elm: a new language for the web

Brian Ginsburg

what do we use in webdev?

- **html** defines the **structure and content** of a web page, **css** defines the **style and layout**
- we use **javascript to interact** with users, the browser, and the outside world
- i will **only discuss frontend webdev**, the parts that **run in your browser**

https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web

what is javascript?

- **javascript** is a **dynamically typed** language that runs in web browsers
- javascript can be included in a web page **without compiling it**
- unless deferred, it is **executed while it is being processed**

https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

why the bad reputation?

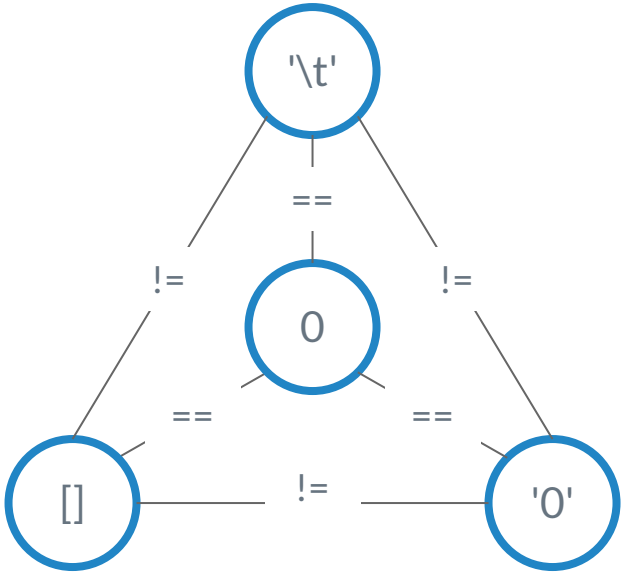
- the design of the language is **not great**
- after all, it was **developed by brendan eich in ten days**
- it was meant to be **forgiving** and **easy to use**

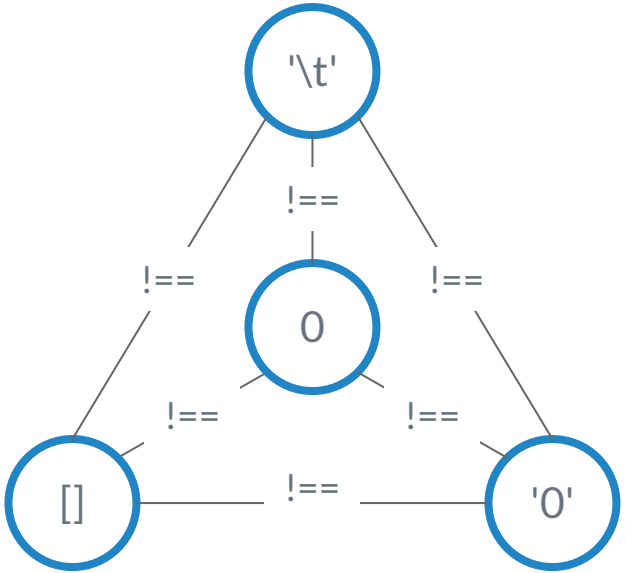
from bill sorensen

so what is not great?

- **equality can be truthy or falsy**, which means we check the content but not the type
- all **numbers are floats**
- **dynamic typing lets through errors** that a compiler could catch, and these errors can show up as **runtime exceptions**

how to test for equivalence?

