Elm: Ноль runtime-ошибок в клиент-сайде.
Возможно ли такое?

~$ whoami

Avito

Яндекс Маркет

N1.RU
НЕДВИЖИМОСТЬ

DenKolesnikov

DenisKolesnikov

```
`${String.random()}.js`;
```

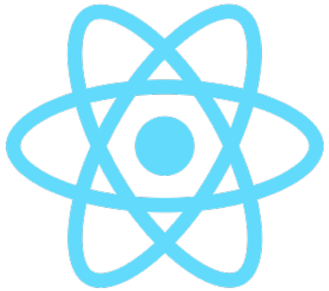

cljs


Knockout.

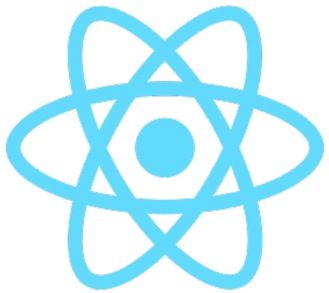

ember


jQuery
write less, do more.


METEOR




RIOT




cljs

React & Om

- Декларативный рендеринг
- Рендер - функция от состояния
- Атомарные изменения состояния

Elm

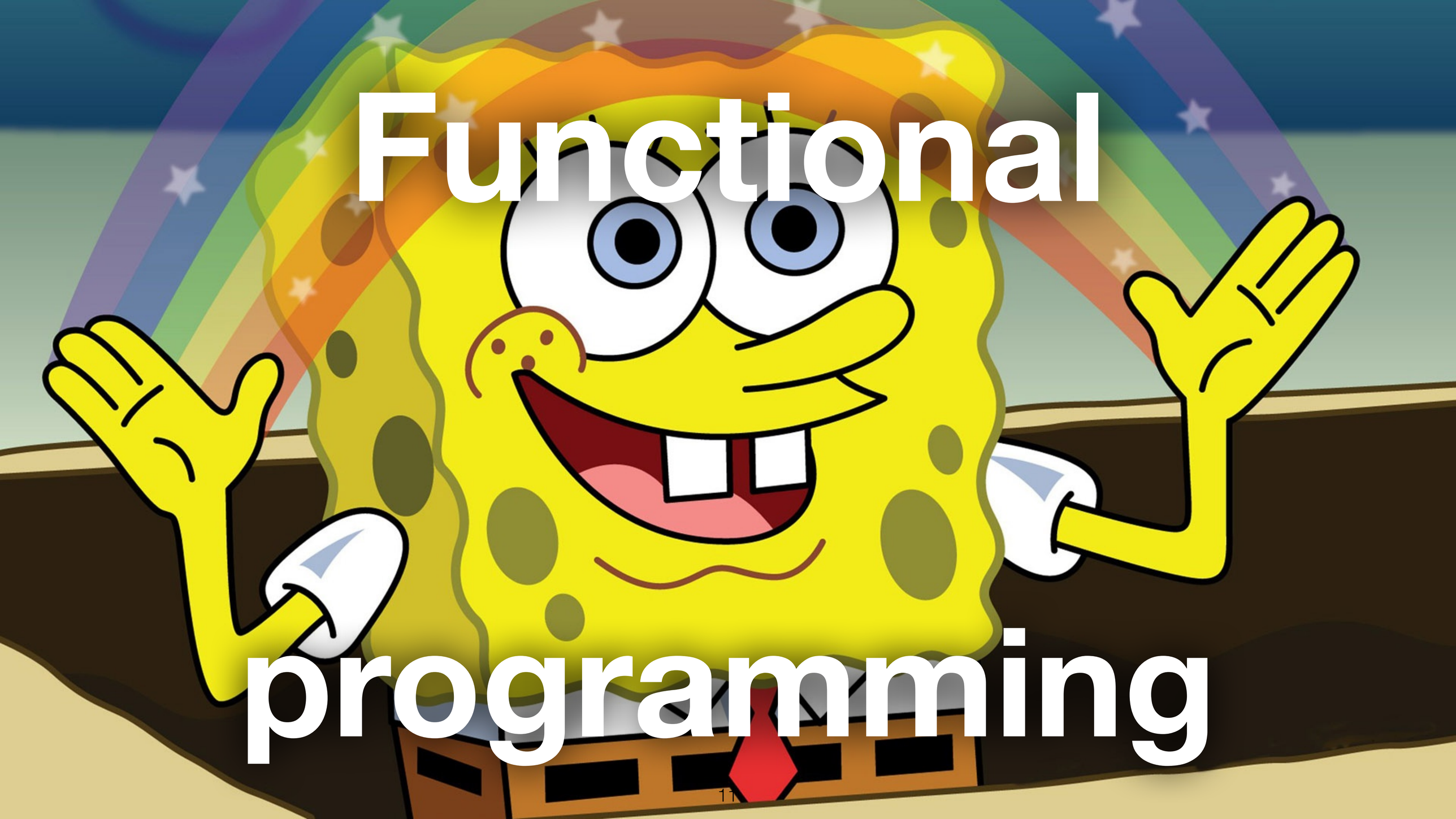
Elm

- Язык
- Появился в 2012
- Автор: Ewan Czaplicki
- Компилируется в JavaScript (и только в JavaScript)
- Функциональный

Дисклеймер

Я не функциональный программист и не имею степени в математике или физике. И целью доклада не является разжигание ~~пуканов~~ святой войны ООП против ФП. Цель доклада – познакомить слушателей с языком Elm и показать какие подходы используются в нем для достижения отказоустойчивости и developer experience.

Functional programming


Functional
programming

Functional programmer: (noun) One who names variables "x", names functions "f", and names code patterns "zygohistomorphic prepromorphism"

–*James Iry*

User ДМИТРИЙ МАЛИК

https://stackoverflow.com/users/570689/ДМИТРИЙ-МАЛИКОВ

stackoverflow Questions Developer Jobs Tags Users user:570689 Log In Sign Up

Profile Activity Meta User Network Profile

13,277 REPUTATION

6 52 109

ДМИТРИЙ МАЛИКОВ top 3% overall

✓

263 answers 84 questions ~1.1m people reached

Prague, Czechia

dmalikov

Member for 7 years, 1 month

81,688 profile views

Last seen 14 hours ago

Communities (28)

Stack Overflow	13.3k
Unix & Linux	4.1k
Code Review	257
Server Fault	213
Super User	198

View network profile →

Top Network Posts

84 Pass command line arguments to bash script

26 How to print the longest line in a file?

Top Tags (235)

haskell	SCORE 379	POSTS 96	POSTS % 28
bash	SCORE 312	POSTS 71	
grep	SCORE 148	POSTS 28	
prolog	SCORE 109	POSTS 56	
sed	SCORE 67	POSTS 28	
list	SCORE 65	POSTS 21	

View all tags →

Top Posts (347)

All Questions Answers | Votes Newest

98	Grep characters before and after match?	Nov 12 '11
48	Independent subset of cabal packages set	Mar 3 '13

Functional programming

- Иммутабельность
- Чистые функции
- Функции высшего порядка
- Рекурсия
- Ленивость

Иммутабельность

```
var a = 123;
a = 'foo';

{
    let b = 'bar';
    b = {
        baz: 'quux';
    }
}
```

Иммутабельность

```
const a = [];
a.push(1, 2, 3);

a // [1, 2, 3]

const b = {};
b.foo = 'bar';

b // { foo: 'bar' }
```

