Elm @ DublinJS

Michael Twomey

@micktwomey

twoistoomany.com

https://github.com/micktwomey/elm-dublinjs

What is Elm?

The best of functional programming in your browser
— elm-lang.org

- Full programming language
- Focussed on the web front end
- Strongly Typed (in a good way)
- ML inspired (try saying Hindley–Milner three times)
- Compiles to Javascript
- Compiler is your friend (really!)

A Taste of Elm

```elm
import Graphics.Element exposing (..)

main : Element
main =
  show "Hello World"
```

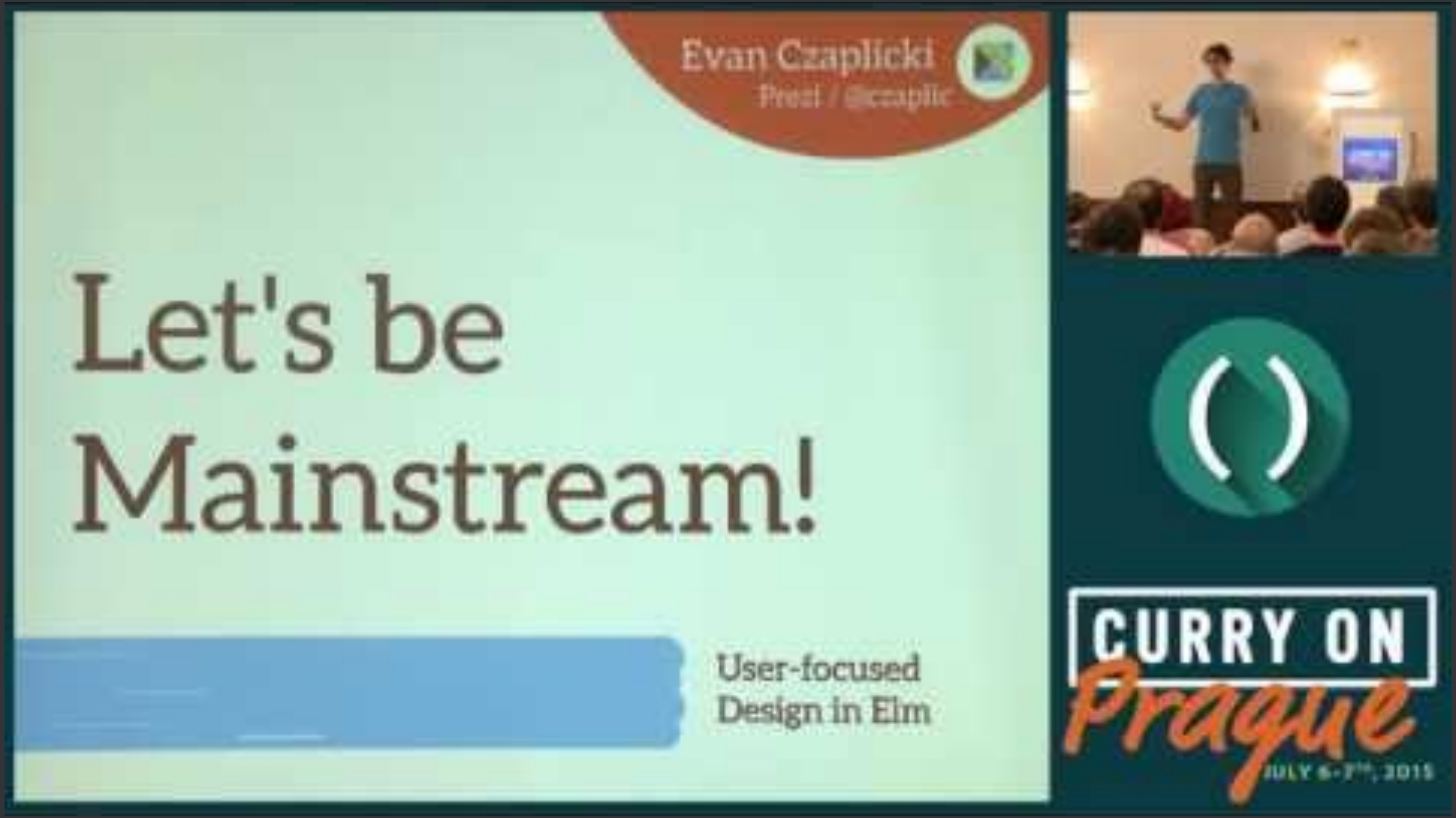
A Centred Taste of Elm

```
import Graphics.Element exposing (..)
import Window

main : Signal Element
main =
  Signal.map view Window.dimensions

view : ( Int, Int ) -> Element
view ( width, height ) =
  container width height middle (hello)
```

A Bigger Taste of Elm

```
hello : Element
hello =
  Text.fromString "Hello, World!"
    |> Text.bold
    |> Text.height 72
    |> leftAligned
```

Elm Philosophy



Evan Czaplicki
Prezi / @czaplic

Let's be Mainstream!

