets
- 2002** Dernière version de Caml Light  
Objective Caml devient OCaml
- Années 2000–2010** Nombreuses améliorations à OCaml

## Situation actuelle

- Le programme (arrêté) laisse le choix du langage, au sein d'une liste
- Une note de service réduit cette liste à Caml Light
- Certains lycées travaillent en Caml Light, d'autres en OCaml
- De nombreux énoncés de concours disent tout simplement Caml
- Les rapports révèlent des positions peu claires et hétérogènes, ou ne disent rien

# Différences OCaml — Caml Light

**char** 'a' au lieu de ‘a‘

**Modules** `List.iter` au lieu de `list__it`

**Noms de variables** Minuscule initiale obligatoire

**Booléens** `&&` et `||` au lieu de `&` et `|` et `or`

**streams** [`<...>`] plus dans le langage, mais via CamlP4

**Bibliothèque standard** Changements de noms

**Chaines** Immuables

# Différences OCaml — Caml Light

**char** 'a' au lieu de ‘a‘

**Modules** List.iter au lieu de list\_\_it

**Noms de variables** Minuscule initiale obligatoire

**Booléens** && et || au lieu de & et | et or

**streams** [<...>] plus dans le langage, mais via CamlP4

**Bibliothèque standard** Changements de noms

**Chaines** Immuables

**Transition très facile !**

# Intérêt d'utiliser OCaml

Stratégiquement et logistiquement

- Difficile d'expliquer aux étudiants qu'on travaille avec un langage mort.
- OCaml est largement utilisé et dispose d'une communauté dynamique.
- Il devient relativement facile à installer.
- On peut en faire en ligne <https://try.ocamlpro.com/>

# Intérêt d'utiliser OCaml

## Pédagogiquement

- Des efforts de convivialité.
- Des nouveautés intéressantes.
- Des branches expérimentales prometteuses.

# Intérêt d'utiliser OCaml

Pédagogiquement

Extension d'Arthur Chaguéraud :

<http://chagueraud.org/teach/ocaml/>

<https://github.com/chaguer/ocaml>

opam switch 4.02.0+improved-errors

## Quelle évolution demander ?

- OCaml comme langage pour l'option informatique
- Maintenir la possibilité de Caml Light pour 4 sessions (plus ?)
- Cerner plus précisément les éléments au programme

# Quelle évolution demander ?

Cerner les éléments au programme

## Oui

- Syntaxe de base
- `Array.make`, `Array.make_matrix`
- `Array.copy`
- `List.hd`, `List.tl`

## Non

- Modules
- Classes, objets
- `Array.sub`
- `List.nth`
- `@`
- `Hashtbl`, `Map`, `Set`, ...

# Quelle évolution demander ?

Cerner les éléments au programme

???

- Exceptions
- `Array.init`, `Array.map` `Array.copy` (copie de matrice)
- `List.append`
- `List.iter`, `List.map` ?

# Liens

- Une réclame pour l'option info [http://pro.yannsalmon.fr/opt\\_present](http://pro.yannsalmon.fr/opt_present)
- Les succès de OCaml (plein de logos)  
<http://ocaml.org/learn/companies.html>  
<http://ocaml.org/learn/success.html>
- Ressources pour l'enseignement  
<https://ocaml.org/learn/teaching-ocaml.html>
- OCaml en ligne <https://try.ocamlpro.com/>  
<http://andrewray.github.io/iocamljs/min402.html>
- Extension d'Arthur Chaugéraud : <http://chaugeraud.org/teach/ocaml/>  
<https://github.com/chauger/ocaml>  
ou via opam : `opam switch 4.02.0+improved-errors`  
Lancer avec `ocaml -easy`.