



# Hello!



**My name is Teodor Lunaas Heggelund**

# **Teaching kids to code**

**With Elm**

**At Oslo Elm Day 2019**

# Outline

- *Part 1.* Why teach kids code.
- *Part 2.* Our hero learns Elm.
  - Elm is Legos with magic bricks
  - The magic tools of our Elm hero
  - The challenges on the hero's journey
- *Part 3.* Conclusions.

# **Part 1. Why teach kids code**

**How I came to think that this is important**



# **The Oslo Code Club Elm Course!**

# The Oslo Code Club Elm Course!

- 10 lessons
- 10 weeks
- Kids ages twelve to sixteen
- Some experience with Scratch and Python
- At the University of Oslo, Blindern
- Thanks to Erik Aasmundrud, Alexander Perry and Tjerand Silde for help starting up.



# **Part 2. Our hero learns Elm**

**What does a kid learning Elm look like?**

**2.1**

**Elm is legos with magic bricks**



**2.2**

# **The magic tools of our Elm hero**

# Ellie and Try Elm

*Enable vision and change without interruption*

# **The compiler**

***Shields us from damage***

# **The browser**

***Lets us share our creation***

**2.3**

## **Challenges on our hero's journey**



# Drawing with SVG: painting

Try Elm - Chromium

Try Elm x Reader

elm-lang.org/try

Bookmarks Turbolinks dev Norconsult Bolig Reporting Parsec org Focus Space Themed W Other bookmarks

More Examples Here Lights Compile

```
1 import Svg exposing (svg, circle, rect)
2 import Svg.Attributes exposing (width, height, viewBox, cx, cy, r, fill)
3
4
5 main =
6   svg
7     [ width "500", height "500", viewBox "0 0 200 200" ]
8     [ circle [ cx "30", cy "50", r "50", fill "blue" ] [ ]
9       , rect [ x "100", y "100", width "80", height "40", fill "red" ] [ ]
10    ]
11
```

svg | 500 x 500

Elements Console Sources Network Timeline Profiles Application Security Audits

```
<frame name="input" src="/examples/try/code"></frame>
<frame name="output" src="/try-message">
  <#document
    <!DOCTYPE html>
    <html>
      <script data-x-lastpass></script>
      <script type="text/javascript">window["_gaUserPrefs"] = { ioo : function() { return true; } }</script>
      <head></head>
      <body>
        <script></script>
        <script>var runningElmModule = Elm.Temp1486311906105292.fullscreen();</script>
        <script>svg width="500" height="500" viewBox="0 0 200 200"></svg> <= $0
      </body>
    </html>
  </frame>
</frameset>
</html>
```

html frameset frame html body svg

Styles Computed Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

```
element.style {
}
svg[Attributes Style] {
  width: 500;
  height: 500;
}
html|* > svg {
  transform-origin: 50% 50% 0px;
}
svg:not(:root), symbol, image, marker, pattern, user agent stylesheet
foreignObject {
  overflow: hidden;
}
* {
  transform-origin: 0px 0px 0px;
}
```

# Drawing with SVG: understanding coordinates?

```
import Svg exposing (svg, circle, rect)
import Svg.Attributes exposing (width, height, viewBox, cx, cy, r, fill)

main =
  svg
    [ width "500", height "500", viewBox "0 0 200 200" ]
    [ circle [ cx "30", cy "50"
              , r "50", fill "blue" ]
      [ ]
    ]
```

# Drawing with SVG: understanding coordinates

Visualize!

```
(0,0) ---- (100, 0) ---- (200, 0) --- x
|
(0, 100)   (100, 100)   (200, 100)
|
(0, 200)   (100, 200)   (200, 200)
|not
y
```

# Challenge: facing one huge main

“ I'm drowning in brackets and indentation! ”