User-focused Design in Elm

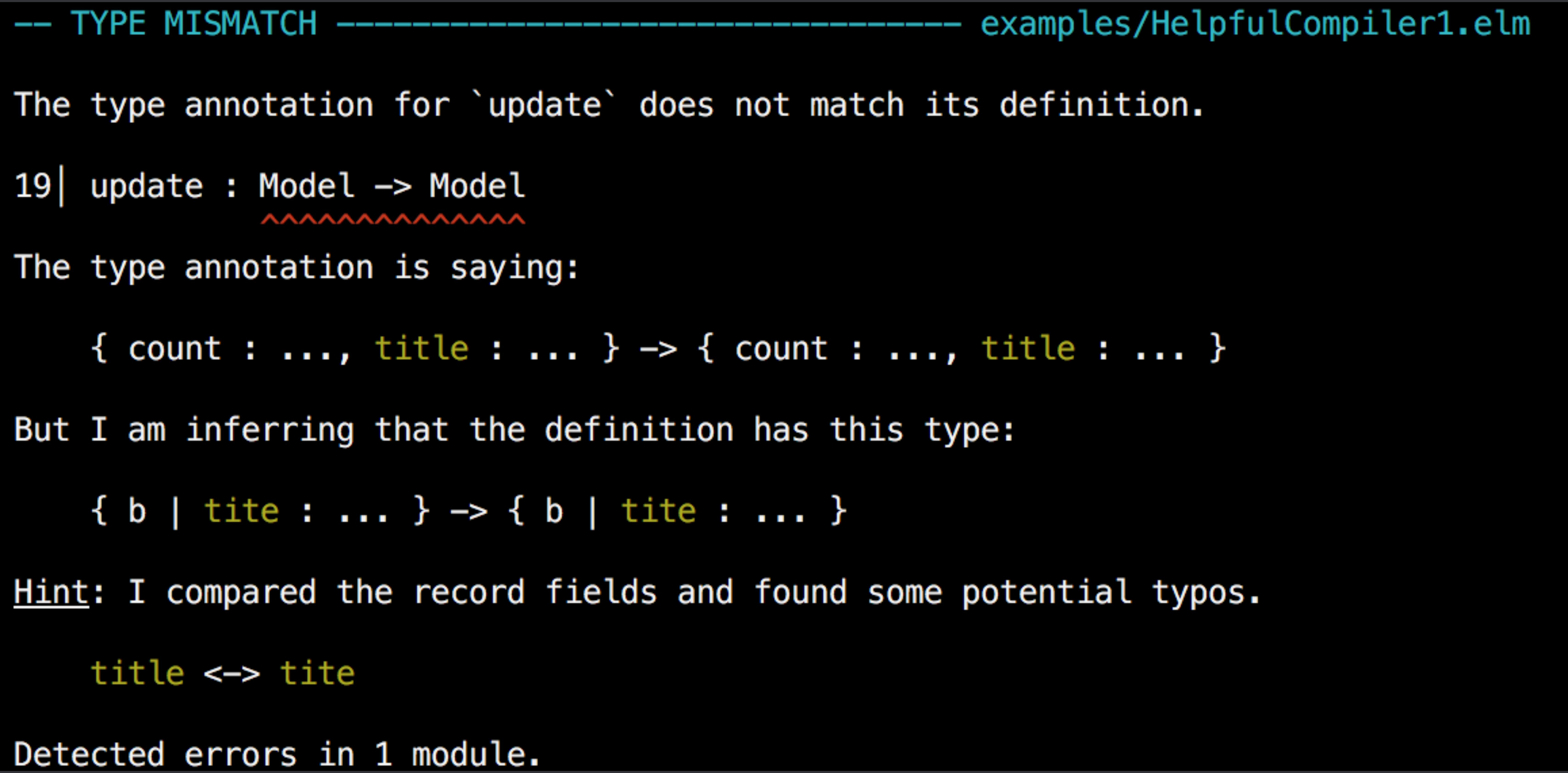
Some Party Tricks

- Nice Type System (really!)
- Helpful compiler error messages
- Time travelling debugger
- No runtime exceptions [2]
- Semantic package versioning baked in

