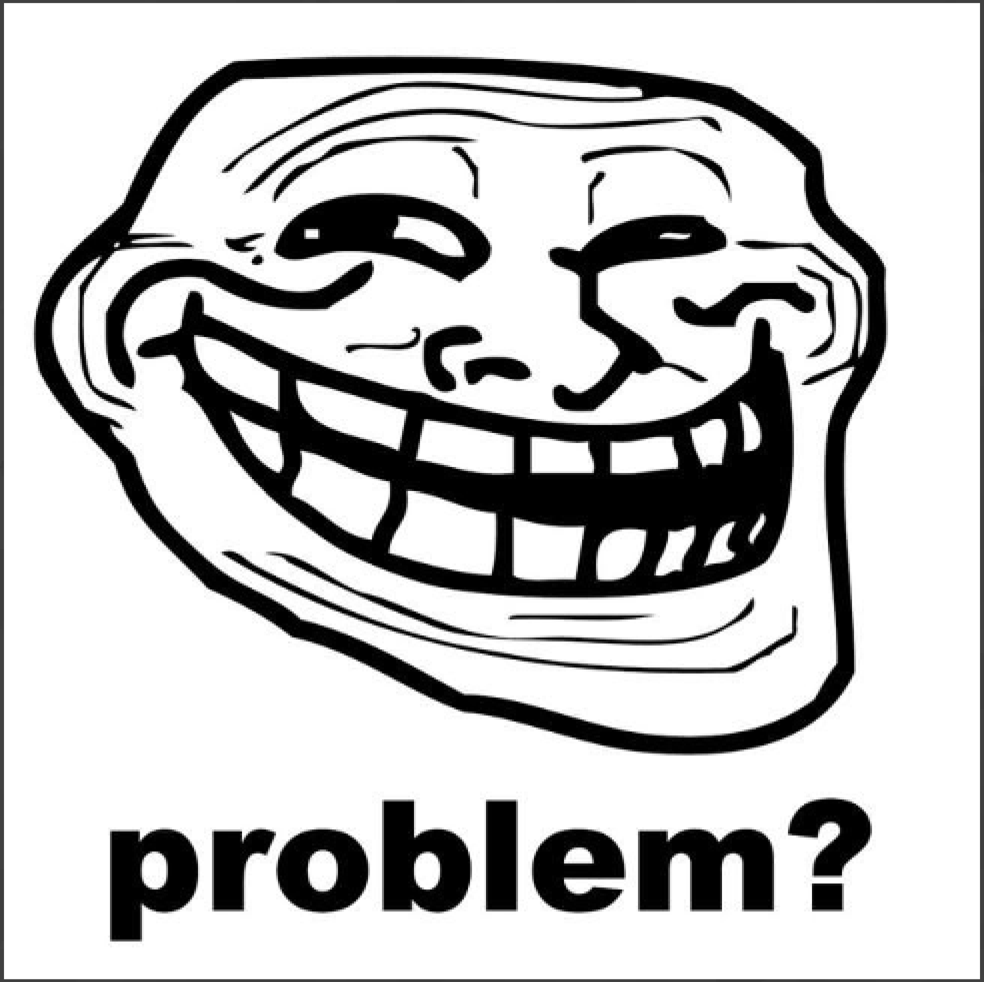
TECHDAY ELM

Mats Stijlaart, 2015-01-28

