Mini Course: Elm

Jeremie Gillet

Session 1: Getting Started

July 7, 2021

About Elm

Quick intro

- Language that compiles to Javascript
- Simple patterns, easy to read
- Clear error messages, ecosystem and documentation
- Functional programming
- Content of the course is from https://guide.elm-lang.org
- For testing one liners:

```
elm repl
```

Jumping right in: Buttons

Making an interactive website

- Download everything from https://github.com/oist/mini-course-elm (or git clone if you know how to do that)
- Open in your editor
- Open a terminal, cd into the folder “session1” and run

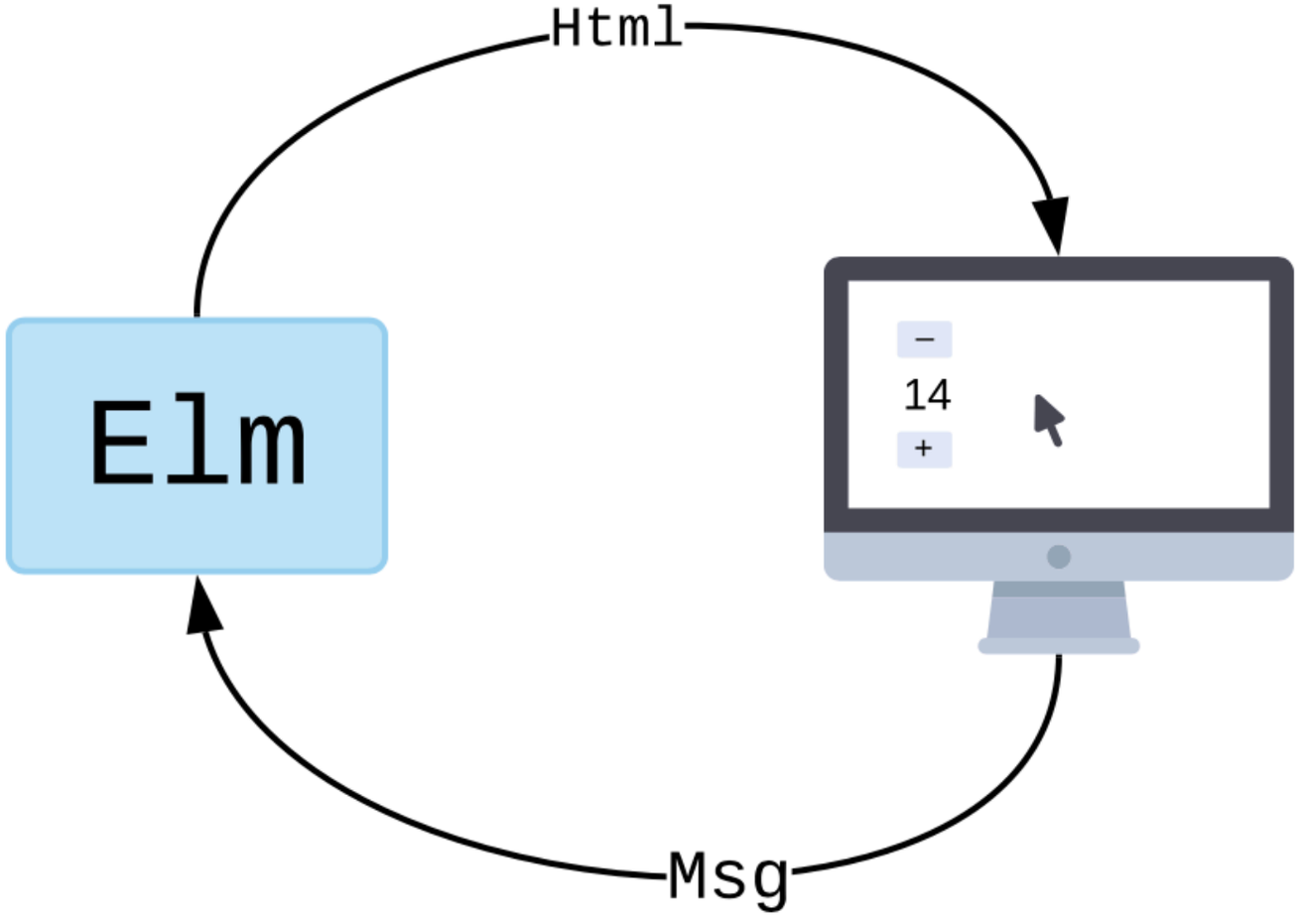
```
elm reactor
```

- Open a browser to http://localhost:8000 > src > Buttons.elm
- Exercise: Make a reset button (detailed instructions in the code)

Elm Architecture

Structure of an Elm program

- Model
  - The state, what your program keeps track of
- View
  - How to turn the model into HTML
- Update
  - Change the model based on messaged



Textfields

Dealing with text

- Open a browser to http://localhost:8000 > src > TextFields.elm
- Exercise: show the length of the text
- Documentation of every package on https://package.elm-lang.org

- New things:
  - Messages can “carry” data
  - Model is a *record*
  - Record update syntax { model | content = newContent }
  - Dot notation to access record field: model.content

Structure of a website

A brief and simplified overview

- Front end
  - HTML: The structure of the website
  - CSS: The look of the website
  - Javascript: interactivity
- Back end
  - On some remote server
  - Can be written any language
  - Talks to front end via some protocol (HTTP, web sockets…)
  - Database

Passwords

Towards a real webform

- Open a browser to http://localhost:8000 > src > TextFields.elm
- Exercise: show the length of the text
- New things:
  - More things in the Model
  - Shortcut notation for initializing a Model
  - Several types of messages, need case statement
  - Defining custom functions