TRY ELM

No runtime errors. Safe and fun refactors!

KATJA MORDAUNT

I LIKE MAKING STUFF THAT HELPS

ASSUMPTIONS I'M MAKING ABOUT YOU

- Fun
- Safety & efficiency
- Client trust
- New tech = fun vs. Old tech = safe ??

THIS TALK WILL NOT

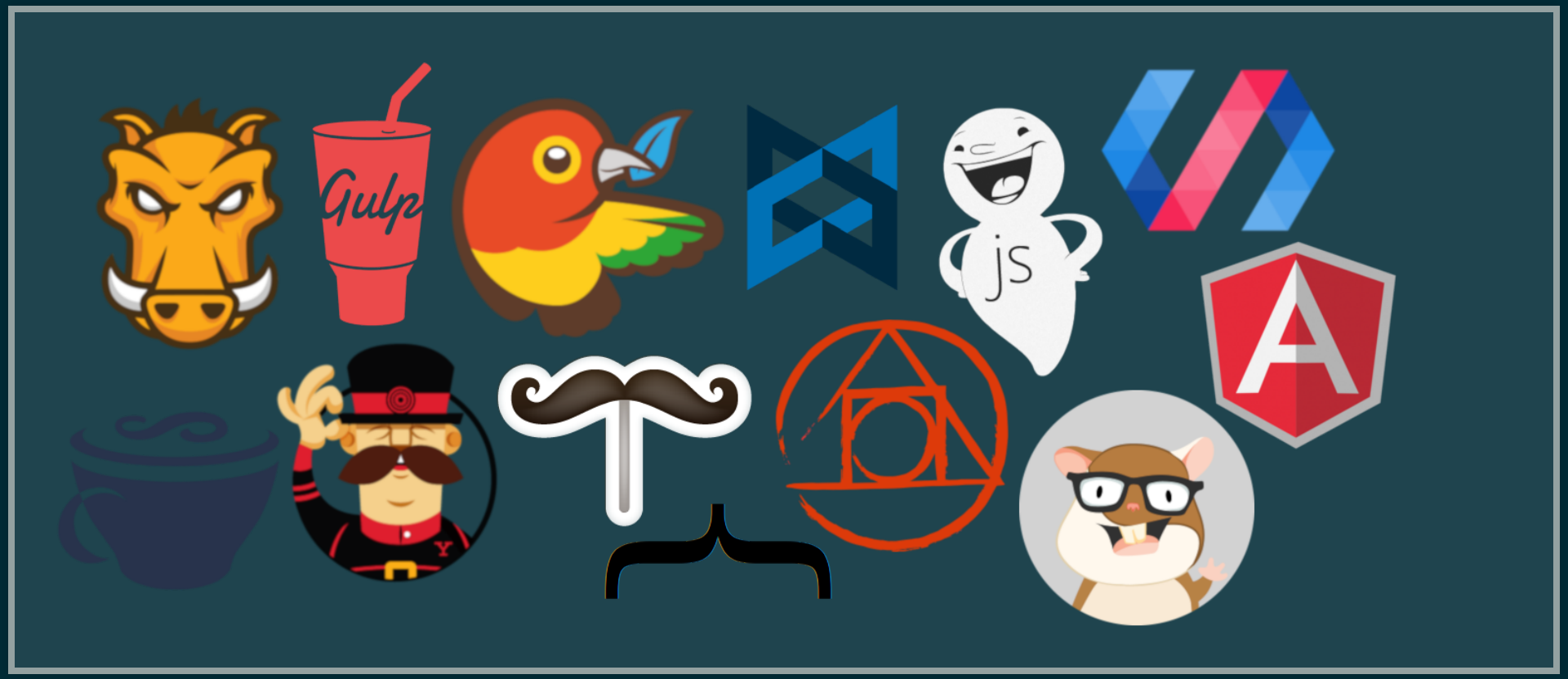
- Teach you functional programming concepts
- Compare Elm with React, Vue and Angular 2.0
- Discuss the newly released Elm 0.19