Иммутабельность

```
foo =
    [ "bar", "baz" ]


foo2 =
    foo ++ "quux"
```

Чистые функции

- Детерменированность
- Отсутствие побочных эффектов

Чистые функции

```
const add = (a, b) ⇒ (a + b);
```

```
let counter = 0;

const incrementCounter = () ⇒ {
    counter += 1;

    return counter;
}
```

Функции высшего порядка

- Могут передаваться как аргументы для других функций
- Могут возвращать функции

Функции высшего порядка

```
const makeAdd = a ⇒ b ⇒ (a + b);

const add5 = makeAdd(5);

add5(10); // 15
```

Рекурсия

- Отсутствие циклов
- Оптимизация хвостовой рекурсии

Рекурсия

```
const list = [1, 2, 3];

const sumList = (list, acc) => {
    return list.length ?
        sumList(list, acc + list.pop()) :
        acc;
}

sumList(list, 0); // 6
```

Ленивость

```
const notLazy = (a, b) ⇒ (a + 1);

notLazy(1, 0); // 2

const makeErr = () ⇒ {throw new Error('BANG')};

notLazy(1, makeErr()); // BANG
```

Elm

Elm syntax

Функции

```elm
module Main exposing (..)


add : Int -> Int -> Int
add a b =
    a + b


sampleFunc : String -> String
sampleFunc str =
    str
        |> toUpper
        |> (++) str


add 1 2 -- 3
sampleFunc "foo" -- fooFOO
```

Типы данных

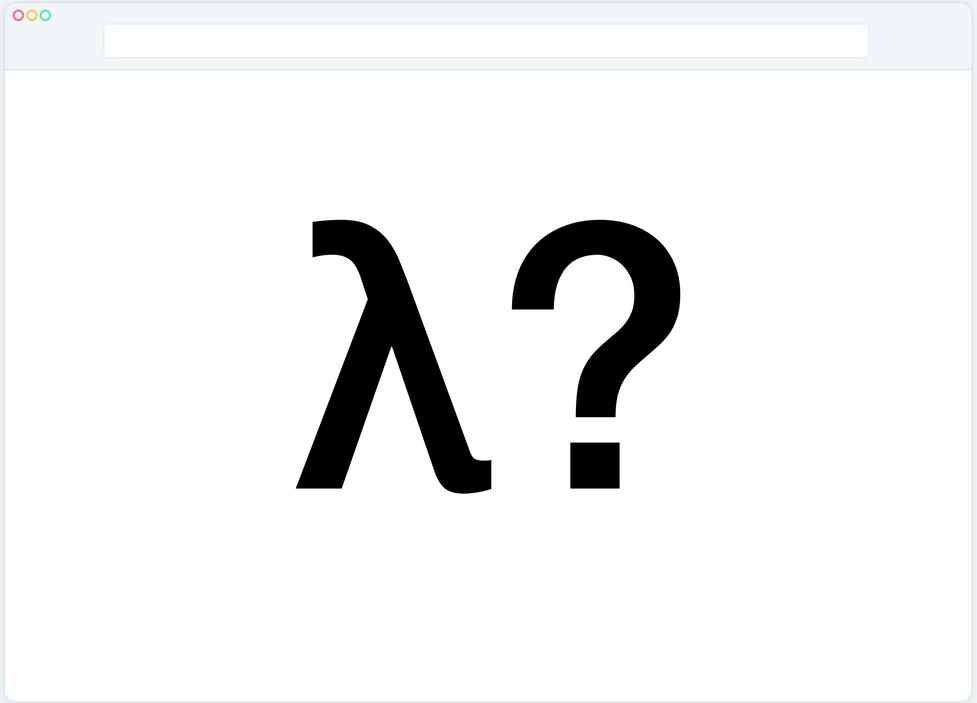
```
module Main exposing (..)


type Action
    = Increment Int
    | Decrement


type alias User =
    { id : Int
    , name : String
    }
```

Управляющие конструкции

```
module Main exposing (..)


baz =
    if foo > bar then
        "foo is greater"
    else if foo == bar then
        "equals"
    else
        "bar is greater"
```

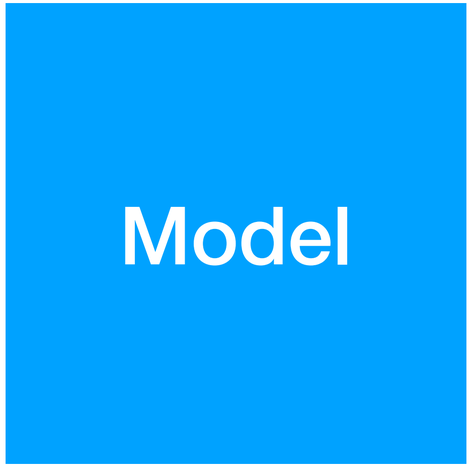
Управляющие конструкции

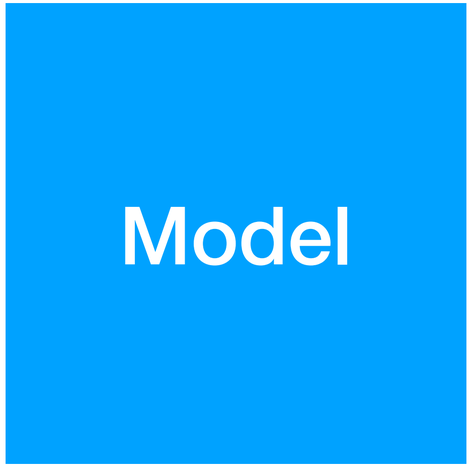
```
module Main exposing (..)


weekday dayInNumber =
    case dayInNumber of
        0 ->
            "Sunday"

        1 ->
            "Monday"

        _ ->
            "Unknown day"
```


λ?

