

# Introduction à Elm



Raphaël Pallies, Xavier Van de Woestyne

LilleFP3

## Intro : A supprimer

- ▶ Web et JS (ici on troll React and co, complexité)
- ▶ Elm : Un langage pour le web (Types, FRP)
- ▶ Let it be Mainstream
- ▶ Elm Architecture
- ▶ Modules

# Installation

- ▶ Via des *installers* (site officiel) ;
- ▶ via NPM :

```
npm install -g elm
```

L'installation offre :

- ▶ `elm-repl` ;
- ▶ `elm-reactor` ;
- ▶ `elm-make` ;
- ▶ `elm-package`.

# Elm-repl

*Une boucle interaction pour tester des expressions Elm rapidement.*

```
---- elm-repl 0.18.0 -----  
:help for help, :exit to exit, more at ...  
-----  
> 1 + 1  
2 : number  
> List.foldl (+) 0 [1, 2, 3, 4, 5]  
15 : number  
>
```

# Elm-reactor

- ▶ Aide à la construction d'un projet Elm ;
- ▶ permet de lancer un serveur de test ;
- ▶ permet d'explorer les fichiers Elm ;
- ▶ recompile à chaque actualisation.

## Un Hello world en Elm

```
import Html exposing (text)
main =
    text "Hello, World!"
```

# Utilisation de la Elm-architecture

## Programme

*Un programme cristallise la Elm-architecture, il représente le cycle de vie de l'application et son mécanisme de communication.*

## Beginner programme

```
beginnerProgram :  
  {  
    model : model,  
    view : model -> Html msg,  
    update : msg -> model -> model  
  } -> Program Never
```

- ▶ Un programme est généralement le point d'entrée d'une application ;
- ▶ il existe d'autre programme plus fins.

*Le `beginnerProgram` exprime une idée simple de la Elm-architecture. Chaque interaction (clique par exemple) du programme produit **un message**. Le message est envoyé à **`l'update`** qui, sur base du **modèle** courant, produit un nouveau modèle. **La vue** se régénère sur base du nouveau modèle. Ces actions se répètent indéfiniment.*

## Représentation du HTML avec des fonctions

- ▶ `tag [attr_list] [children_list] : tag normal ;`
- ▶ `text "mon texte" : noeud textuel (PCDATA).`



# Live-coding !

- ▶ Un compteur réactif,
- ▶ une liste de présentations !

*Gros gros challenge, effet **WHAOU** garanti.*

## History : déboguer vos programmes Elm

- ▶ Permet de rejouer des états de l'application ;
- ▶ permet d'observer le modèle à des états donnés ;
- ▶ permet d'importer/exporter des collections d'états. (Génial pour demander (ou prodiguer) de l'aide !)

# Ecosystème et communauté

- ▶ Elm-package : gestionnaire de paquet de Elm ;
- ▶ Awesome-elm : liste de ressources Elm ;
- ▶ beaucoup de tutoriels/exemples ;
- ▶ beaucoup d'intégrations dans des éditeurs ;
- ▶ Elm-test ;
- ▶ Elm-format (un peu extrême) ;
- ▶ Elm-router, Elm-Lazy etc.

# Outro : a supprimer

- ▶ Point forts (+héritage)
- ▶ Points faibles (modules)
- ▶ Future (PureScript)
- ▶ Conclusion