m.stijlaart@avisi.nl / @stil4m

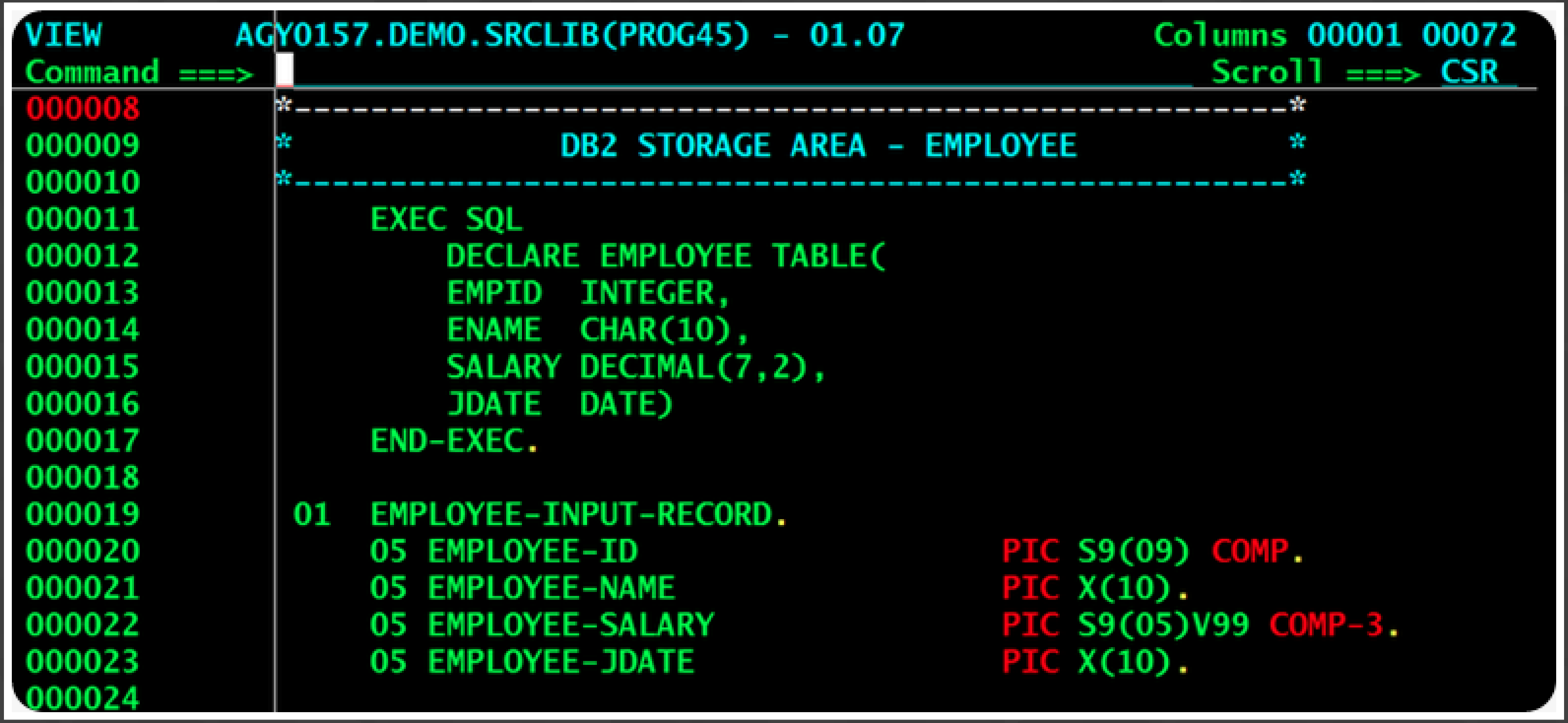
WHY?