# Challenge: one huge main

- Abstraction is hard

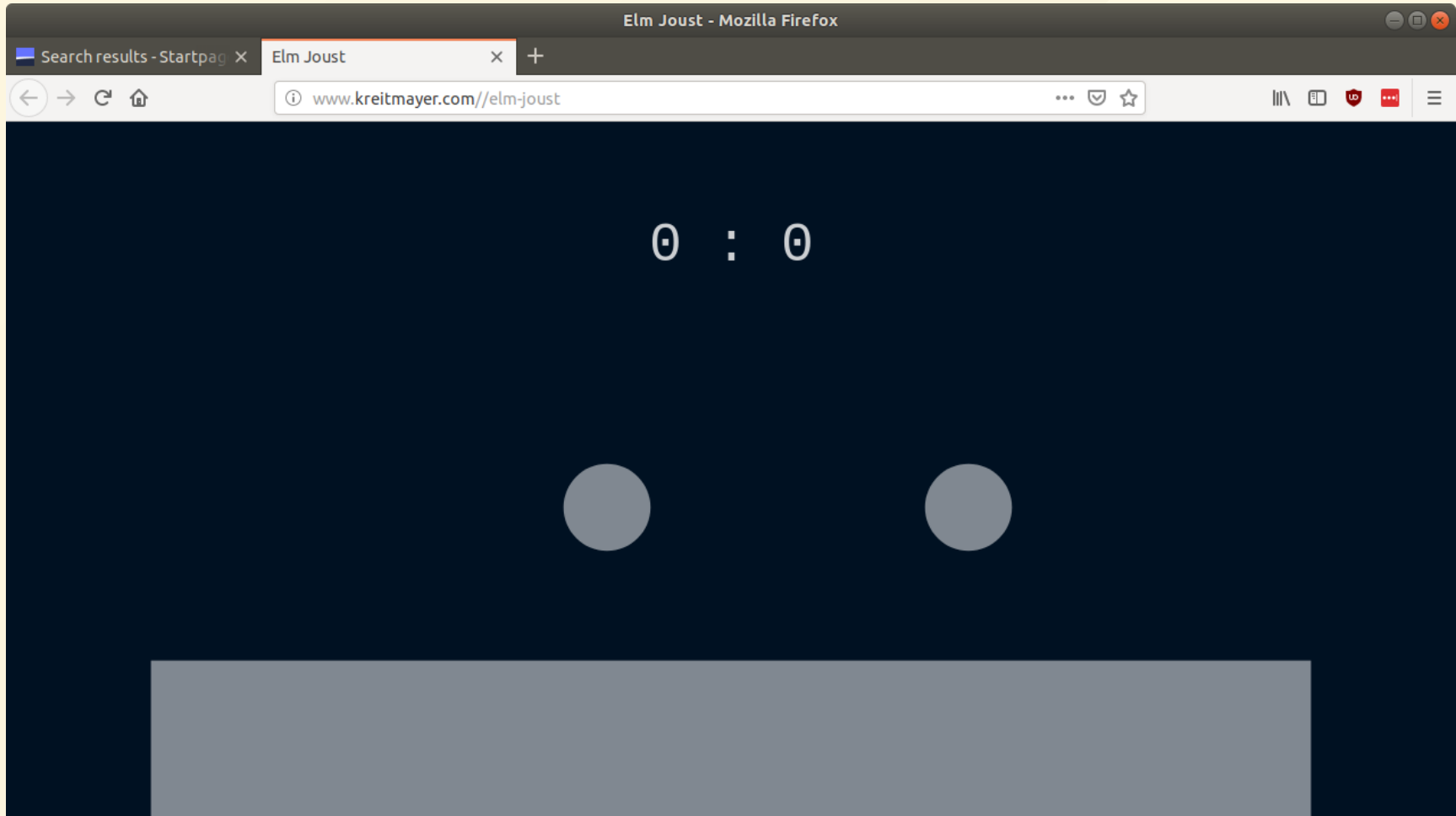
# Challenge: one huge main

- Abstraction is hard
- Abstraction is *hard*.

