

# Métaprogrammation en Elixir

*Une introduction naïve*

Xavier Van de Woestyne

`xaviervdw@gmail.com` - *margo.com*

**Paris.ex 12** - Mai 2019

# Bonjour/Bonsoir !

Xavier Van de Woestyne

- Bruxelles, Lille, Paris
- *Data Engineer* chez Margo Bank

## Réseaux

- <https://xvw.github.io>
- *vdwxv* sur Twitter
- *xvw* sur Github/Gitlab
- *xvw@merveilles.town* sur Mastodon

OCaml, F#, Erlang/Elixir, Kotlin, Io, Ruby, Elm, Racket

## LilleFP

- Meetup régulier
- Langages applicatifs
- Programmation fonctionnelle
- Systèmes de types
- Fusion de LilleFP, Lille Elixir, Lille Rust et LilleSUG
- **On recrute des speakers !**

# Après presque 10 ans de Erlang

*J'appartiens aux gens qui n'ont pas été particulièrement emballé par Elixir*

- Je préférais la syntaxe de **Erlang** (amoureux de **Prolog**) ;
- j'étais inquiet de la transmissions d'idiomes (Ruby) ;
- donc je trouvais que Elixir ne servait à rien (à tort...).

## Ce qui m'a fait changé d'avis sur Elixir

- **Phoenix** (plus spécifiquement, **Ecto** et **Plug**) ;
- l'unification des types pour les chaînes de caractères ;
- les mécanismes de méta-programmation ;
- **mix** ;
- et plus tard : `gen_stage` et `Flow`.

# Sommaire

- Comprendre ce qu'est *rapidement* la méta-programmation ;
- survoler les perspectives qu'Elixir offre pour *méta-programmer* ;
- observer quelques cas d'usages aux **macros** ;
- conclure sur les usages et avantages.

*Ceci est introduction naïve ... pour les non-initiés*

# Meta-programmation

*L'écriture de programme qui manipulent des données **décrivant** elles-mêmes des programmes. (d'où le "meta")*

# Meta-programmation

On peut méta-programmer de plusieurs manières

- Via la réflexion (implicite ou explicite) ;
- via des raccourcis du langage ;
- via des générateurs de code (par exemple les affreux templates de C++) ;
- via des étapes de compilation (multi-staged metaprogramming) ;
- via des macros ;
- en utilisant un Lisp (aha !)

## A quoi ça sert ?

- Limiter le **boilerplate** ;
- limiter le **boilerplate** ;
- limiter le **boilerplate** ;
- étendre le langage (eDSL) ;
- déléguer à une machine des tâches rémanantes.

*Elixir offre plusieurs outils de méta-programmation !*



Introspection via `Module.__info__(subject)`

```
iex> Map.__info__(:functions)
[delete: 2, drop: 2, ... values: 1]
```

*Dispatch* dynamique

```
def call(module) do
  apply(module, function, [args])
end
```

# Génération de code et dérivation d'interfaces

- **Génération:** Behaviour
- **Dérivation:** Protocol

*Ce sont les manières rapides de faire de la meta-programmation.*

# Pré-processeur et macros

On entre enfin dans le vif du sujet !