Because of reasons

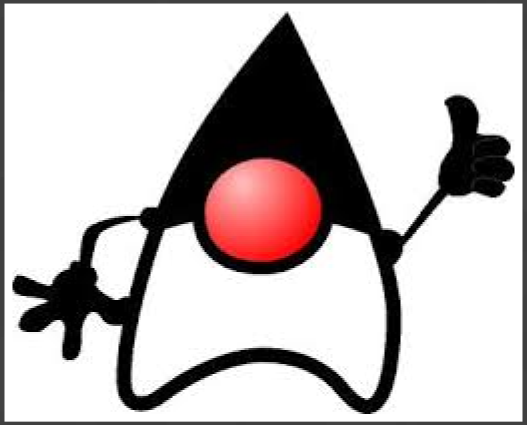


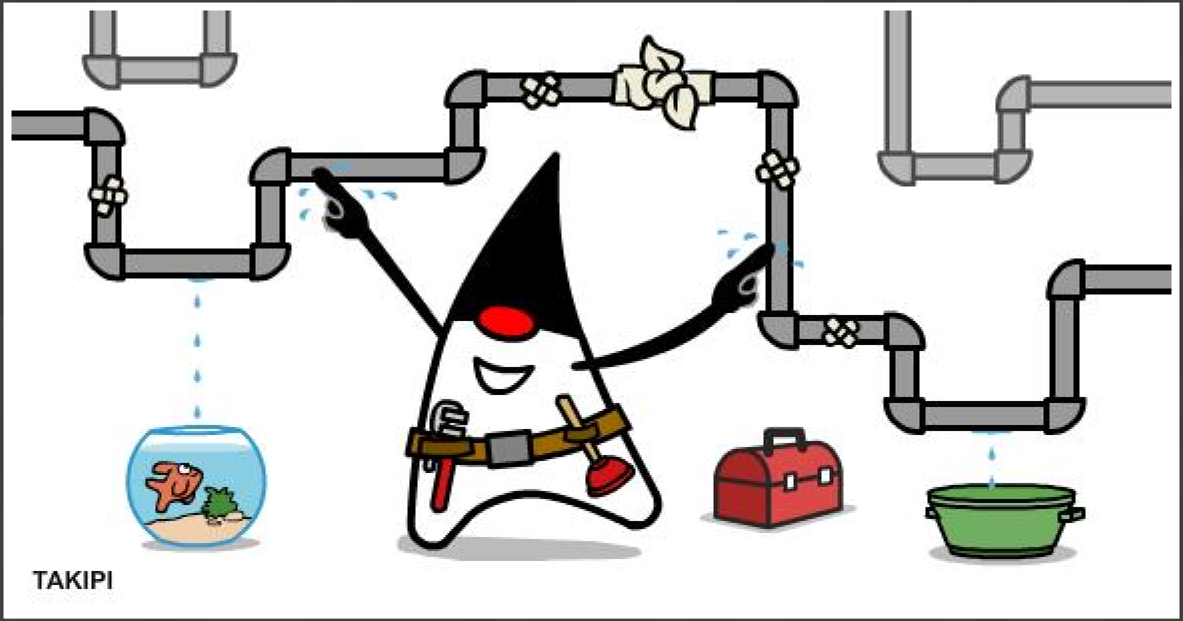
BECAUSE (1)

YOU NEED TO CHANGE



```
VIEW       AGY0157.DEMO.SRCLIB(PROG45) - 01.07              Columns 00001 00072
Command ===>                                                   Scroll ===> CSR
000008     *-----------------------------------------------------*
000009     *             DB2 STORAGE AREA - EMPLOYEE             *
000010     *-----------------------------------------------------*
000011          EXEC SQL
000012              DECLARE EMPLOYEE TABLE(
000013              EMPID  INTEGER,
000014              ENAME  CHAR(10),
000015              SALARY DECIMAL(7,2),
000016              JDATE  DATE)
000017          END-EXEC.
000018
000019      01  EMPLOYEE-INPUT-RECORD.
000020          05 EMPLOYEE-ID                PIC S9(09) COMP.
000021          05 EMPLOYEE-NAME              PIC X(10).
000022          05 EMPLOYEE-SALARY            PIC S9(05)V99 COMP-3.
000023          05 EMPLOYEE-JDATE             PIC X(10).
000024
```




TAKIPI

BECAUSE (2)

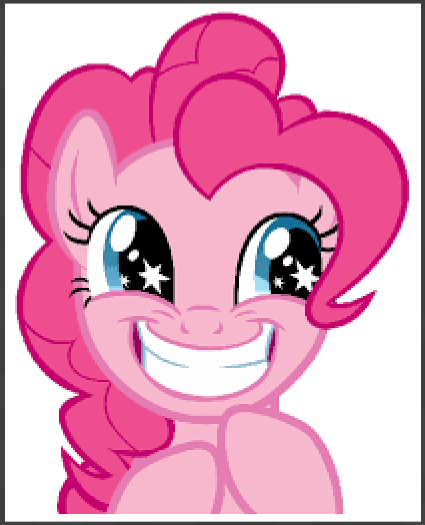
OF THE HISTORY OF PROGRAMMING (ISH)

Assembly	*Maintainability*
C	*Memory management*
Java	*All those types!*
Javascript	*Maintainability*

BECAUSE (3)

WE HAVE TO CHOOSE LIBRARIES AND FRAMEWORKS

OPTIMISTIC



REALITY

BECAUSE (4)