# Challenge: one huge main

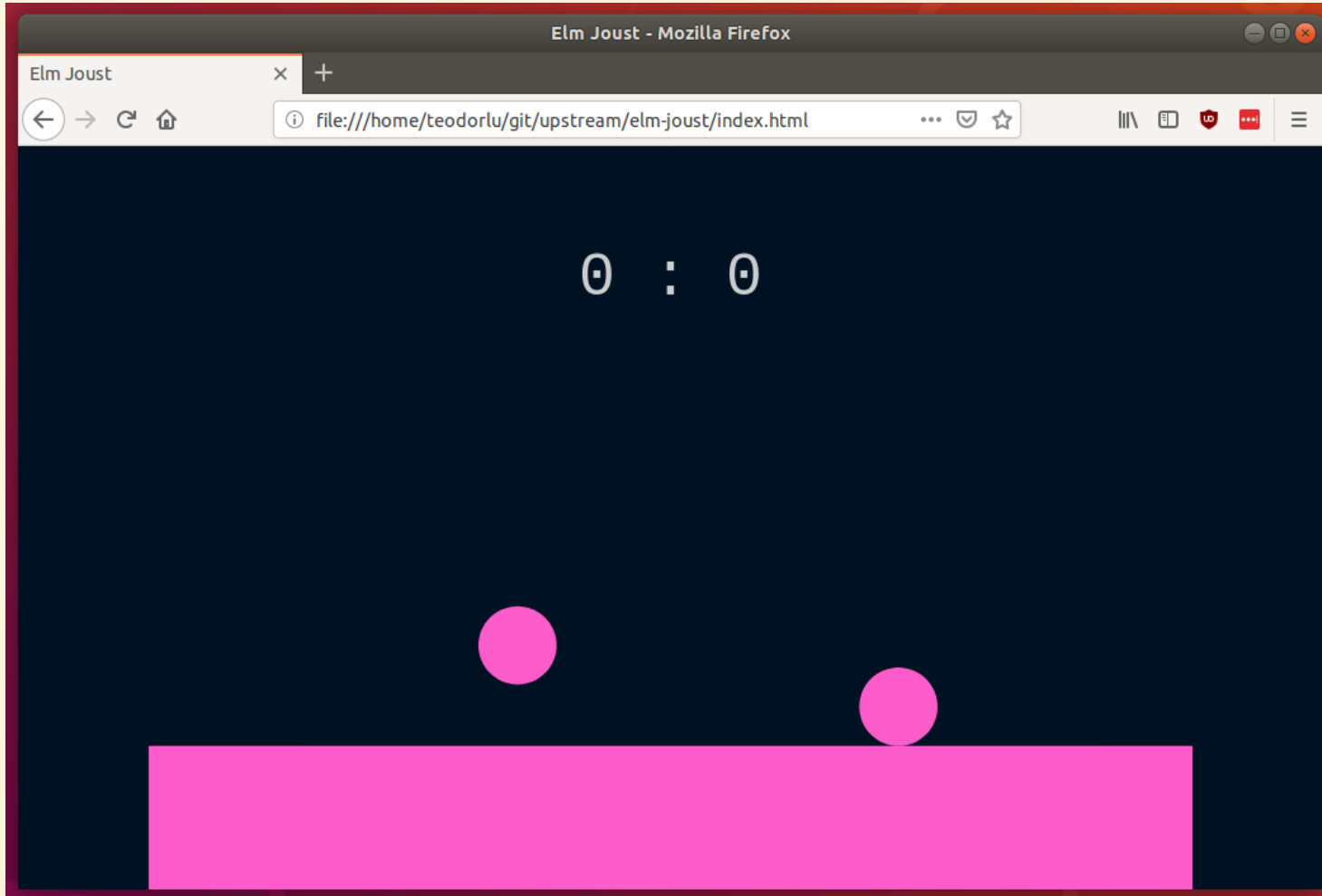
- Abstraction is hard
- Abstraction is *hard*.
- Show, don't tell.
  - Create good guides
  - Show good examples

# Elm Joust: exploring some real code





# Elm Joust: a new visual appeal



# Elm Joust: changing the rules



*Neo also likes changing the rules*

# **Part 3. Conclusions and the way forward**

**What did we learn?**

# Functional programming isn't that hard.

Kids don't fear it!

# **Too many moving parts may get you stuck**

I just couldn't introduce the same amount of fun per lesson with JavaScript.

# Enable others to make magic!

- *Make solid ground for others!*
- *Make it possible to change!*

... and the pieces you make may come into life in someone else's hands.

# Teach Kids Code makes it simple to volunteer



Visit [kidsakoder.no](https://kidsakoder.no) for more information!

# References

- [Bret Victor](#)'s *Learnable Programming* and other material
- [Jean-Paul Sartre](#) and [Simone de Beauvoir](#)'s philosophical ground for turning statics and dynamics into magic
  - [Philosophize This #106](#) is a nice resource
- My own musings on [static-dynamic interaction](#)



# Thank you!

- To you for listening.
- Evan and the Elm community for making Elm great!
- LÃfÃr Kidsa Koding (Teach Kids Code in Norway) for doing important work
- [Pure Logic AS](#) for letting me spend work time on this, and doing important work in civil engineering

# Thank you!

- The organizers of Oslo Elm Day for making this conference happen
- Erik, Perry and Tjerand for helping build the Code Clob Elm course

# Questions?

If there's any time!

Otherwise, just come talk to me.