**What is Elm?**

Elm is a programming language that **compiles** to JavaScript. Elm is purely **functional**. Elm has static **types**.  
It is known for its friendly error messages, helping you find issues quickly and refactor large projects with confidence.   
Elm is also very fast and very small when compared with React, Angular, Ember, etc.

**Getting started with Elm:**

Install Elm on your computer: **npm install -g elm elm-format elm-repl**

Clone into the repository for today’s workshop: git clone https://github.com/atom-box/MKE

The repository above has textbooks by Pawan Poudel, Evan Cz, and eight others for learning Elm online.

**Why a purely functional language, with static types?**

* Less testing
* very few errors: “no runtime exceptions”
* Very maintainable. Scales well.
* outperforms most popular rendering libraries
* clean syntax
* Search Youtube for ‘Why Elm’
* “I like frontend again... “

**Functional Language Syntax**

Option 1: nested functions in Elm:

`main = Html.text (toString (add 5 (multiply 10 (divide 30 10))))

Option 2: pipes for the same four functions, still Elm:

`main =

divide 30 10

|> multiply 10

|> add 5

|> toString

|> Html.text

**Elm has no “This”.**

*Objects in Elm:*

\* You cannot ask for a field that does not exist.

\* No field will ever be \_undefined\_.

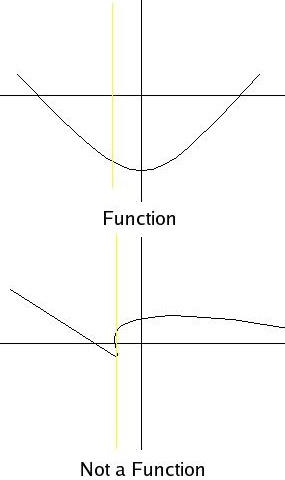
\* You cannot create recursive records with a \_this\_ keyword.

**Typing**

| **Language** | **Typing** |  |  |
| --- | --- | --- | --- |
| Kotlin | static |  | inferred |
| Elm | static | strong | inferred |
| C++ | static |  | partiallyInferred |
| Javascript | dynamic | duck |  |
| Python | dynamic | duck | strong |
| Typescript(MS) | dynamic | structural |  |
| Elixir | dynamic | strong |  |

**functional programming definition**

(Wikipedia explanation)

In computer science, functional programming is a programming paradigmâ€”a style of building the structure and elements of computer programsâ€”that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data. It is a declarative programming paradigm, which means programming is done with expressions or declarations instead of statements.

(My explanation) ---------------------------->

**Resources**

**Overview**: http://elm-lang.org/   
**Examples**: https://github.com/evancz/elm-architecture-tutorial  
**Beginner**: https://elmprogramming.com/  
**Beginner**: https://css-tricks.com/introduction-elm-architecture-build-first-application/  
**Intermediate**: https://guide.elm-lang.org/   
**Advanced**: https://github.com/izdi/elm-cheat-sheet   
**Connection to Elixir**: https://hackernoon.com/elmchemy-write-type-safe-elixir-code-with-elms-syntax-part-1-introduction-8968b76d721d   
**Bruce Tate**: https://pragprog.com/book/7lang/seven-more-languages-in-seven-weeks