

# La Programmation Fonctionnelle

"ML, Lisp, and Haskell are the only programming languages that I've seen where one spends more time thinking than typing." - Philip Greenspun



# Plan

Différentes façons de programmer

Qu'est-ce que la PF ?

Et dans la pratique ... ?

Avantages et inconvénients

Démos LIVE from La Boate : cace-dédi au

PLUG

# Différents paradigmes

Principalement :

- Programmation impérative
  - Procédurale (C, Fortran)
  - Objet (Java, C++, C#)
- Programmation déclarative
  - Fonctionnelle (Haskell \o/)
  - Logique (Prolog)
- Et bien d'autres...

# Programmation impérative

Procédurale

Fonctions

Structures

... et c'est tout

Objet

Classes

Méthodes

Fonctions libres

→ On modifie des variables, tout le temps !

→ On spécifie chaque instruction !

# Programmation déclarative

- Fonctionnelle
    - Fonctions, fonctions, fonctions
    - Facilités pour les faire interagir
    - Pas d'instructions, que des expressions
  - Logique
    - Prédicats, prédicats, prédicats
    - Moteur d'inférence logique
- Effets de bords limités, tout ce que l'on écrit est une valeur, un "calcul", pas un "ordre"

# Essence de la PF

On peut définir, mélanger, composer [...] des fonctions

```
wc_linenb filename = length . lines . readFile filename  
add_n_to_list n list = map (\x → x+n) list
```

Selon les langages, on peut définir des *types de données algébriques*

```
data Direction = Up | Down | Left | Right  
data Number = Positive Int | Zero | Negative Int
```

On peut se servir de ces types efficacement :  
pattern matching

*is\_positive (Positive n) = True*  
*is\_positive (Zero) = True*  
*is\_positive (Negative n) = False*

On a les clôtures (transparence référentielle) :

*x = 4*  
*f y = x+y* – c'est une fonction  
*x = 5*  
*f 1* – on appelle la fonction  
*cela retourne ... 5, et pas 6*

# Essence de la PF

- Finie la gestion de la mémoire
- On applique des fonctions plutôt que de modifier des variables



## Et dans la pratique ... ?

Langages de PF populaires :

- Lisp : vieux, puissant mais laborieux, communauté assez restreinte
- Objective Caml : pas tellement vieux, rapide, autorise l'impératif, communauté correcte, nombre respectable d'outils & bibliothèques
- Haskell : pas très vieux, purement fonctionnel, communauté hyperactive, énormément d'outils & bibliothèques !
- Erlang, Clojure, Scheme, F#, ...
  - généralement opensource, comités pour l'évolution, projets communautaires, ...

## Et dans la pratique ... ?

- Souvent interpréteur interactif (Ocaml, Haskell notamment)
- Souvent compilateur vers bytecode, donc avec une VM – souvent munie d'un garbage collector
- Quelques compilateurs vers du code natif (→ ASM x86\_64 puis compilation de l'ASM) pour les performances
- Souvent une Foreign Function Interface pour appeler des fonctions C depuis le langage directement

## Et dans la pratique ... ?

- Soit statiquement typé (OCaml, Haskell), soit dynamiquement typé (Lisp, Scheme)
- On fait ressortir autant de choses que possible dans les types (Maybe, ...)
- Les types sont généralement inférés automatiquement
- Une fois que le code est testé, chances infimes que “cela plante en route”
- La récursion remplace les boucles classiques (for, while & co), et l'on a des primitives de récursions (“schémas” de récursion qui permettent presque de tout faire)

# Avantages

- Le compilateur facilite beaucoup le travail
- Une fois la compilation passée, peu de risques de dysfonctionnement
- Un programme fonctionnel est presque un ensemble de phrases, on exprime ce qu'on veut et pas chaque étape → plus proche des langues humaines
- Limiter voire interdire les effets de bord limite les surprises à l'exécution : 2 appels avec les mêmes arguments renvoient la même chose → et bien d'autres avec le temps...

# Inconvénients

- Très (très très très) différent des langages impératifs, qui sont les plus utilisés & connus
- Desfois, les effets de bord, ça aide ...
- Manque de bibliothèques / outils pour “rivaliser” avec des grosses technos (Java, .NET, etc)
- Trop peu de boîtes qui utilisent les langages fonctionnels (même si dedans il y a Intel, MS, IBM, des boîtes de finances, l'aéronautique, aérospatial, ...)

# Démo LIVE

Qui reconnaît ?