[2] You can call Debug.crash to get one if you want

Compiler: Spot the Typo

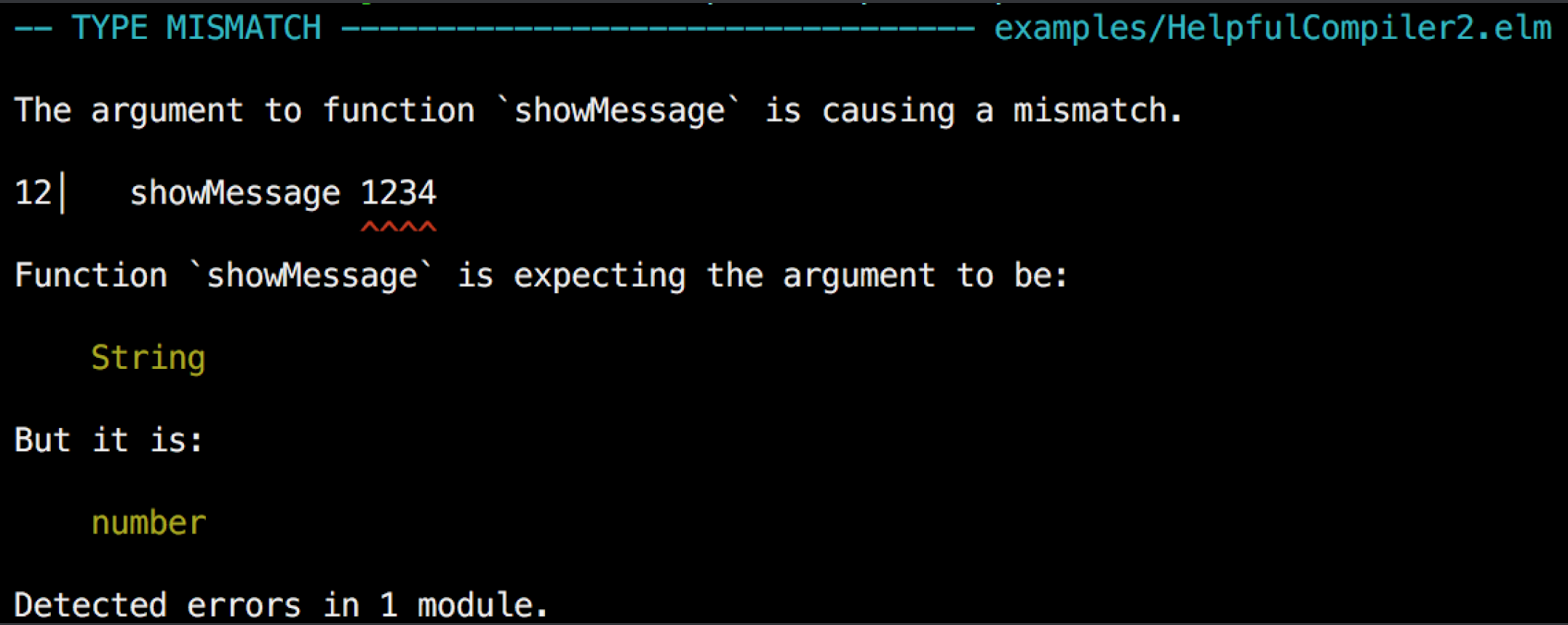
```
type alias Model = { title : String }

init : Model
init = { title = "Foo" }

update : Model -> Model
update model = { model | tite = "Bar" }

main : Element
main = show (update init)
```



```
-- TYPE MISMATCH ------------------------------- examples/HelpfulCompiler1.elm

The type annotation for `update` does not match its definition.

19| update : Model -> Model
             ^^^^^^^^^^^^^^
The type annotation is saying:

    { count : ..., title : ... } -> { count : ..., title : ... }

But I am inferring that the definition has this type:

    { b | tite : ... } -> { b | tite : ... }

Hint: I compared the record fields and found some potential typos.

    title <-> tite

Detected errors in 1 module.
```