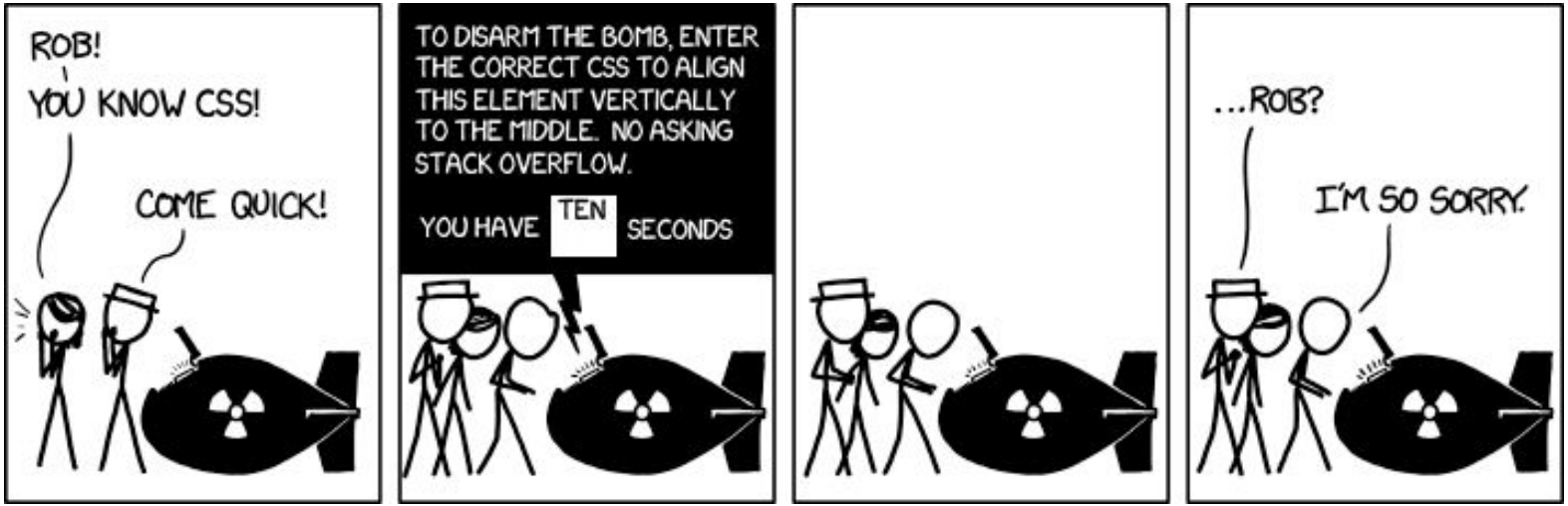




https://twitter.com/hsjoihs/status/792206474902515712

what about css?

- it has more concepts than javascript and is **sometimes unintuitive**



from xkcd

why does it matter?

- we now **make more than web pages**, we make **web applications**
- the **problems with javascript and css are worse now** that we have large codebases
- web development can **benefit from advances in programming languages**

why does it matter?

"The kinds of problems I kept running into were so silly. For example, trying to center an image in a box or reuse visual elements on multiple web pages was so incredibly difficult. I became obsessed with fixing these foundational issues."

- Evan Czaplicki

https://www.seas.harvard.edu/audiences/alumni/stories/2015/10/alumni-profile-evan-czaplicki-ab-12

what is elm?

- elm is a **functional language** that **compiles to javascript**
- you have already seen **haskell** in this class
- elm is a bit like haskell, but **simpler** and **designed for frontend web development**

http://elm-lang.org/

what is elm?

- elm is **statically typed** and has **type inference**
- all **data is immutable** and all **functions are pure**
- **null** and **exceptions** do not exist in the language

https://guide.elm-lang.org/

what is great about elm?

- it is a **compiled language** with **friendly error messages**
- **no runtime exceptions** in practice
- the **elm runtime** manages the state of our application and renders web pages for us

why would we want a compiler?

- when elm compiles our code, we know everything will be **defined and well-typed**
- this means we only have **runtime exceptions in rare cases**
- **logic errors** are still possible, and we still need to test for them

http://elm-lang.org/blog/compilers-as-assistants

why would we want a compiler?

- the elm compiler is **quite friendly** and explains what went wrong in detail
- it often **guesses at what we meant** and **gives beginner hints** and links to relevant documentation

http://elm-lang.org/blog/compilers-as-assistants

```
-- NAMING ERROR ---------------------------------------- teach/ErrorMisname.elm

Cannot find variable `List.nap`.

6|   List.nap identity (List.range 1 10)
     ^^^^^^^^
`List` does not expose `nap`. Maybe you want one of the following?

    List.map
    List.any
    List.map2
    List.map3

Detected errors in 1 module.
```

```
-- TYPE MISMATCH ------------------------------------- teach/ErrorMultiIf.elm

The 2nd and 3rd branches of this `if` produce different types of values.

 5|   if n < 0 then
 6|     "negative"
 7|   else if n > 0 then
 8|     "positive"
 9|   else
10|>    42

The 2nd branch has this type:

    String

But the 3rd is:

    number

Hint: All the branches of an `if` need to match so that no matter which one we
take, we get back the same type of value overall.

Detected errors in 1 module.
```

```
-- TYPE MISMATCH ------------------------------------ teach/ErrorTruthiness.elm

This condition does not evaluate to a boolean value, True or False.

9|     if String.length user.name then
           ^^^^^^^^^^^^^^^^^^^^^^^^^
You have given me a condition with this type:

    Int

But I need it to be:

    Bool

Hint: Elm does not have "truthiness" such that ints and strings and lists are
automatically converted to booleans. Do that conversion explicitly.

Detected errors in 1 module.
```

```
-- TYPE MISMATCH ---------------------------------------- teach/ErrorConcat.elm

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

5|    "Name: " + repo.name
      ^^^^^^^^
(+) 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.
```

why use elm?