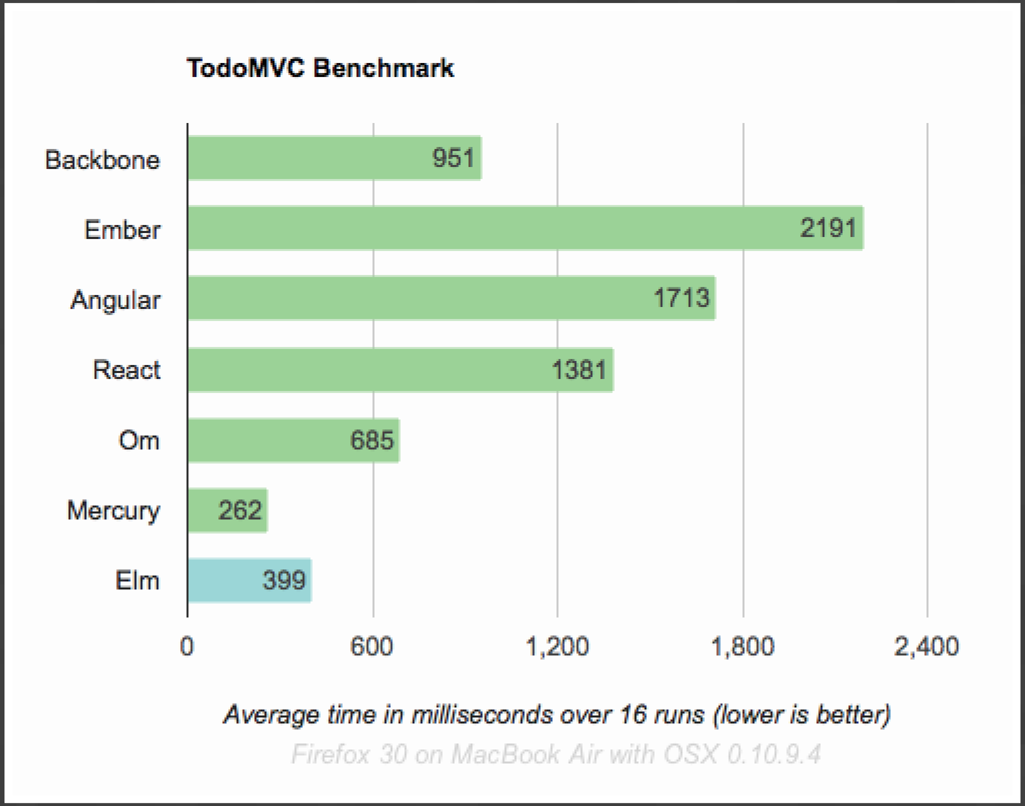
(IMHO) JAVASCRIPT IS THE NEW PHP

GOAL

- Use the concepts you encounter
- Do some hacking
- Consider to use Elm

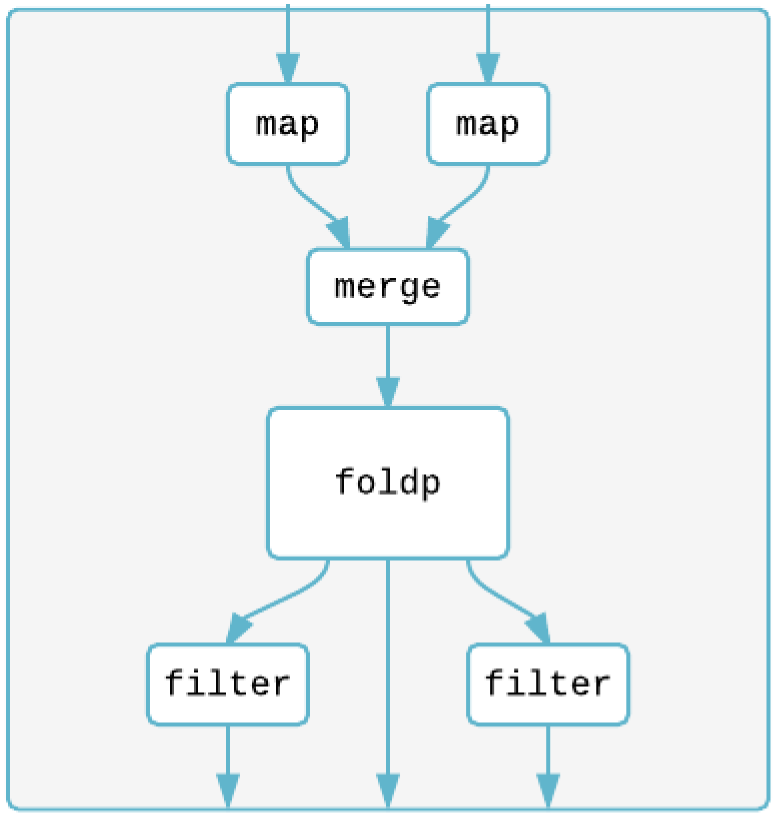
WHY USE ELM?