Spot the "Type"-o

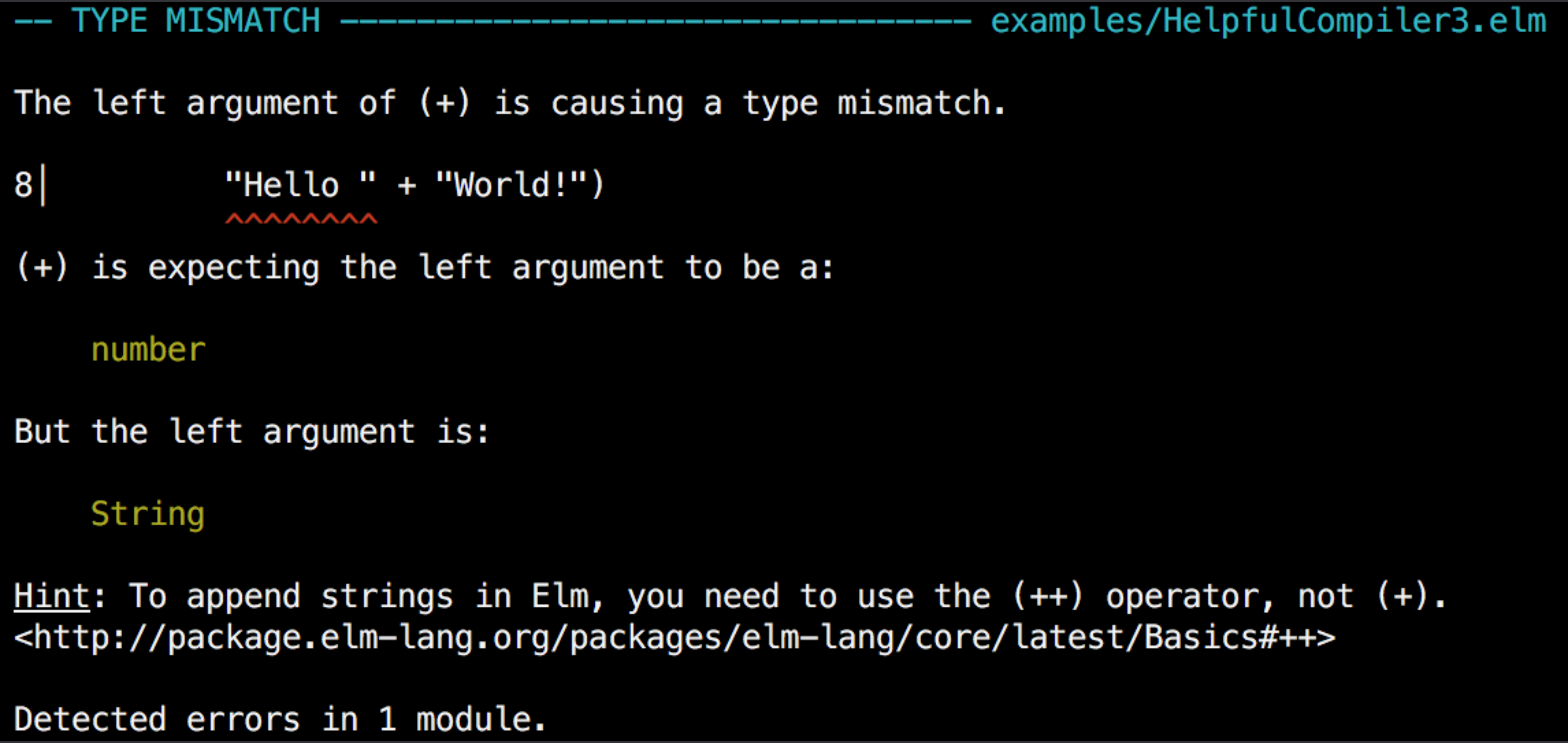
```
showMessage : String -> Element
showMessage message =
  show message

main : Element
main =
  showMessage 1234
```



```
-- TYPE MISMATCH --------------------------------- examples/HelpfulCompiler2.elm

The argument to function `showMessage` is causing a mismatch.

12|    showMessage 1234
                   ^^^^

Function `showMessage` is expecting the argument to be:

    String

But it is:

    number

Detected errors in 1 module.
```

