





**LES LANGAGES MATHS**

CONVINCENOS A PRESENTARLES

Haskell + Java => Scala

Haskell + Js => Purescript ou Elm

OCaml + Ada + C++ => Rust

OCaml + C# => F#

OCaml + Java => Kotlin

OCaml + ruby => Swift

OCaml + Js => ReasonML ou Rescript

C# + Js => Typescript

Erlang + ruby => Elixir

Lua + C + Ada => Go

C# + common Lisp + Java => Clojure

MAGNET 2-DUALS - TRANSLES

ML et Haskell ont beaucoup influencé les évolutions  
des langages ces dernière années 👍

# LES LANGAGES MAINSTREAM

## COMPOSEZ VOS APPRENTISSAGES

Haskell + Java => Scala

Haskell + Js => Purescript ou Elm

OCaml + Ada + C++ => Rust

OCaml + C# => F#

OCaml + Java => Kotlin

OCaml + ruby => Swift

OCaml + Js => ReasonML ou Rescript

C# + Js => Typescript

Erlang + ruby => Elixir

Lua + C + Ada => Go

C# + common Lisp + Java => Clojure

ML et Haskell ont beaucoup influencé les évolutions des langages ces dernière années 👍



# LES LANGAGES RÉSOLVENT DE PROBLÈMES

## QUELLES SONT VOS PRIORITÉS

Être résilient par compilation : Haskell, OCaml, F#, Scala, Rust ... dans une certaine mesure Kotlin, Swift

Être résilient par parallélisation : Erlang / Elixir

Valider les états (ADT) : Kotlin, Scala, F#, Swift, Rust, OCaml, Haskell

Valider les transitions (GADT) : OCaml, Haskell, Scala

Écrire des DSL génériques (Higher Kinded Types) : Haskell, Scala

Faire de l'embarqué : Go, Rust, OCaml

Faire des jeux vidéos AAA+ : C#, C++ ; AA+ : Lua, Haxe, Javascript, Rust

Faire du calcul parallèle : Rust, C++, Erlang/Elixir

Faire du micro-frontend : F#, C#, Elixir, Typescript

Faire un OS : C, OCaml, Rust

Faire une blockchain : OCaml, Haskell, Rust

Faire de la data science : python, julia, lua, C++, swift

Faire du prototype rapide : javascript, Ruby

« Tournevis multifonction » : F#, Kotlin, OCaml ... Tout langage avec un bon compilateur Javascript et FFI C

MIAGE M2 - QUALITÉ DU SI - THOMAS HAESSLÉ