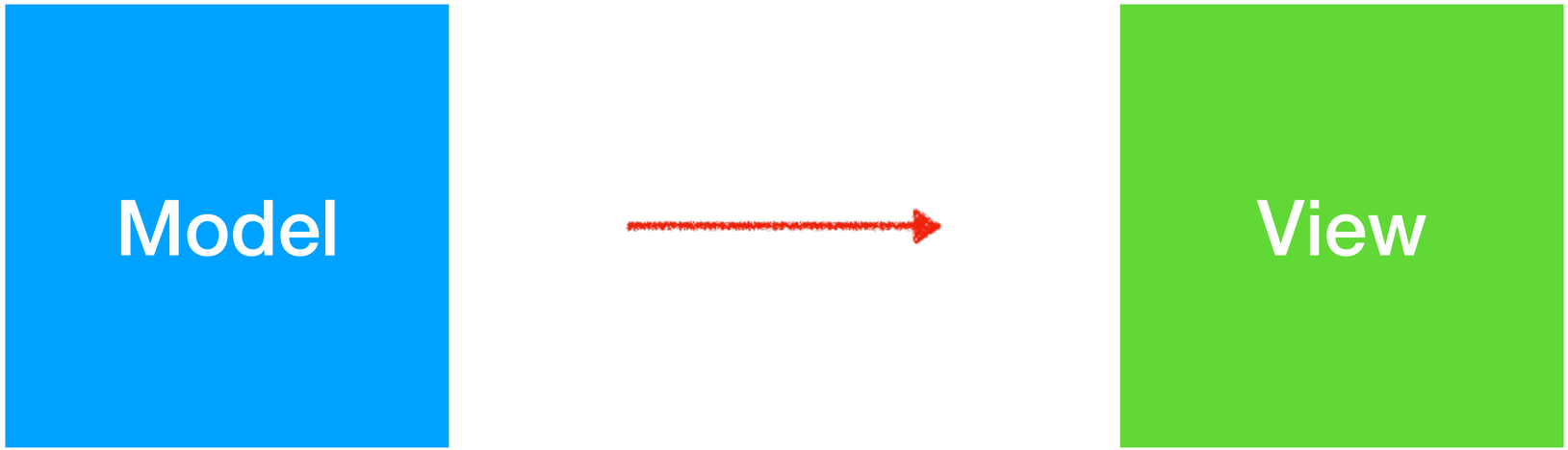
TEA

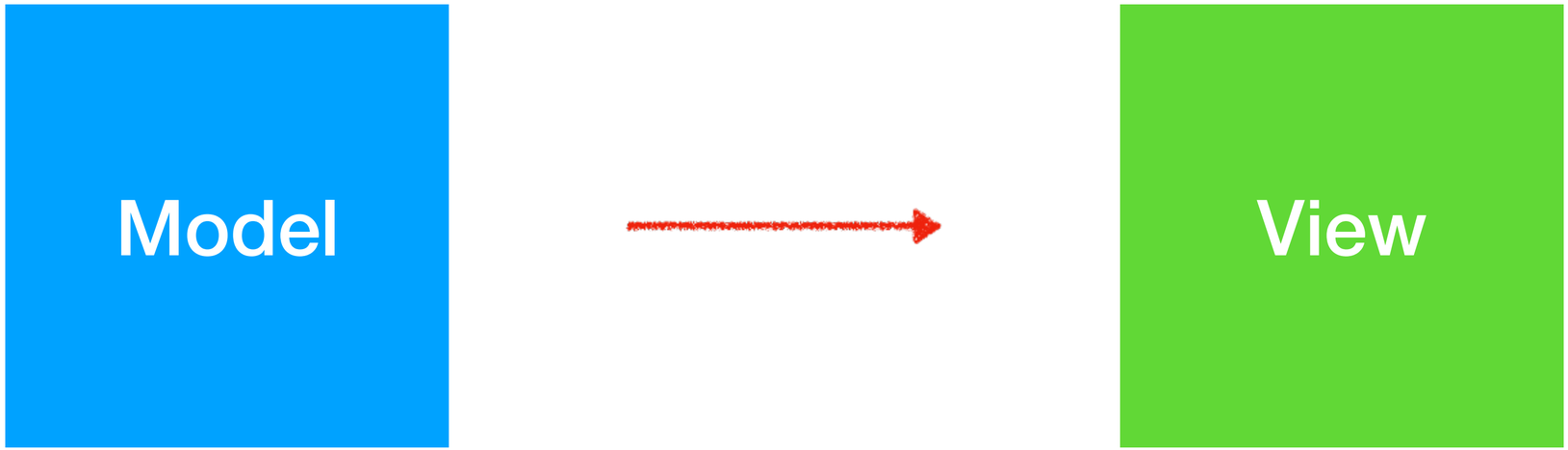
The Elm Architecture


Model

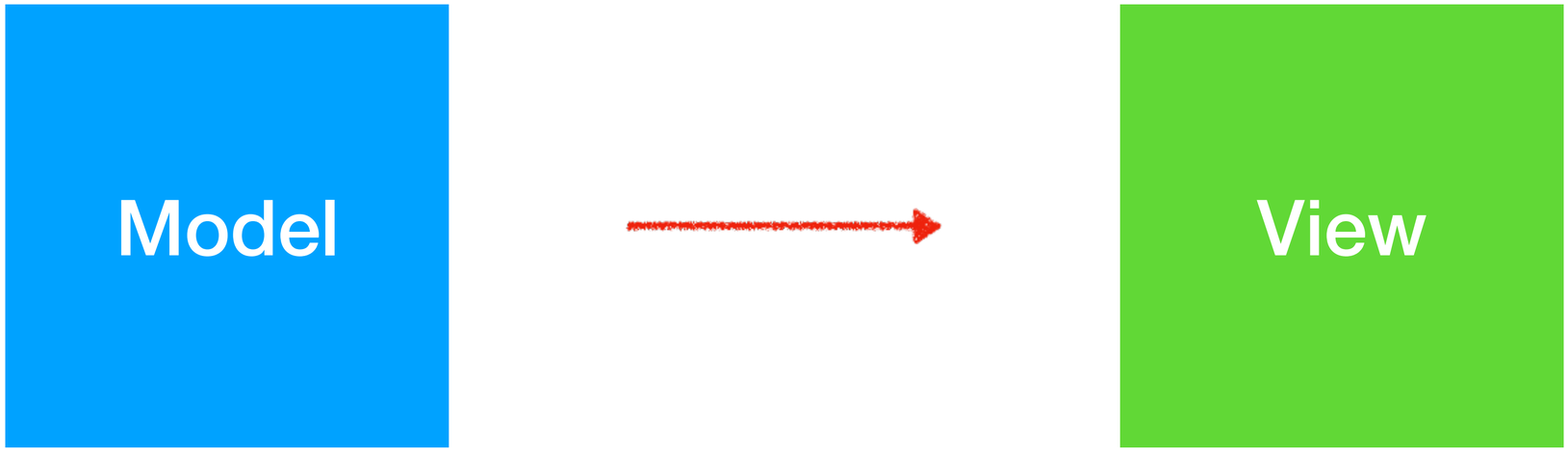



```
type alias Model =
    { counter : Int }
```


Model
View




```
view : Model → Html Msg
view model =
    div [] [ text (toString model.counter) ]
```




```
view : Model → Html Msg
view model =
    div []
        [ button [ onClick Decrement ] [ text "-" ]
        , text (toString model.counter)
        , button [ onClick Increment ] [ text "+" ]
        ]
```

```
React.createElement(
  type,
  [props],
  [...children]
);
```

```
node :
    String
    → List (Attribute msg)
    → List (Html msg)
    → Html msg
```