Learning via the Compiler

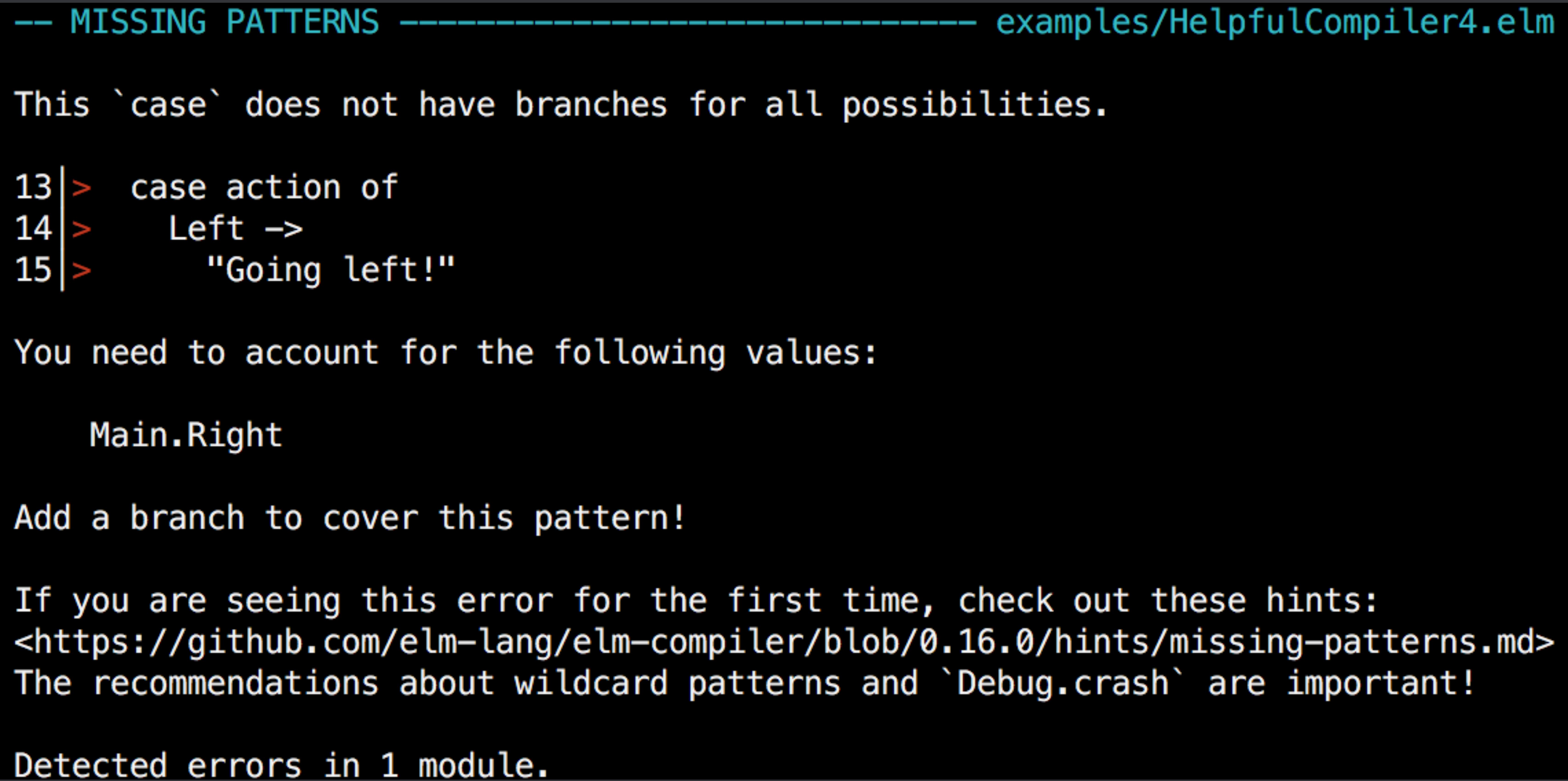
```
main : Element
main =
  show ("Hello " + "World!")
```



```
-- TYPE MISMATCH ------------------------------ examples/HelpfulCompiler3.elm

The left argument of (+) is causing a type mismatch.

8|           "Hello " + "World!")
             ^^^^^^^^
(+) is expecting the left argument to be a:

    number

But the left argument is:

    String

Hint: To append strings in Elm, you need to use the (++) operator, not (+).
<http://package.elm-lang.org/packages/elm-lang/core/latest/Basics#++>

Detected errors in 1 module.
```