THIS TALK WILL

- Shine a little light on Elm and why I enjoy using it

OUR (FUN) PROBLEM

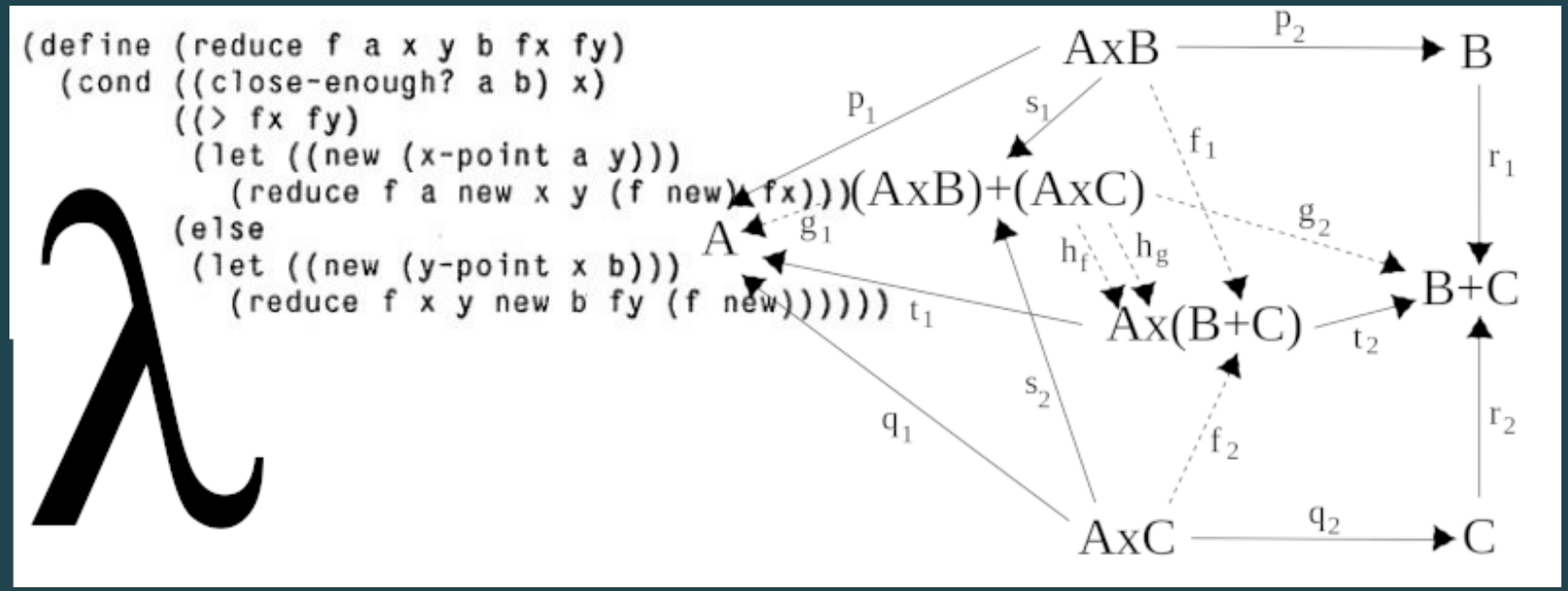
Any of these look familiar?



