

Partie 1

Contexte & Historique

Aux origines

Erlang chez Ericsson (1980s)

- Créé chez Ericsson pour les systèmes télécom robustes, tolérance aux pannes, "run forever"
- VM BEAM + OTP : processus légers, supervision, distribution
- Passage en open-source en 1998

ERICSSON REVIEW

2 1993

RACE Open Services Architecture
Erlang – A new Programming Language
New Technology for Prototyping New Services
Prototyping Cordless Using Declarative Programming
Low-Loss Splicing Technique for Erbium-Doped Optical Fibre Amplifiers



José Valim

Créateur d'Elixir

- Initialement développeur Ruby, frustré par les limites du langage concernant l'exécution concurrente
- Crée Elixir en 2012 : syntaxe moderne, métaprogrammation, tooling meilleur
- Compilé sur la VM BEAM → interopérabilité avec Erlang



Adoption croissante

- Un des langage les plus appréciés par ses développeurs
- Commence à être utilisé chez Apple, Google et d'autres grandes entreprises
- Discord : chat en temps réel à grande échelle, backend massif reposant sur Elixir
- Autres usages : IoT, APIs hautes-disponibilité, plateformes concurrentielles

Raisons d'apprendre Elixir

- Découvrir une autre façon de programmer: la programmation *fonctionnelle*
- Découvrir la programmation concurrente

Elixir fera de vous de meilleurs programmeurs dans l'absolu, y compris en Python et dans tout autre langage que vous explorerez car il vous apprendra à "penser" comme un informaticien

Partie 2

Syntaxe & Premiers pas

Prérequis

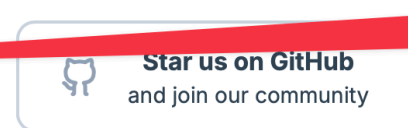
installer LiveBook

visitez <https://livebook.dev>



Automate **code** & **data** workflows
with interactive notebooks

Get rid of scripts, manual steps, and outdated docs. Use Elixir and Livebook to share knowledge, deploy apps, visualize data, run machine learning models, debug systems, and more!



Install Livebook in just a minute

Livebook is open source, free, and ready to run anywhere.



Run on your machine
with Livebook Desktop



Mac (Universal)



Windows



Run in the cloud
on select platforms



Run on Hugging Face

LiveBook ?

- c'est le "Jupyter d'Elixir"
- nous allons l'utiliser pour illustrer ce cours
- ... et vous découvrirez un outil *beaucoup plus ergonomique* que Jupyter Labs 😊



Pour l'instant, je vais vous montrer comment l'utiliser, puis ce sera votre tour !

Premier contact avec Elixir

Comme le veut la tradition, écrivons un exemple de "Hello World" en Elixir:

```
IO.puts("Hello, world !")
```

ce qui donne, dans LiveBook:

```
1 IO.puts("Hello, world !")
```

Evaluated

Hello, world !

:ok

Premier contact avec Elixir

(suite)

```
IO.puts("Hello, world !")
```

- nous avons appelé la fonction `puts/1`
- du module `IO`
- en lui passant en paramètre la chaîne de caractères `"Hello, world"` (les données)

*Vous avez sous les yeux tous les éléments d'un programme Elixir: une suite de **fonctions**, groupées en **modules** auxquelles on passe des **données** et qui retournent un résultat*

Structure d'un programme