- Purity
- Immutable
- Forces architecture
- Designed for front-end
- Tooling: repl, packages, reactor
- Time Traveling Debugger
- New community, no ambiguities
- Virtual DOM
- Memoization


TodoMVC Benchmark
Backbone
951
Ember
2191
Angular
1713
React
1381
Om
685
Mercury
262
Elm
399
0
600
1,200
1,800
2,400
Average time in milliseconds over 16 runs (lower is better)
Firefox 30 on MacBook Air with OSX 0.10.9.4

REACTIVITY

SIGNALS



transform inputs into the right shape

merge the inputs into a single signal

update the state of your application
(The Elm Architecture)

route values to the appropriate service

REAL-LIFE EXAMPLE

CRASH COURSE

FUNCTIONS

```
add x y = x + y

add 10 11 -- 21

add' = (\x y -> x + y)
```

TYPES

```
add : Int -> Int -> Int
add x y = x + y


map : (a -> b) -> List a -> List b

sum : List Int -> Int

appSecret : String
```

PATTERN MATCHING

```
or b1 b2 =
  case b1 of
    True -> True
    False -> b2
```

LET ... IN

```
f x =
  let
    double = x * 2
  in
    double + double
```

LISTS

```
xs = [1, 2, 3]

4 :: xs -- [4, 1, 2, 3]

head list
  case list of
    x::xs -> Just x
    []    -> Nothing
```

TUPLES

```
t = ("Toepel", "Tuppel")
t' = (1, "Foo", 3.0, Just True)
```

UNION TYPES

```
type Action = Increment
            | Decrement


calc action m =
  case action of
    Increment -> m + 1
    Decrement -> m - 1


calc Increment 41
calc Decrement 1
```

UNION TYPES

WITH VALUES

```
type Counter = Cntr Int
type Action = Increment
            | Decrement


calc action (Cntr n) =
  case action of
    Increment -> Cntr (n + 1)
    Decrement -> Cntr (n - 1)


calc Increment (Cntr 41)
calc Decrement (Cntr 1)
```

POLYMORPHIC UNIONS

```
type Maybe a = Just a
             | Nothing
```

RECORDS

```
person = { name = "Mats Stijlaart", "age" = 25}
company = { name = "Avisi", "location" = "Arnhem"}


person.name  -- "Mats Stijlaart"
.name person -- "Mats Stijlaart"


List.map .name people
List.map .name [person, company]


printName : { a | name : String } -> String
printName x = "Name is: " ++ x.name
printName {name} = "Name is: " ++ name
```

COMPILER

(THE BEAST)

TYPES

```
add : Int -> Int -> Int
add x y = x + y

main = show (toString (add 1 "zero"))
```

```
Function `add` is expecting the 2nd argument to be:

    Int

But it is:

    String
```

SUGGESTIONS

```
sayHello : String -> String
sayHello name = "Hello " + name


Hint: To append strings in Elm, you need to use
the (++) operator, not (+).

<http://package.elm-lang.org/packages/elm-lang/core/latest/Basics#+-
```