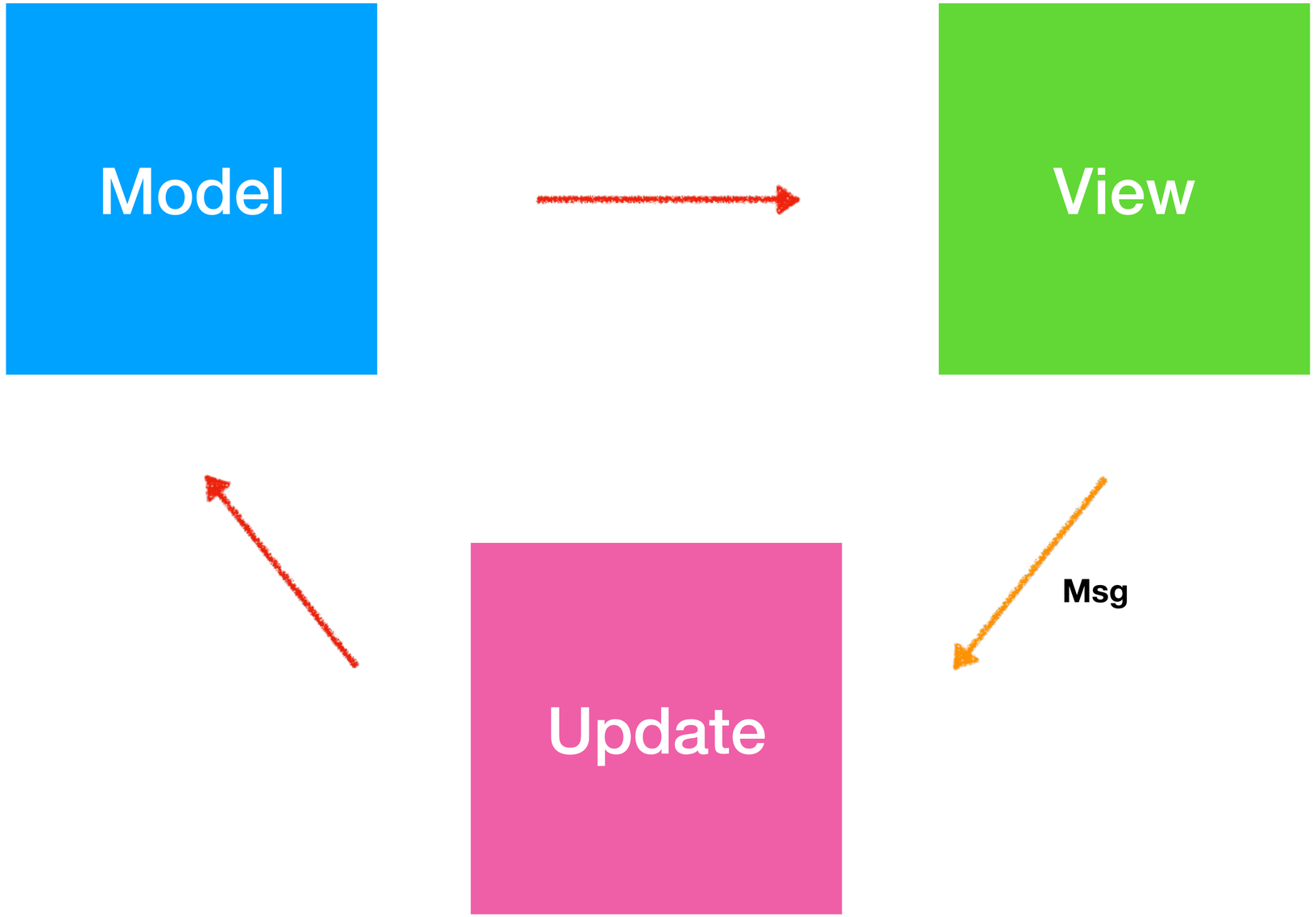
Update

Update

```
update : Msg → Model → Model
update msg model =
    case msg of
        Increment →
            { model | counter = model.counter + 1 }

        Decrement →
            { model | counter = model.counter - 1 }
```

Update

```elm
type Msg
    = Increment
    | Decrement


update : Msg → Model → Model
update msg model =
    case msg of
        Increment →
            { model | counter = model.counter + 1 }

        Decrement →
            { model | counter = model.counter - 1 }
```

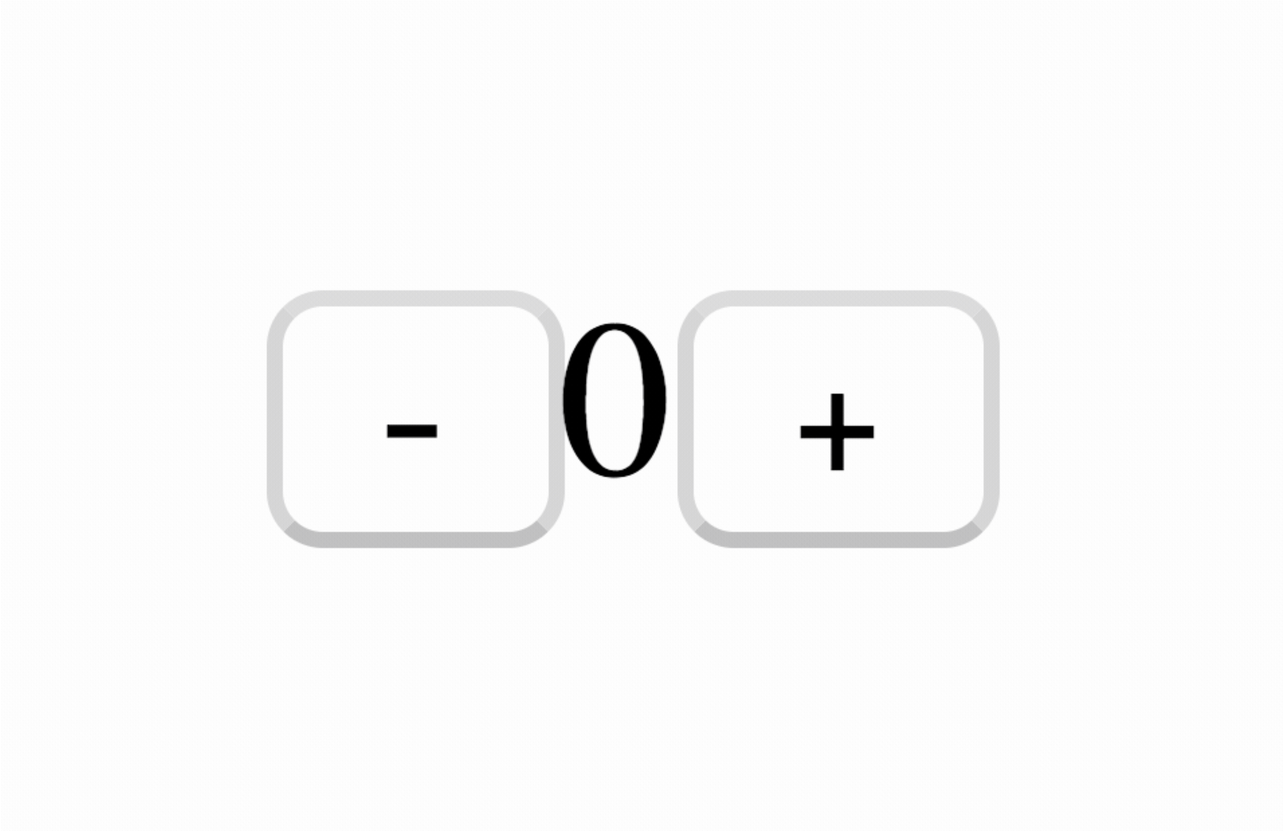

Model
View
Update
Msg

Counter App

```elm
module Main exposing (main)

import Html exposing (..)
import Html.Events exposing (..)


type alias Model =
    { counter : Int }


initialModel : Model
initialModel =
    { counter = 0
    }


type Msg
    = Increment
    | Decrement


update : Msg -> Model -> Model
update msg model =
    case msg of
        Increment ->
            { model | counter = model.counter + 1 }

        Decrement ->
            { model | counter = model.counter - 1 }


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


main =
    Html.beginnerProgram
        { model = initialModel
        , view = view
        , update = update
```

https://ellie-app.com/cn5zm6Sm5a1/0

Counter App



Runtime Errors

Error types in JS

- TypeError
- ReferenceError
- SyntaxError
- EvalError
- URIError
- RangeError

TypeError

```
Undefined is not a function
```

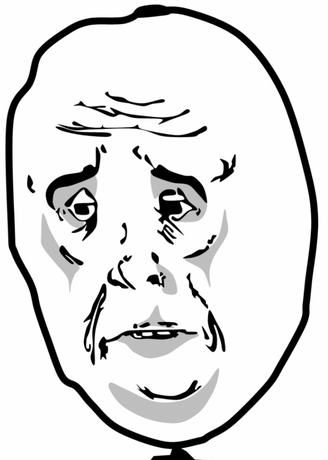
ReferenceError

```
ReferenceError: foo is not defined
```

