

# Sensibilisation à la programmation fonctionnelle avec Haskell

Rémi Douence

Mines de Nantes

# Objectifs du cours

- les pieds sur terre. . .
  - ne pas former des ingénieurs jetables (ne pas être très vite dépassé)
  - former des ingénieurs qui peuvent même avoir une longueur d'avance (dépasser les "autres")
  - sensibiliser aux innovations : pour que vous continuiez à rester devant (continuer à avancer, vous (auto)former pendant votre carrière)
- ... et la tête dans les étoiles : beauty is my business, une esthétique de la programmation.
- Très concrètement
  - présenter un langage donc les concepts sont en train de basculer de recherche vers industrie
  - offrir des concepts que vous retrouverez demain en position d'ingénieur ou de décideur
  - faire de vous de meilleurs programmeurs C/Java/Cobol. . . dès aujourd'hui

# Quelques caractéristiques de Haskell

Quelques caractéristiques de Haskell :

- Haskell est (fortement et statiquement) typé : bcp bcp bcp de problèmes réglés avant l'exécution
- Haskell est pur (ou plutôt très structurant : le typage isole les bulles fonctionnelles et impératives, bcp bcp bcp de problèmes réglés avant l'exécution)
- Haskell est d'ordre supérieur (généraliser et régénéraliser pour la réutilisation)
- Haskell est paresseux (écriture modulaire et colle)
- Haskell propose des modèles de concurrence innovants (stratégies map-reduce, mémoire transactionnelle)
- Haskell est plein d'autres choses mais pas le temps de voir (déjà on ne parlera presque pas d'impératif ni de concurrence).

- The Pragmatic Programmer

http:

`//pragprog.com/the-pragmatic-programmer/extracts`



- The Passionate Programmer (2nd edition): Creating a Remarkable Career in Software Development

`http://pragprog.com/book/cfcar2/`

`the-passionate-programmer`



- Je vais vous apprendre à parler portugais.
- En moins de 20h.
- Jamais vous n'ouvrirez la bouche pendant la formation.
- Par contre vous regarderez des séries françaises à la télé.
- Ça sera du portugais pour écrire des actes notariés.

- Je vais vous apprendre à monter à cheval.
- En moins de 20h.
- Jamais vous n'approcherez un cheval pendant la formation.
- Par contre vous téléphonerez à tous vos amis.
- Ça sera pour faire des concours de dressage.

- Je vais vous initier à la programmation fonctionnelle.
- En moins de 20h.
- Vous allez lire du code, écrire du code et réécrire du code.
- Et en plus vous allez prendre des notes (avec un crayon).
- Ça sera juste une sensibilisation mais vous saurez faire choses.

- Mindset <http://coreymandell.net/blog/screenwriting-advice/the-right-mind-set/>





# Lectures (si vous avez envie)

- Real World Haskell

<http://book.realworldhaskell.org/read/>



- A gentle introduction to Haskell 98 <http://www.haskell.org/tutorial/haskell-98-tutorial.pdf>



- Introduction to Functional Programming using Haskell



- Google. . .

- 5 blocs =  $5 * (1C + 2TP) = 15$  seances =  $15 * 1.25 = 18.75h$
- même pas vrai : 1TP en moins (le 4.2) et un cours en moins (le 5) :-(
- Cours = je programme, vous notez + peut être des exercices sur feuille
- TP = vous programmez, je vous aide à rester dans les clous du langage
- Evaluation : sur clavier au bloc 5

- programmer n'est pas sale
- Yves Caseau
- entraineur sportif / manager ce que l'on ne connaît pas / il y a ceux qui savent faire et ceux qui enseignent

- haskell platform <http://hackage.haskell.org/platform/>
- bookmarquer la page de doc
- emacs (mac os x aqua-emacs <http://aquamacs.org/>)
- interprète ghci dans un shell

- ghci une calculatrice (question - réponse)
- écrire des définitions dans un fichier
- démarrage : `ghci TP1ACompleter.hs`
- reload : `:r`

- split horizontal : CTRL + x ; 2
- change buffer : CTRL + x ; o
- run shell : ESC ; x ; shell ; ENTER
- save : CTRL + x ; CTRL + s
- couper : (CTRL + k)\*
- coller : CTRL + y
- etc...