

Programmation Fonctionnelle: Conclusion

Adrien Durier

13 septembre 2023

Sources

- Slides & supports de cours :

- Pablo Aghiri
- Kim N'Guyen
 - lri.fr/~kn/ipf_en.html
 - lri.fr/~kn/ppfa_en.html
- Sylvain Conchon
 - lri.fr/~conchon/IPF/

- Ouvrages :

- Apprendre à programmer avec OCaml
Sylvain Conchon, Jean-Christophe Filliâtre
- OCaml Programming : Correct + Efficient + Beautiful
Michael R. Clarkson

Conclusion

- **Avantages Généraux**

- **Immutabilité** : Réduit les bugs
- **Réutilisabilité** : Composabilité
- **Transparence référentielle** : Facilite les tests
- **Concurrence** : Plus sûre

- **Avantages Généraux**

- **Immutabilité** : Réduit les bugs
- **Réutilisabilité** : Composabilité
- **Transparence référentielle** : Facilite les tests
- **Concurrence** : Plus sûre

- **Avantages OCaml**

- **Performance** : Proche de C/C++
- **Types forts** : Moins d'erreurs
- **Interopérabilité** : Utilisable avec C

- **Avantages Généraux**

- **Immutabilité** : Réduit les bugs
- **Réutilisabilité** : Composabilité
- **Transparence référentielle** : Facilite les tests
- **Concurrence** : Plus sûre

- **Avantages OCaml**

- **Performance** : Proche de C/C++
- **Types forts** : Moins d'erreurs
- **Interopérabilité** : Utilisable avec C

- **Applications Réelles en OCaml**

- **Infer** : Analyse statique (Facebook)
- **Astrée** : Systèmes embarqués
- **Coq** : Assistant de preuve