SyntaxError
EvalError
URIError

RangeError

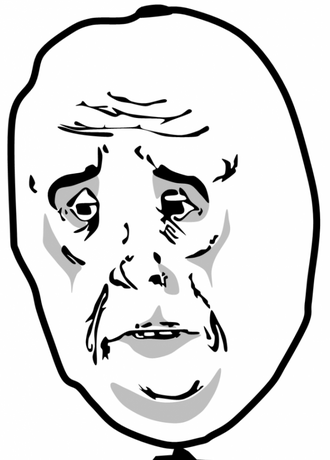
```
RangeError: Maximum call stack size exceeded
```



RangeError

```
RangeError: Maximum call stack size exceeded
```

```
foo : Int → Int
foo a =
    1 + foo a

foo 1 -- BANG
```



```
runtimeErrorsCount == 0 ?
```

Elm

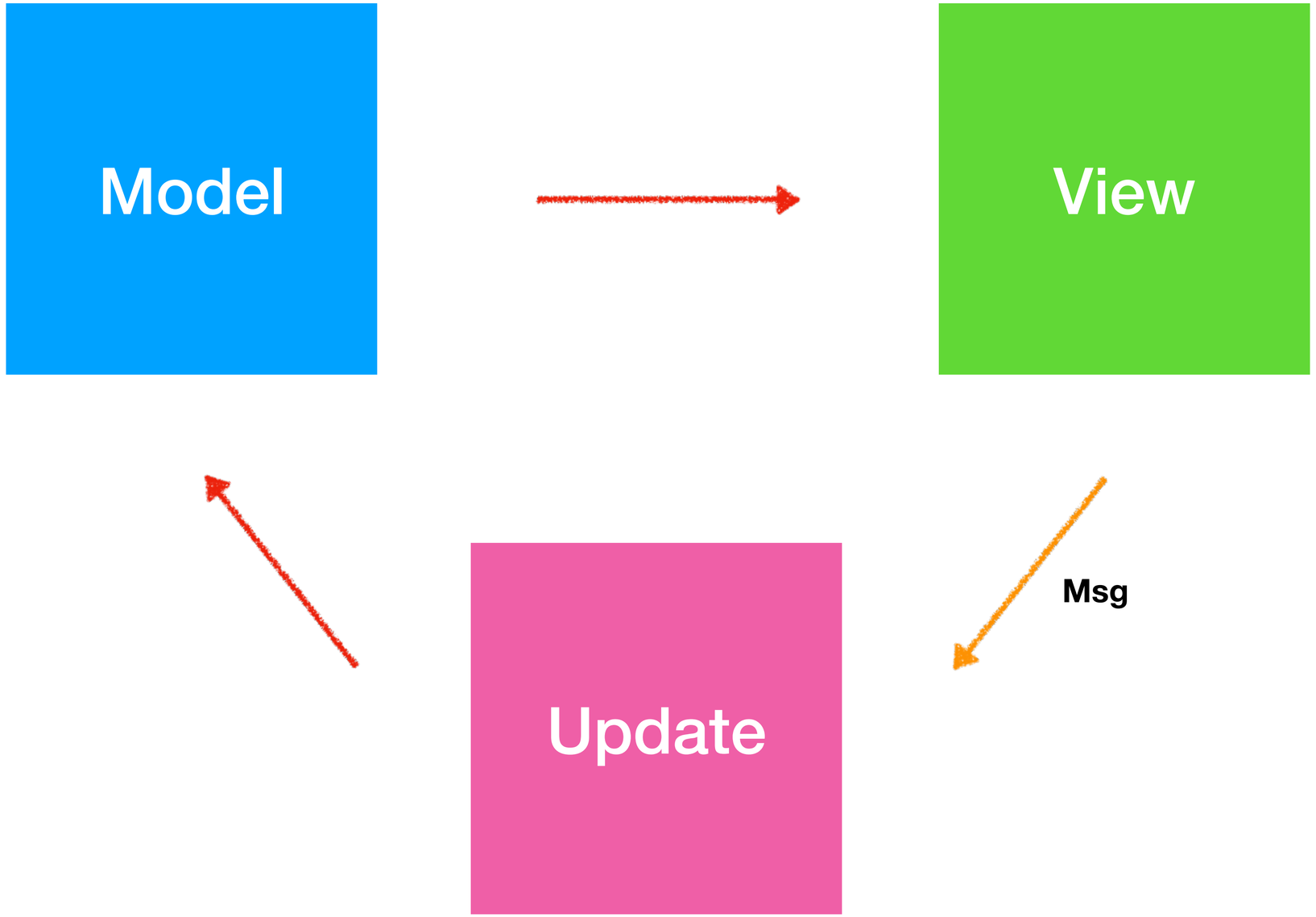
- Иммутабельность и чистые функции
- Контроль над сайд-эффектами (Cmd, Sub)
- Строгий паттерн матчинг
- Нет null и undefined (Maybe)

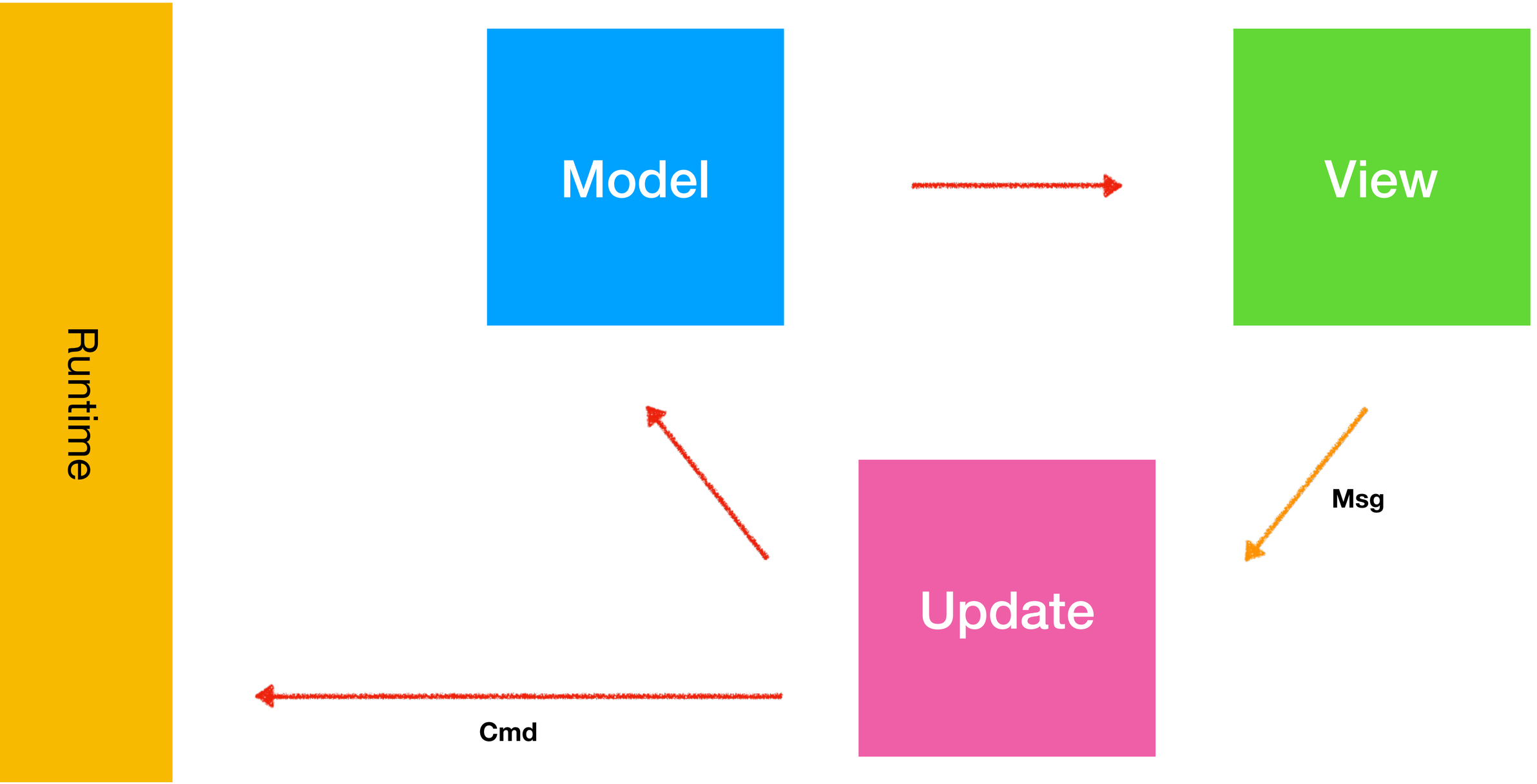
Elm pure functions

```
module Main exposing (main)

import Html exposing (Html, div, text)


counter =
    0


foo a =
    counter + a


main : Html msg
main =
    div []
        [ text (toString (foo 1))
        , text (toString (foo 1))
        ]
```