elm

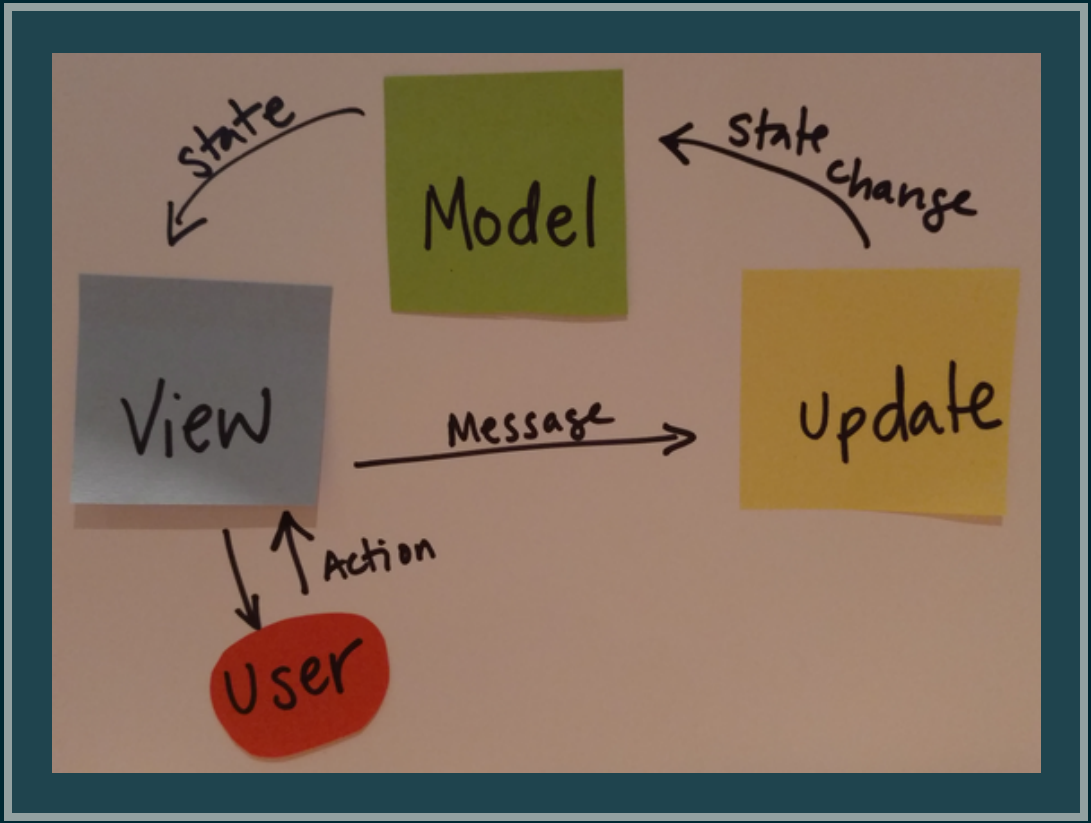
A delightful language for reliable webapps.

Generate JavaScript with great performance and no runtime exceptions.



- Uses ML style syntax
- Based on functional programming principles
- Statically typed
- Compiles to Javascript
- More than a language, defines an architecture that helps us write good code

ANATOMY OF AN ELM APP



SYNTAX

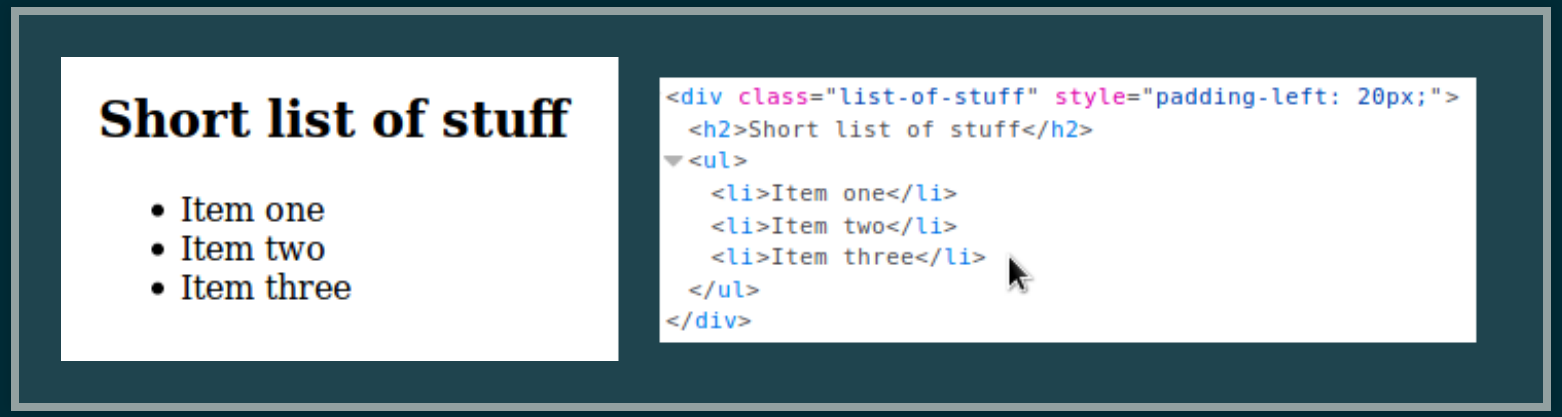
```
01  "Hello " ++ "Exeter!"
02  > Hello Exeter!
03
04  5 / 2
05  > 2.5
06
07  5 // 2
08  > 2
09
10  addTwoString : Int -> Int -> String
11  addTwoString x y =
12    toString x ++ "+" ++ toString y ++ "=" ++ toString (x + y)
13
14  addTwoString 2 3
15  > "2+3=5"
16
```