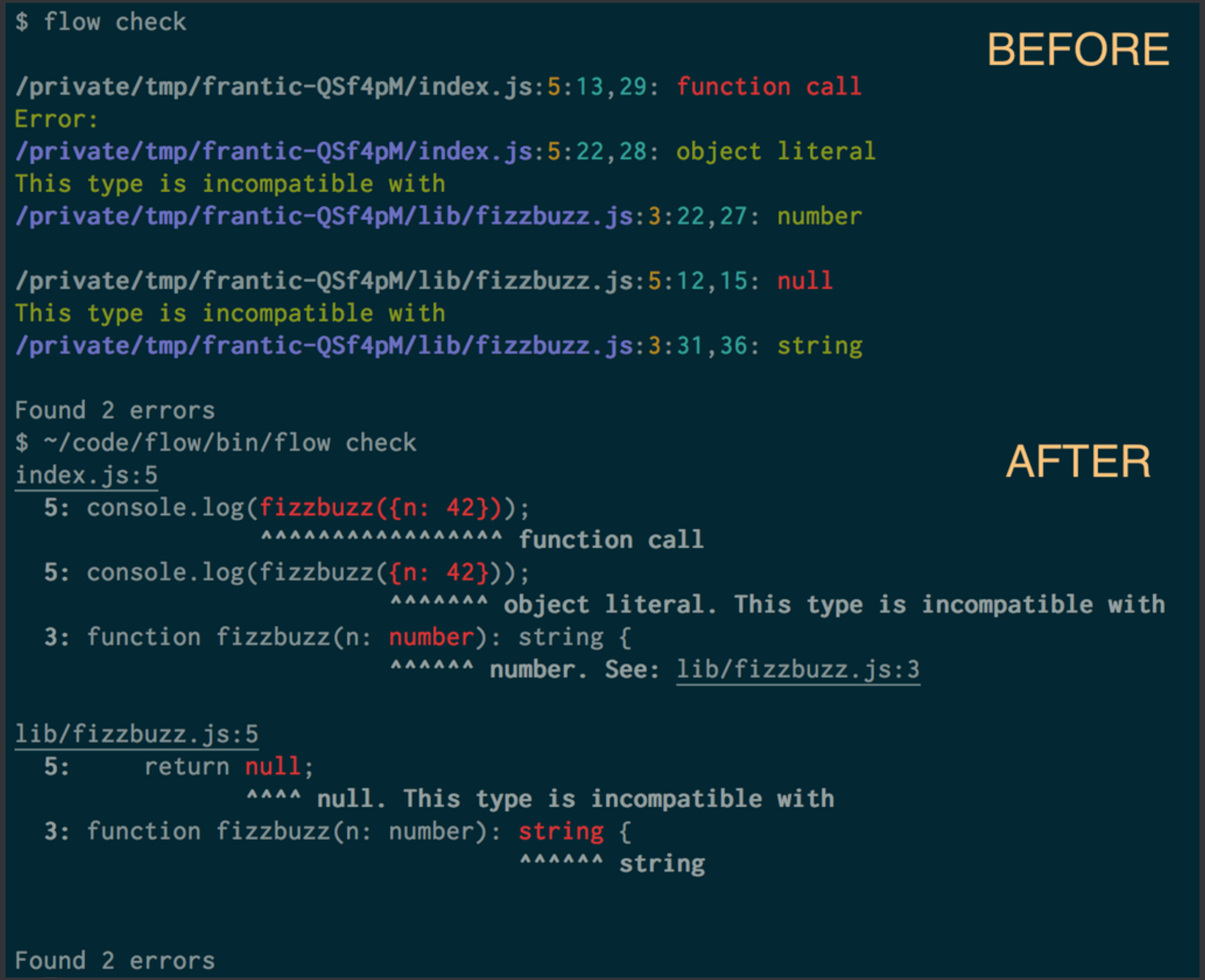
Hard to Express Incorrect Code

```
type Action
  = Left
  | Right

act : Action -> String
act action =
  case action of
    Left -> "Goto fail, I mean, going left!"

main : Element
main =
  show (act Right)
```



```
-- MISSING PATTERNS ------------------------------ examples/HelpfulCompiler4.elm

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

13|>  case action of
14|>    Left ->
15|>      "Going left!"

You need to account for the following values:

    Main.Right

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/0.16.0/hints/missing-patterns.md>
The recommendations about wildcard patterns and `Debug.crash` are important!

Detected errors in 1 module.
```

