

# Choisir son langage, le dilemme du web

Par Philippe Roberge  
28 mai 2020



# elm

A delightful language  
for reliable webapps.

# Sur l'agenda

## **Historique et principes**

D'où vient elm?

Pourquoi c'est intéressant

## **Atelier 01**

Commencer un projet!

## **Atelier 02**

De la théorie... encore de la  
théorie

## **Période de questions**

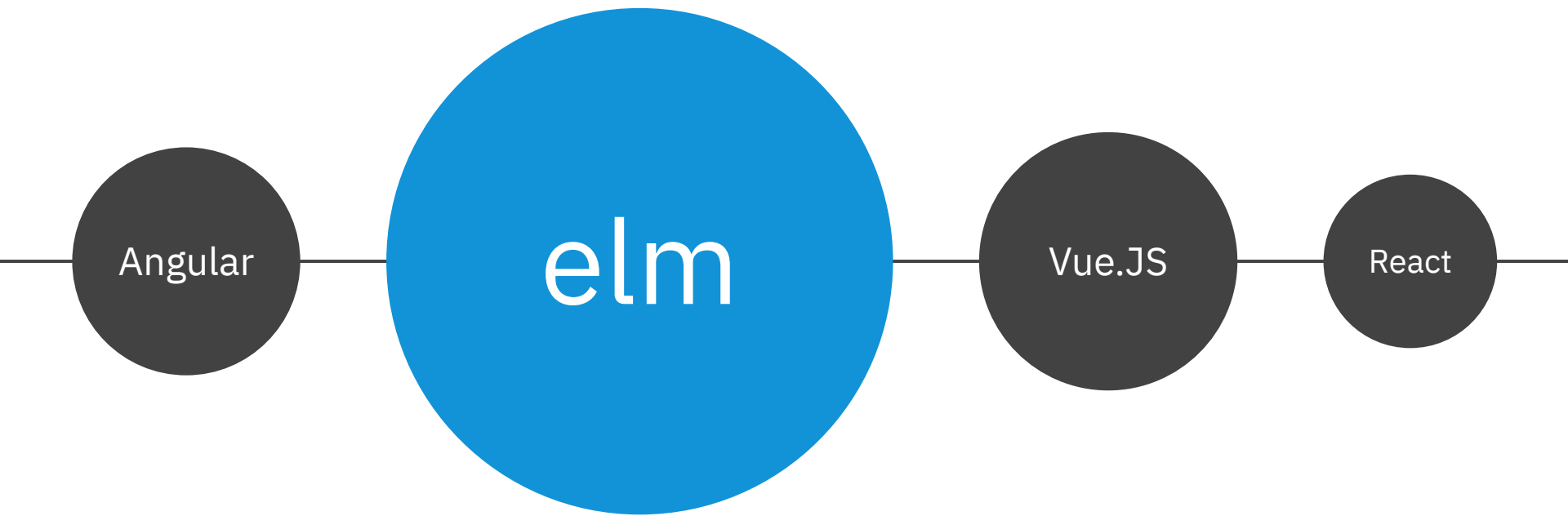
---

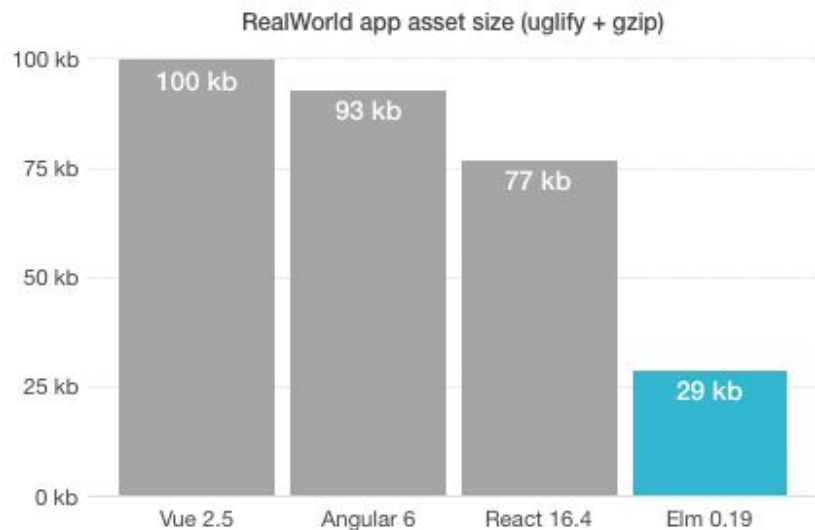
# À propos d'elm

---

- Créé en 2012 par Evan Czaplicki
- Orienté vers les interfaces graphiques
- Principes du langage:
  - « Model - View - Update »
  - Les abonnements
  - L'immuabilité des valeurs
  - Le typage statique
  - L'interopérabilité avec HTML, CSS et JavaScript