HTML?

```
01  div [ class "list-of-stuff" ]
02  [
03    h2 [] [ text "Short list of stuff"],
04    ul []
05      [
06        li [] [ text "Item one"],
07        li [] [ text "Item two"],
08        li [] [ text "Item three"]
09      ]
10  ]
```

HTML?

```
01  div [ class "list-of-stuff" ]
02  [
03    h2 [] [ text "Short list of stuff"],
04    ul []
05      [
06        li [] [ text "Item one"],
07        li [] [ text "Item two"],
08        li [] [ text "Item three"]
09      ]
10  ]
```



HTML?

divs and headings and lists

```
01  div [ class "list-of-stuff" ]
02  [
03    h2 [] [ text "Short list of stuff"],
04    ul []
05      [
06        li [] [ text "Item one"],
07        li [] [ text "Item two"],
08        li [] [ text "Item three"]
09      ]
10  ]
```

...and buttons and links etc

```
01  button [ onClick DoThing ] [ text "Do thing" ]
02  a [ href "/my-path" ] [ text "Follow the high road" ]
```

HTML?

USING ELM-FORMAT

```
01  div [ class "list-of-stuff" ]
02      [ h2 [] [ text "Short list of stuff" ]
03      , ul []
04          [ li [] [ text "Item one" ]
05          , li [] [ text "Item two" ]
06          , li [] [ text "Item three" ]
07          ]
08      ]
```

- elm-format -> One agreed standard
- Machine can parse with confidence
- No one forgets the commas
- Don't need trailing commas to eliminate bad diffs

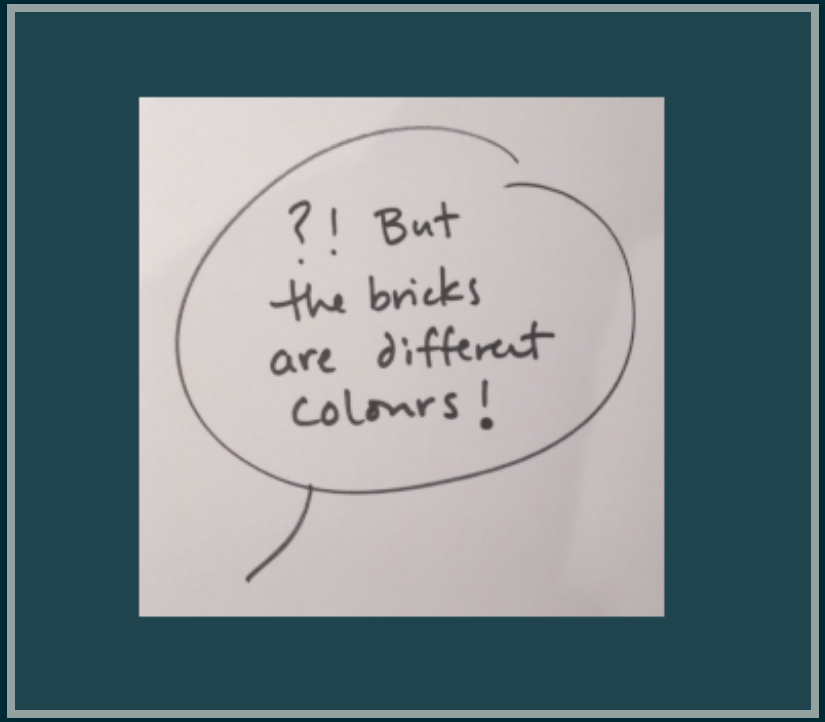
QUICK SCENARIO

AWESOME BRICK COUNTER!

```
01  module Main exposing (..)
02
03  import Html exposing (..)
04  import Html.Attributes exposing (class, style)
05  import Html.events exposing (onClick)
06
07  -- MODEL
08
09
10  type alias Model =
11      { bricks : Int }
12
13
14  model : Model
15  model =
16      { bricks = 0 }
```

AWESOME BRICK COUNTER!