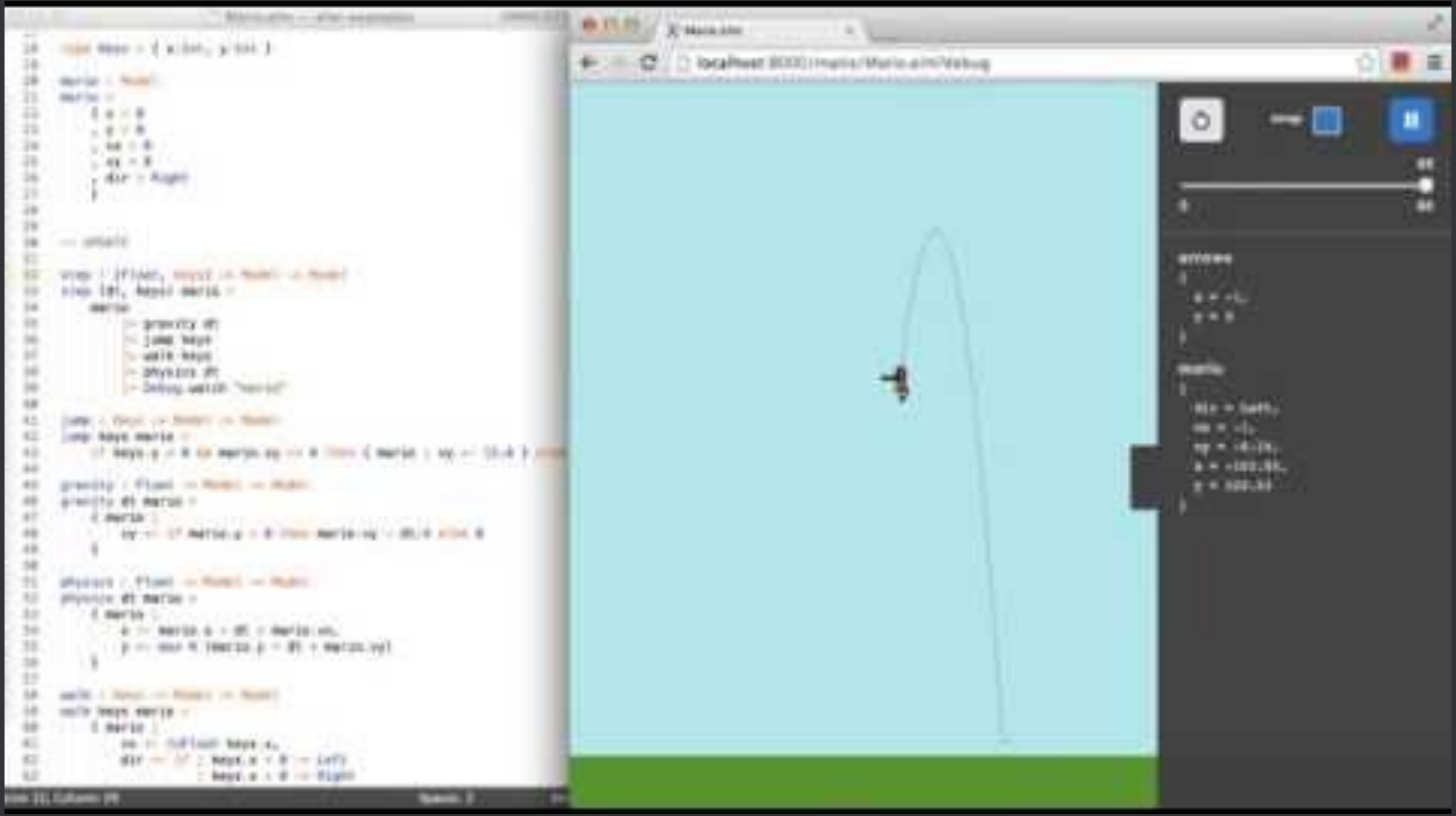
Good Ideas Spread [3]



```
$ flow check

/private/tmp/frantic-QSf4pM/index.js:5:13,29: function call
Error:
/private/tmp/frantic-QSf4pM/index.js:5:22,28: object literal
This type is incompatible with
/private/tmp/frantic-QSf4pM/lib/fizzbuzz.js:3:22,27: number

/private/tmp/frantic-QSf4pM/lib/fizzbuzz.js:5:12,15: null
This type is incompatible with
/private/tmp/frantic-QSf4pM/lib/fizzbuzz.js:3:31,36: string

Found 2 errors
$ ~/code/flow/bin/flow check
index.js:5
  5: console.log(fizzbuzz({n: 42}));
                 ^^^^^^^^^^^^^^^^^ function call
  5: console.log(fizzbuzz({n: 42}));
                          ^^^^^^^ object literal. This type is incompatible with
  3: function fizzbuzz(n: number): string {
                          ^^^^^^ number. See: lib/fizzbuzz.js:3

lib/fizzbuzz.js:5
  5:     return null;
                ^^^^ null. This type is incompatible with
  3: function fizzbuzz(n: number): string {
                                   ^^^^^^ string


Found 2 errors
```