# Les technologies « Front-End »

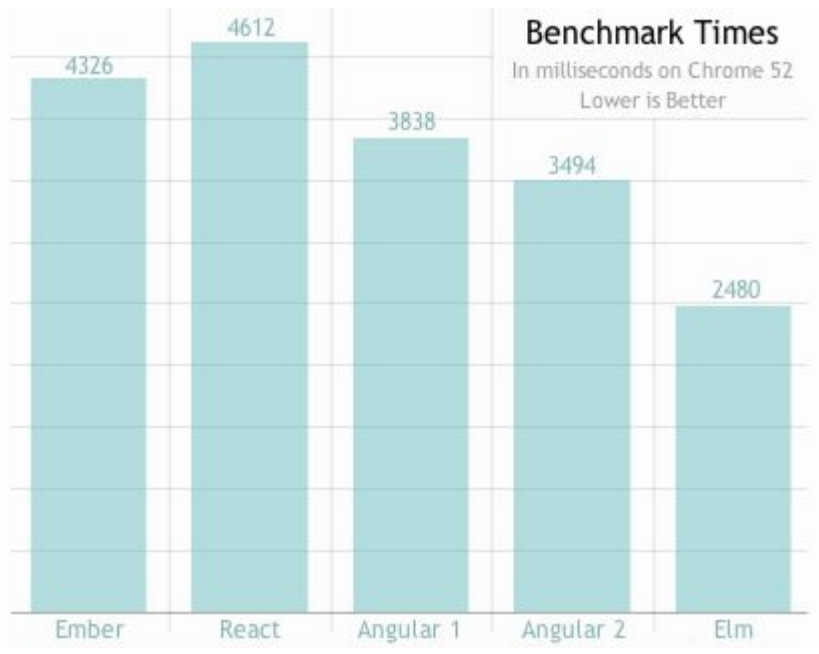




- Création d'un seul fichier
- Élimination du code mort
- Optimisation du compilateur

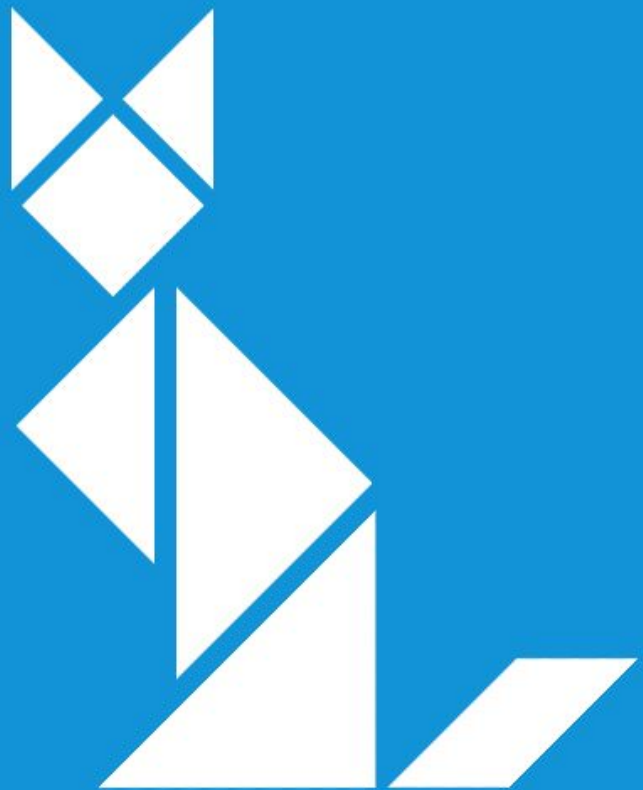
---

## Taille des fichiers sources



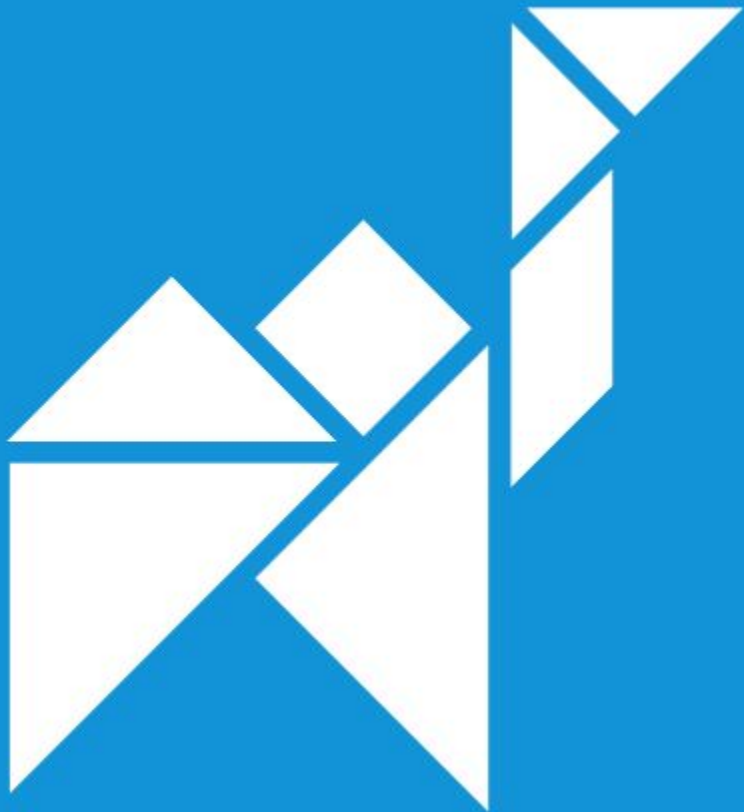
- Langage fonctionnel pure
- Optimisation aggressive du code
- Simplicité pour l'utilisateur