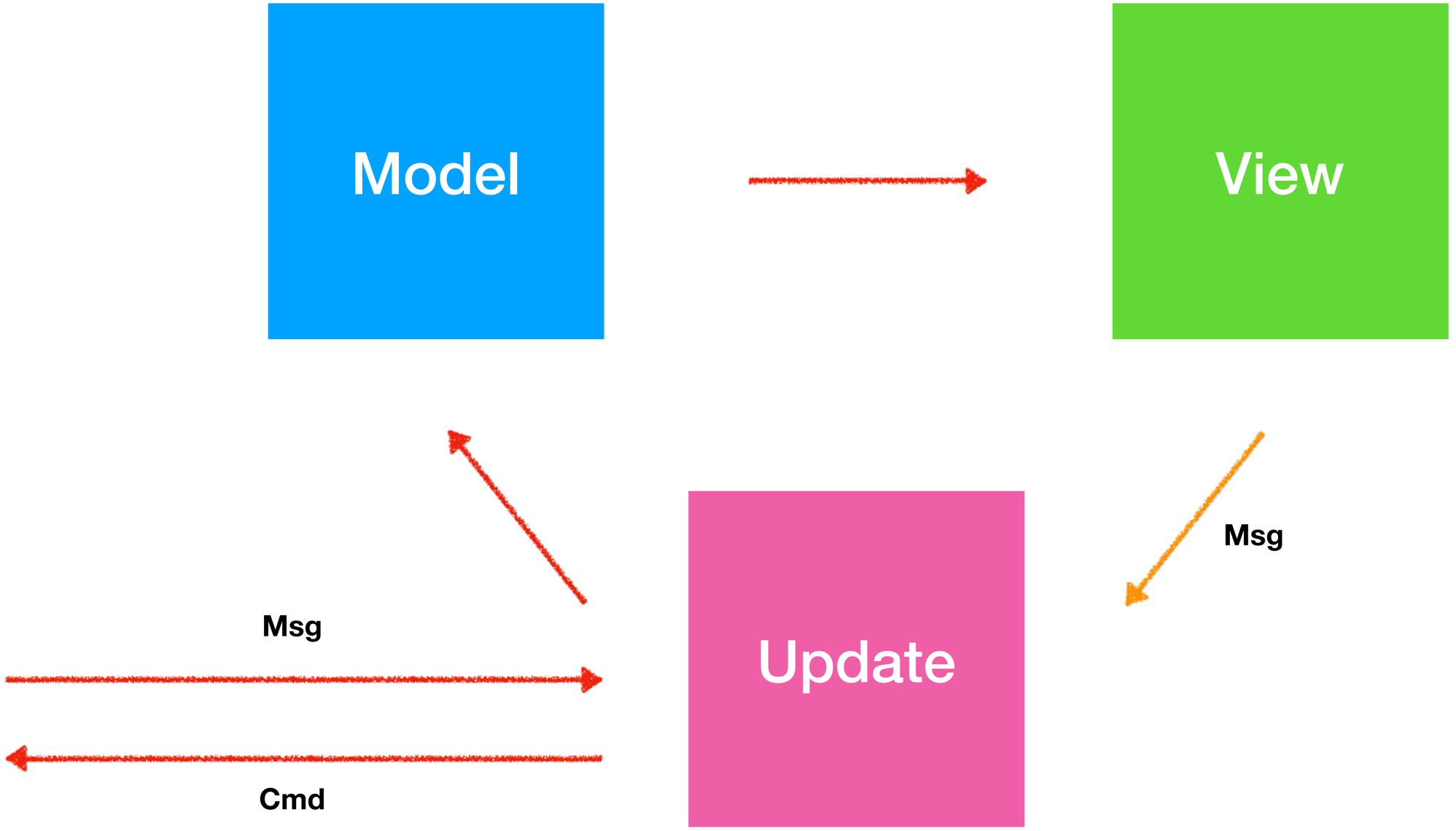
Elm side-effects

```
-- VIEW


view : Model -> Html Msg
view model =
    div []
        [ button [ onClick Roll ] [ text "Roll" ]
        , text (toString model)
        ]



-- UPDATE


update : Msg -> Model -> ( Model, Cmd Msg )
update msg model =
    case msg of
        Roll ->
            ( model, Random.generate OnResult (Random.int 1 6) )

        OnResult res ->
            ( res, Cmd.none )
```


Model
View
Update
Msg


Runtime
Model
View
Update
Msg
Cmd

Runtime



Elm pattern matching

```elm
type Animals
    = Cat
    | Dog
    | Cow


say : Animals → String
say animal =
    case animal of
        Cat →
            "Meow"

        Dog →
            "Bow wow"
```

```
-- MISSING PATTERNS ---------------------------------------------- test.elm

This `case` does not have branches for all possibilities.

14|>    case animal of
15|>        Cat →
16|>            "Meow"
17|>
18|>        Dog →
19|>            "Bow wow"

You need to account for the following values:

    Main.Cow

Add a branch to cover this pattern!

If you are seeing this error for the first time, check out these hints:
<https://github.com/elm-lang/elm-compiler/blob/0.18.0/hints/missing-patterns.md>
The recommendations about wildcard patterns and `Debug.crash` are important!

Detected errors in 1 module.
```

Elm pattern matching

```
say : Animals → String
say animal =
    case animal of
        Cat →
            "Meow"

        Dog →
            "Bow wow"

        Cow →
            "Shazooo"
```

```
say : Animals → String
say animal =
    case animal of
        Cat →
            "Meow"

        Dog →
            "Bow wow"

        _ →
            "I love jQuery"
```