[3] https://twitter.com/alex_frantic/status/651498914252648448

Time Travelling Debugger



```
type Keys = { x:Int, y:Int }

mario : Model
mario =
    { x = 0
    , y = 0
    , vx = 0
    , vy = 0
    , dir = Right
    }


-- UPDATE

step : (Float, Keys) -> Model -> Model
step (dt, keys) mario =
    mario
        |> gravity dt
        |> jump keys
        |> walk keys
        |> physics dt
        |> Debug.watch "mario"

jump : Keys -> Model -> Model
jump keys mario =
    if keys.y > 0 && mario.vy == 0 then { mario | vy <- [illegible] } [illegible]

gravity : Float -> Model -> Model
gravity dt mario =
    { mario |
        vy <- if mario.y > 0 then mario.vy - dt/4 else 0
    }

physics : Float -> Model -> Model
physics dt mario =
    { mario |
        x <- mario.x + dt * mario.vx,
        y <- max 0 (mario.y + dt * mario.vy)
    }

walk : Keys -> Model -> Model
walk keys mario =
    { mario |
        vx <- toFloat keys.x,
        dir <- if | keys.x < 0 -> Left
                  | keys.x > 0 -> Right
```

```
arrows
{
  x = -1,
  y = 0
}

mario
{
  dir = Left,
  vx = -1,
  vy = [illegible],
  x = [illegible],
  y = [illegible]
}
```

Semantic Versioning

```
$ elm package diff evancz/elm-html 3.0.0 4.0.2
Comparing evancz/elm-html 3.0.0 to 4.0.2...
This is a MAJOR change.

------ Changes to module Html.Attributes - MAJOR ------

    Removed:
        boolProperty : String -> Bool -> Attribute
        stringProperty : String -> String -> Attribute


------ Changes to module Html.Events - MINOR ------

    Added:
        type alias Options =
            { stopPropagation : Bool, preventDefault : Bool }
        defaultOptions : Html.Events.Options
        onWithOptions : String -> Html.Events.Options -> Json.Decode.Decoder a -> (a -> Signal.Message) -> Html.Attribute
```

Caveats

- Language still evolving, so can change with each major release [4]
- Interop with other JS might surprise you at first

[4] e.g. in 0.15.1 to 0.16.0 a bunch of syntax around records was removed to simplify

Integrating with Javascript

- Can embed elm app in pages
- Use ports to communicate
- Can rewrite everything in Elm [5]
- (Tangent: "native")

[5] My favourite approach 😄

Ports

```
port addUser : Signal (String, UserRecord)

port requestUser : Signal String
port requestUser =
    signalOfUsersWeWantMoreInfoOn
```

```
myapp.ports.addUser.send([
    "Tom",
    { age: 32, job: "lumberjack" }
]);

myapp.ports.requestUser.subscribe(databaseLookup);
function databaseLookup(user) {
    var userInfo = database.lookup(user);
    myapp.ports.addUser.send(user, userInfo);
}
```

Bonus: WebGL GLSL Compiler for Free!

```
vertexShader : Shader { attr | position:Vec3, color:Vec3 }
                      { unif | rotation:Mat4, perspective:Mat4, camera:Mat4 }
                      { vcolor:Vec3 }
vertexShader = [glsl|
attribute vec3 position;
attribute vec3 color;
uniform mat4 perspective;
uniform mat4 camera;
uniform mat4 rotation;
varying vec3 vcolor;
void main () {
    gl_Position = perspective * camera * rotation * vec4(position, 1.0);
    vcolor = color;
}
|]
```

Lot's of Nice Stuff

- Useful, strong types and a helpful compiler (thanks ML!)
- Hard to express bad code
- Needs way less tests!
- Fun!
- Time travelling debugger
- Semantic package versioning
- Can use it today!

Lot's More Stuff I Didn't Talk About

- Reactive programming (Effects, Tasks, Signals)
- Re-usable components with Elm's architecture
- Fast virtual DOM
- Nice canvas graphics
- No undefined / NULL / None !

Thank you! [6]

- Elm: http://elm-lang.org
- Elm User Group: http://www.meetup.com/Elm-User-Group-Dublin/
- Functional Kats: http://www.meetup.com/FunctionalKats/
- Slides: https://github.com/micktwomey/elm-dublinjs

[6] In a shock turn of events Udemy is hiring! michael.twomey@udemy.com