SCENARIO UPDATE!



COMPILER LEAD DEVELOPMENT

CONFIDENT REFACTOR. 0 RUNTIME ERROR.

```
01  module Main exposing (..)
02
03  import Color exposing (Color, blue, brown, red)
04  import Html exposing (Html, button, div, h1, h2, li, text, ul)
05  import Html.Attributes exposing (class, style)
06  import Html.Events exposing (onClick)
07
08
09  -- MODEL
10  type alias Model =
11      { bricks : List Color }
12
13
14  model : Model
15  model =
16      {bricks = []}
```

OOPS!

I FORGOT TO ADD COLOUR TO THE UPDATE MESSAGE

```
Detected errors in 1 module.


-- TOO MANY ARGUMENTS ---------------------------------------- src/Main2.elm

Pattern Main.AddOne has too many arguments.

31|         AddOne colour ->
            ^^^^^^^^^^^^^^

Expecting 0, but got 1.
```

OOPS!

I FORGOT IT'S AN AMERICAN COLOR

```
Detected errors in 1 module.


-- NAMING ERROR ------------------------------------------------ src/Main2.elm

Cannot find type `Colour`

24|     = AddOne Colour
              ^^^^^^

Maybe you want one of the following?

    Color
    Color.Color
```

NOT THE COLOURS YOU WERE LOOKING FOR?

USING UNION TYPES

```
01  module Main exposing (..)
02
03  import Html exposing (Html, button, div, h1, h2, li, text, ul)
04  import Html.Attributes exposing (class, style)
05  import Html.Events exposing (onClick)
06
07
08  -- MODEL
09
10
11  type Colour
12      = StringColor String
13      | RgbColor Int Int Int
14
15
16  type alias Model =
```

We need to write the cssColor function

```
Detected errors in 1 module.

-- NAMING ERROR ------------------------------------------------- src/Main3.elm

Cannot find variable `cssColor`

72|             [ ( "background-color", cssColor colour )
                                        ^^^^^^^^
```


```elm
module Main exposing (..)

import Html exposing (Html, button, div, h1, h2, li, text, ul)
import Html.Attributes exposing (class, style)
import Html.Events exposing (onClick)


-- MODEL


type Colour
    = StringColor String
    | RgbColor Int Int Int


type alias Model =
```

Cover the new case everywhere Colour is used

```
type Colour
    = StringColor String
    | RgbColor Int Int Int
    | HexColor String
```

```
Detected errors in 1 module.
================================== ERRORS ==================================


-- MISSING PATTERNS ---------------------------------------------- src/Main3.elm

This `case` does not have branches for all possibilities.

87|>    case color of
88|>        RgbColor r g b ->
89|>            "rgb(" ++ toString r ++ "," ++ toString g ++ "," ++ toString b ++ ")"
90|>
91|>        StringColor colorString ->
92|>            colorString

You need to account for the following values:

    Main.HexString _

Add a branch to cover this pattern!

If you are seeing this error for the first time, check out these hints:
<https://github.com/elm-lang/elm-compiler/blob/0.18.0/hints/missing-patterns.md>
The recommendations about wildcard patterns and `Debug.crash` are important!
```

COOL THINGS FOR FREE

- Tooling elm-make and elm-reactor or create-elm-app or webpack
- Ellie - live online editor to share and compile code
- debugger import/ export
- tiny bundle sizes
- no accidental breaking changes
- Escape hatches to existing js code and libraries

JOIN THE ELM COMMUNITY

- Elm town, meetups, slack, discourse, great docs, conferences
- Great for learning - even for primary school children
- Easy for non-programmers - e.g. scientists needing visualisations