TYPO

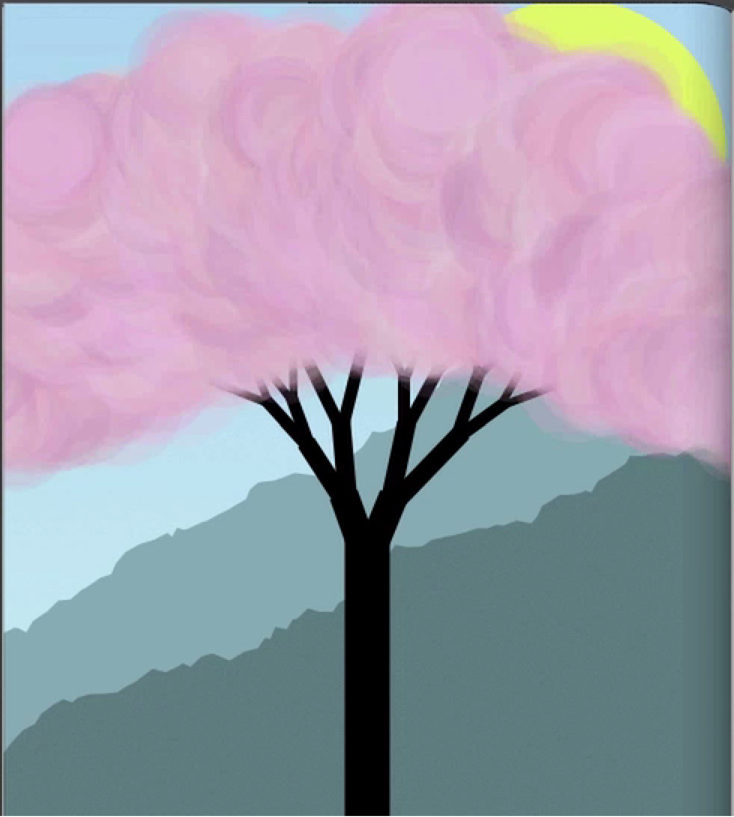
```
type alias Company = { name : String, address : String}

avisi : Company
avisi = { name = "Avisi", addres = "Meander 251" }
```

```
The type annotation is saying:

    { ..., address : ... }

But I am inferring that the definition has this type:

    { ..., addres : ... }

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

    address <-> addres
```

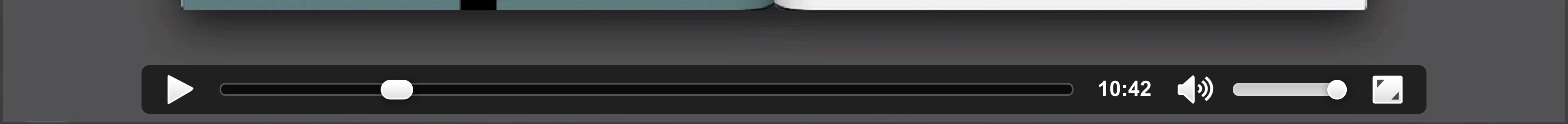
DEMO

- Signals
- Start-app

TIME TRAVELING DEBUGGER



```
    ctx.fillStyle = "#000";
    ctx.fillRect(0, -width/2, length, width);

    ctx.restore();

    var tipX = x + (length - width/2) * Math.cos(angle);
    var tipY = y + (length - width/2) * Math.sin(angle);

    if (i > 4) {
        blossomPoints.push([x,y,tipX,tipY]);
    }

    if (i < 6) {
        drawBranches(i + 1, angle + random(-0.15, -0.05) * Math.
        drawBranches(i + 1, angle + random( 0.15,  0.05) * Math.
    }
    else if (i < 12) {
        drawBranches(i + 1, angle + random( 0.25, -0.05) * Math.
    }
}

function drawBlossoms (blossomPoints) {
    var colors = ["#f5ceea", "#e8d9e4", "#f7c9f3", "#ebb4cc", "#
    ctx.globalAlpha = 0.15;

    for (var i = 0; i < blossomPoints.length; i++) {
        var p = blossomPoints[i];
        for (var j = 0; j < 40; j++) {
            var x = lerp(p[0], p[2], random(0,1)) + random(-10,1
            var y = lerp(p[1], p[3], random(0,1)) + random(-10,1

            ctx.fillStyle = colors[Math.floor(random(0,colors.le
            ctx.fillCircle(x, y, random(2,40));
        }
    }

    ctx.globalAlpha = 1.0;
}
```



HANDS ON

INSTALL

```
npm install -g elm
```

START APP

https://github.com/evancz/start-app
https://github.com/stil4m/techday-elm

```
<REPO>/handson
```

ARCHITECTURE

https://github.com/evancz/elm-architecture-tutorial

IDEAS

- Calculator
- Snake
- Memory
- Digital Clock
- Todo List
- Http Requests
- Dashboard
- 2D RPG
- http://builtwithelm.co/