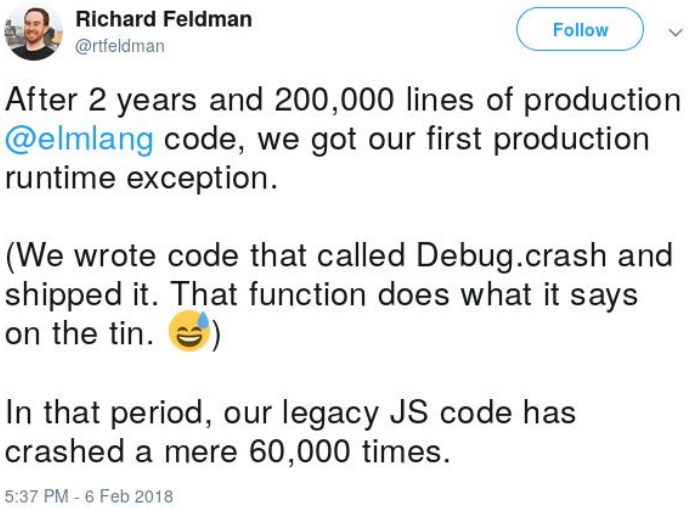
- elm applications are **highly reliable and maintainable**

“Elm is really *bullet proof, it’s not fake advertisement. We have had no runtime errors, as promised. Refactoring is easy, with the compiler telling you just about everything. And yes, really, when it compiles, it works !”*

https://discourse.elm-lang.org/t/ibm-releases-elm-powered-app/2364

why use elm?

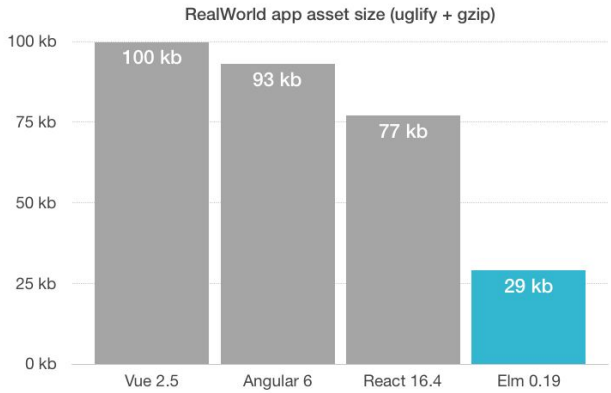
- **runtime exceptions rarely occur**
- they are far **more common in** applications written in **javascript**



https://twitter.com/rtfeldman/status/961051166783213570?lang=en

why use elm?

- the elm compiler **uses function-level dead code elimination** even on elm's core libraries
- **small assets mean faster delivery** to the client



https://elm-lang.org/blog/small-assets-without-the-headache

why use elm?

- elm is **fast** and **easier to optimize** than other frameworks
- elm is **fun to write**!
- **elm's community is super friendly**, you can find them on slack and elm discourse

https://discourse.elm-lang.org/
https://elmlang.herokuapp.com/

can we see some demos?

- **noredink:** https://www.noredink.com/
- **mybrocante.fr:** https://www.mybrocante.fr/
- **elm-cubik:** https://unsoundscapes.itch.io/cubik

anything to be aware of?

- **elm is a new language** and some web apis have not been implemented
- anything that cannot be done in elm can be done in javascript through **ports**
- **releases are infrequent** since stability and getting it right are prioritized

how does elm render a page?

- elm tracks the state of our application **in a model** and keeps its own representation of our web page
- we **define a model** for our application
- we **write a view** to display our state

https://guide.elm-lang.org/architecture/

how does elm render a page?

- **view is a function** that takes our model and produces a **description of the web page**
- the **elm runtime renders html** from this description