Maybe

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

Maybe

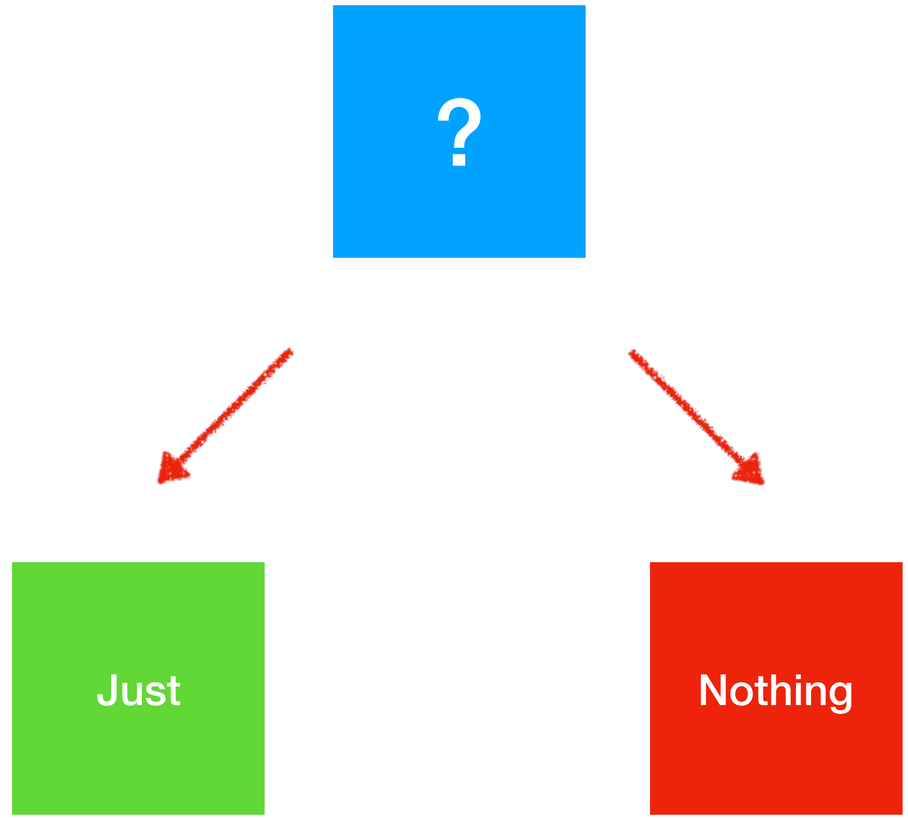
```
type Maybe a
    = Just a
    | Nothing
```