## Rapidité d'exécution



# Atelier 01





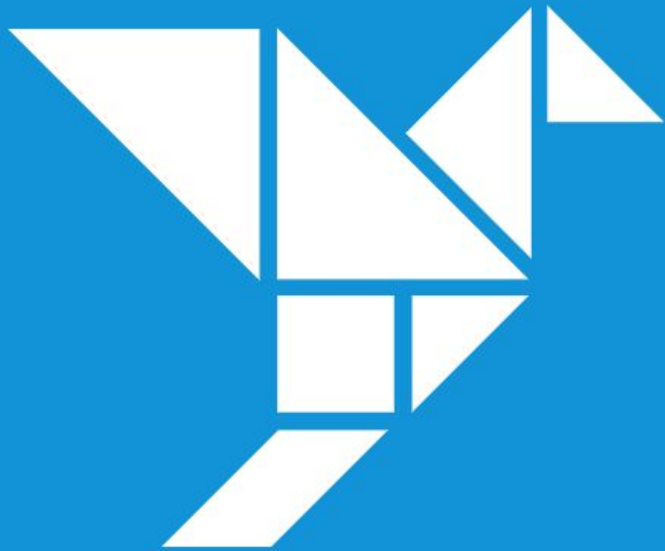
A delightful language  
with no runtime exceptions.

# Est-ce possible?

— — —

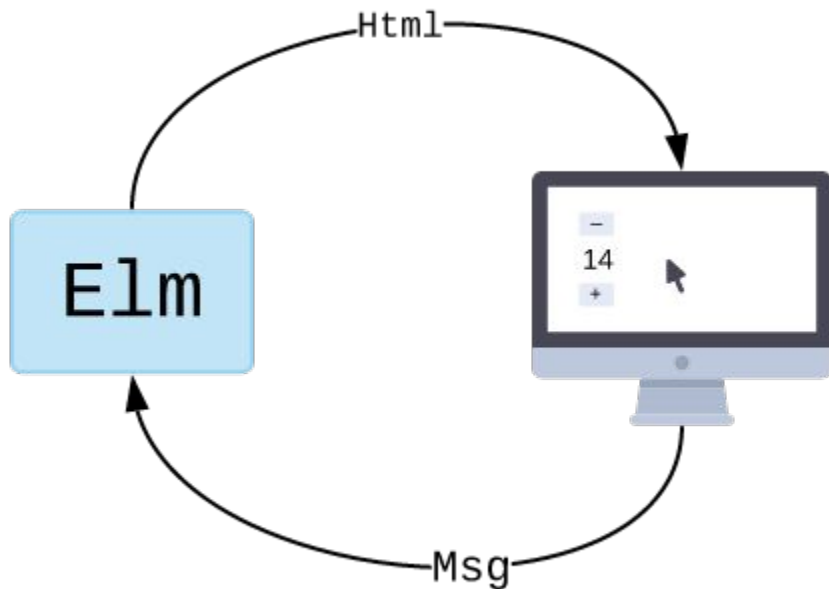
L'impossibilité d'avoir des variables nulles élimine les erreurs comme celle-ci.





# Atelier 02

# L'architecture de Elm



## **Model**

Structure de données représentant l'état de notre application

## **View**

Fonction qui génère du HTML avec le modèle

## **Update**

Fonction qui intercepte les actions du client pour changer le modèle

# En action

```
main =  
  Browser.sandbox  
    { view = view  
      , init = init  
      , update = update  
    }
```

## En action (suite)

```
init : Model  
init =  
| { login = ""  
| , password = ""  
| }
```

## En action (suite)

```
view : Model -> Html Msg
view model =
  div [ id "form" ]
    [ input [ type_ "text", value model.login, onInput ChangeLogin ] []
      , input [ type_ "password", value model.password, onInput ChangePassword ] []
      , button [ onClick Submit, disabled (isInvalidFormat model) ] [ text "Submit" ]
      , p [] [ text model.login ]
      , p [] [ text model.password ]
    ]
```

# En action (suite)

```
update : Msg -> Model -> Model
update msg model =
  case msg of
    ChangeLogin login ->
      { model | login = login }

    ChangePassword password ->
      { model | password = password }

    Submit ->
      model
```





## Conclusion

- Confiance
- Aisance
- Productivité



Evan, C. (s. d.). elm lang. Consulté le 22 mai 2020, à l'adresse <https://elm-lang.org/>

Wikipedia contributors. (2020, avril 19). Elm (langage). Consulté le 22 mai 2020, à l'adresse [https://fr.wikipedia.org/wiki/Elm\\_\(langage\)](https://fr.wikipedia.org/wiki/Elm_(langage))

## Bibliographie