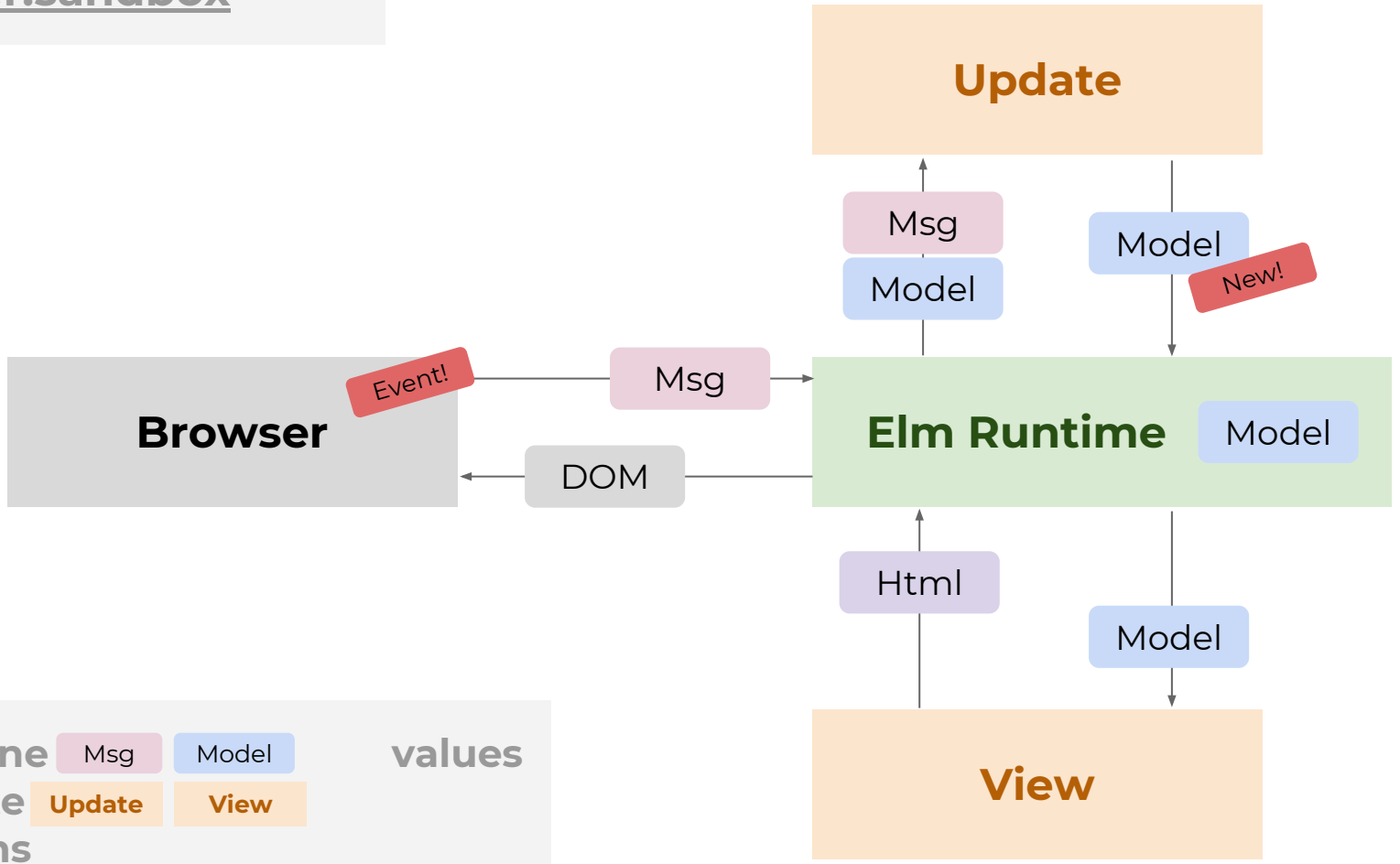
https://guide.elm-lang.org/architecture/

how do we update our state?

- we **write an update function** to handle user interactions as **messages that we define**
- all **changes to our state** happen in update
- this approach to managing state in part **inspired redux** which is used in react and angular

https://redux.js.org/

Browser.sandbox



We define Msg Model values
We write Update View functions
Elm does the rest!

how can we try elm?

- we can **try elm in ellie**, a browser-based coding environment
- elm is **easy to install using npm**, but you need to install node first
- node is also easy to install, binaries are available for all platforms

https://guide.elm-lang.org/install.html
https://nodejs.org/en/download/

can we see an example?

```
import Browser
import Html exposing (Html, button, div, text)
import Html.Events exposing (onClick)

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

type alias Model = Int

init : Model
init =
  0

type Msg = Increment | Decrement
```

```
update : Msg -> Model -> Model
update msg model =
  case msg of
    Increment ->
      model + 1

    Decrement ->
      model - 1

view : Model -> Html Msg
view model =
  div []
    [ button [ onClick Decrement ] [ text "-" ]
    , div [] [ text (String.fromInt model) ]
    , button [ onClick Increment ] [ text "+" ]
    ]
```

- the only **state** in our example is a counter initialized to zero
- our application is **defined in main** with **init for state**, **update for update**, and **view for view**

https://ellie-app.com/3ZqPyZNJn6ga1

who made the fancy slides?

- the error examples and the animated slide were **made by Mario Rogic**, and his original presentation is linked below
- **thanks to Mario** for graciously sharing them

https://docs.google.com/presentation/d/14rYuf7BzCZj8qjxGGRWpT1YVWdo_rG-FFiwNSEmFBV4/edit#slide=id.g2b22218a28_0_83