Maybe


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

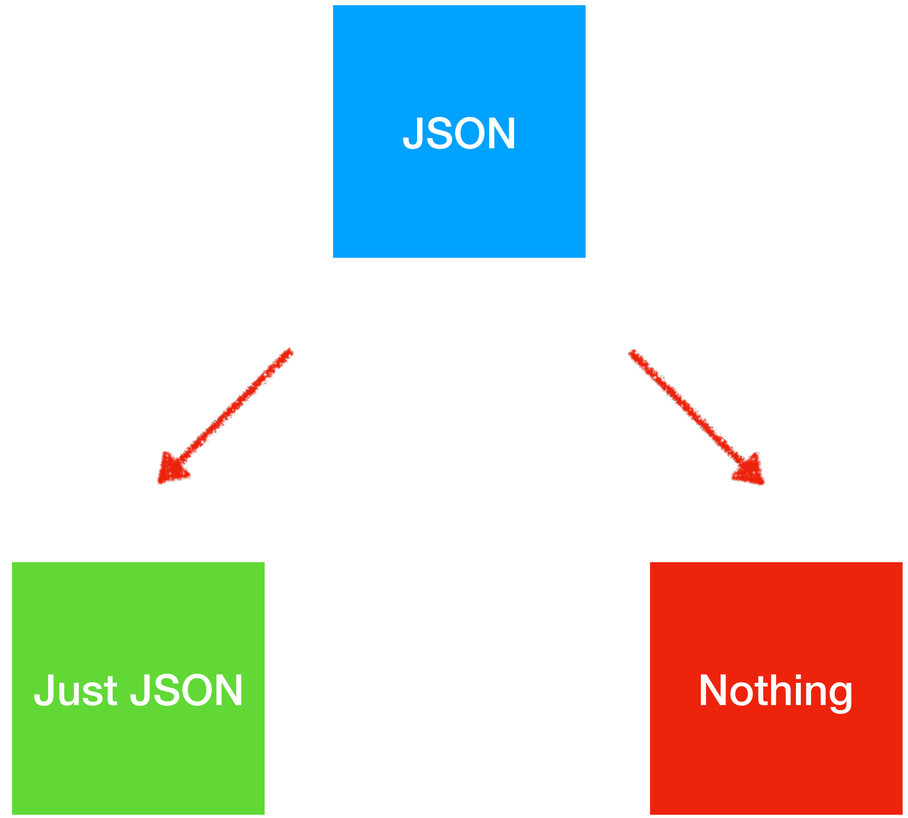
Maybe

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



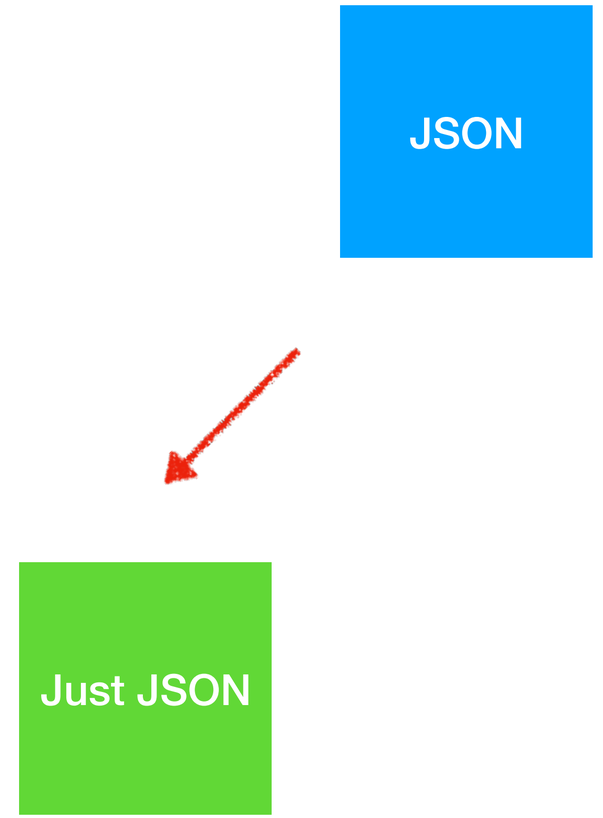
Maybe

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



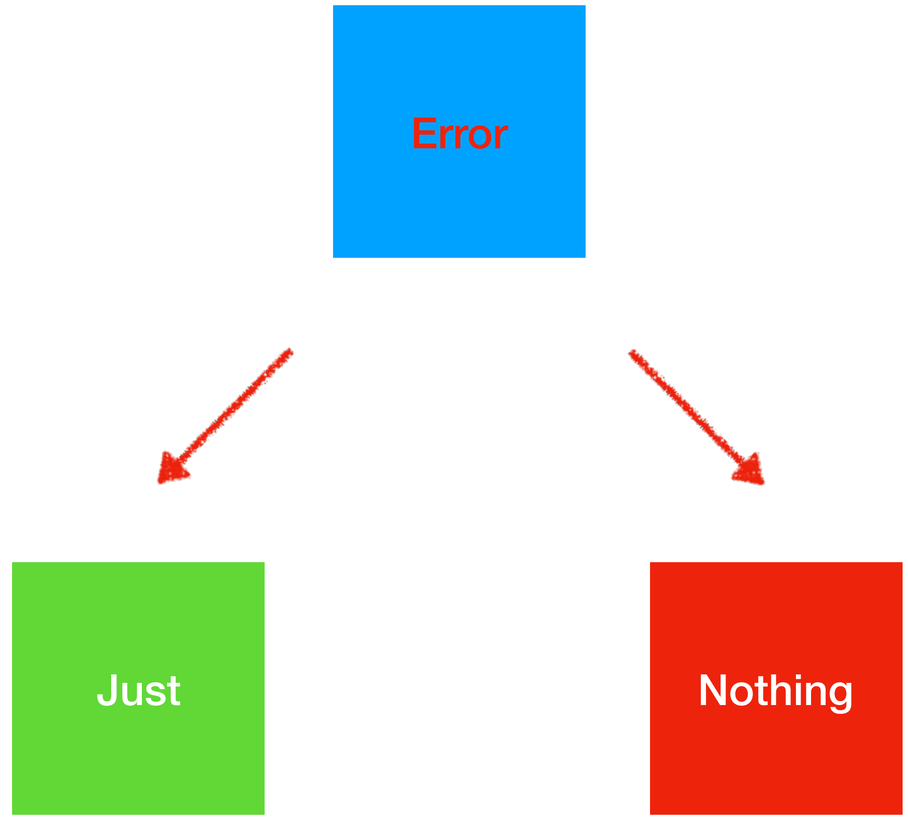
Maybe

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



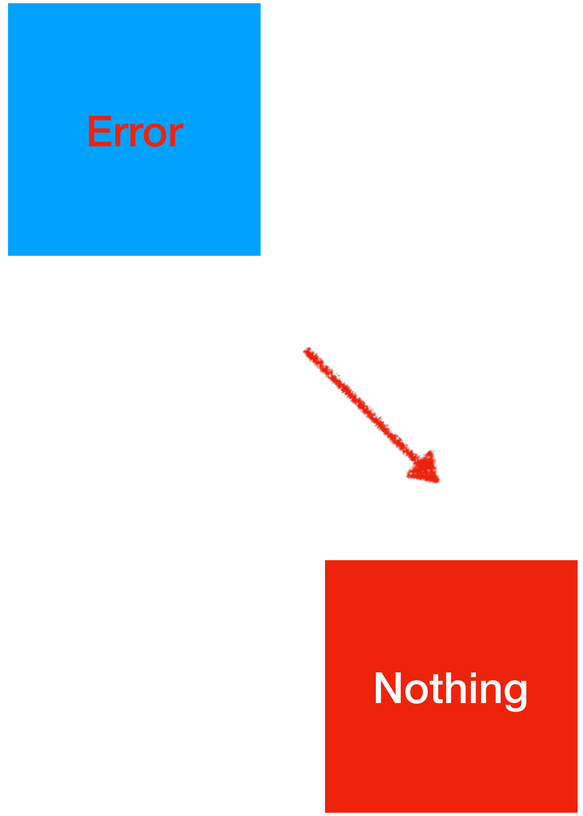
Maybe

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



Maybe

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



Regex

```
import Regex


bang =
    Regex.regex "\\"
```

Division by zero

```
module Main exposing (main)


bang =
    1 % 0
```

Arrays

```
module Main exposing (main)

import Array


bang =
    Array.repeat 33 0 ▷ Array.slice 0 34
```

toString

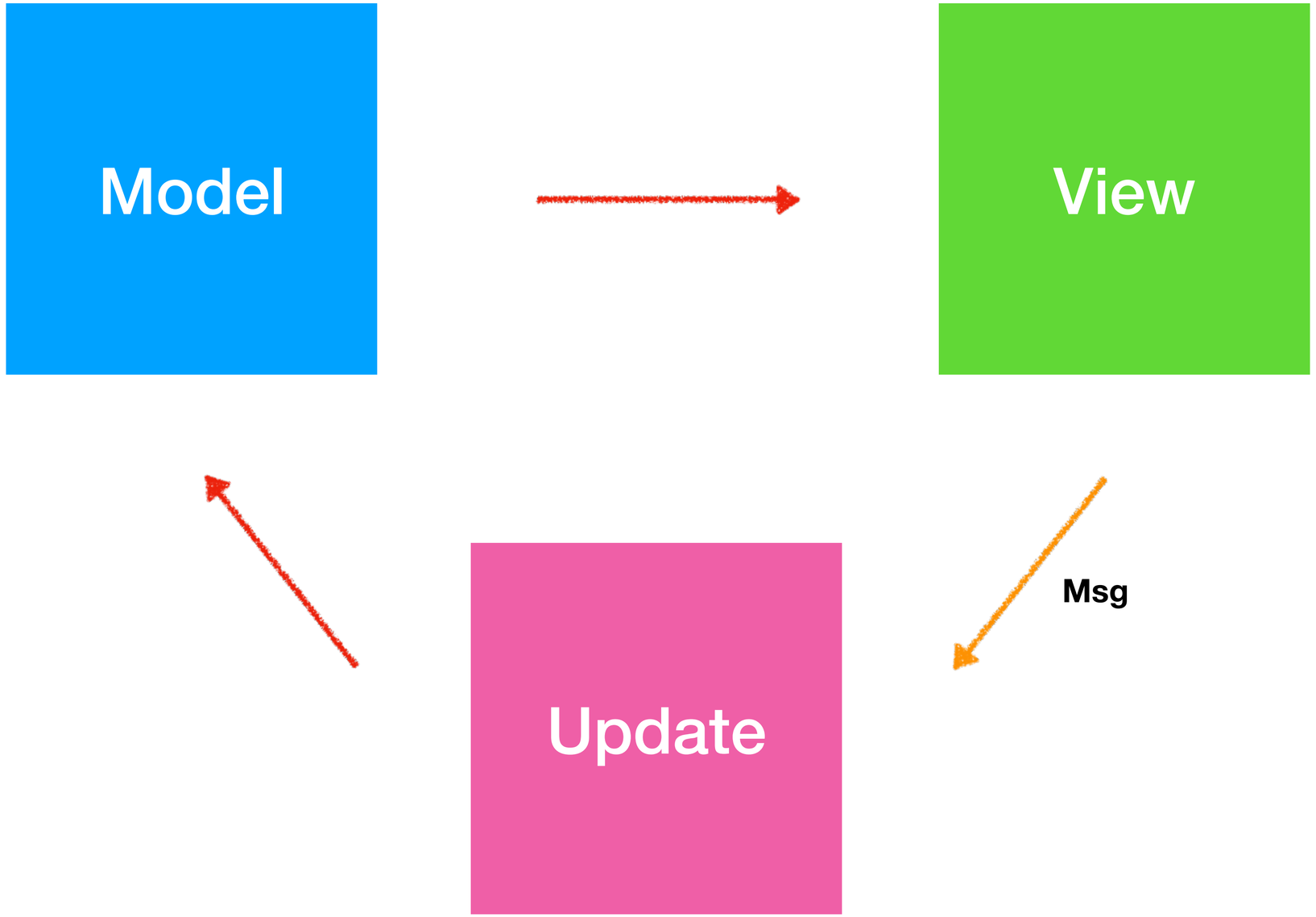
```
module Main exposing (main)


bang =
    toString { ctor = [ 0 ] }
```

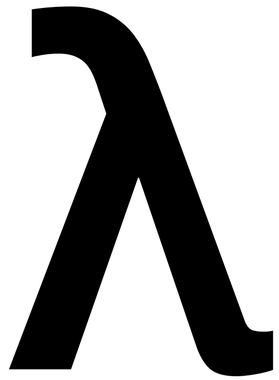
Laziness

```
module Main exposing (main)


bang a b =
    a


bang "foo" (Debug.crash "oops")
```

Html attributes

```
module Main exposing (main)


import Html exposing (Html, text)
import Html.Attributes exposing (type_)



main =
    Html.select [ type_ "number" ] []
```


Model
View
Update
Msg



Рефакторинг

If it compiles, it works

Ecosystem

- Компилятор
- Time-travelling debugger
- Автоформаттер
- Пакетный менеджер

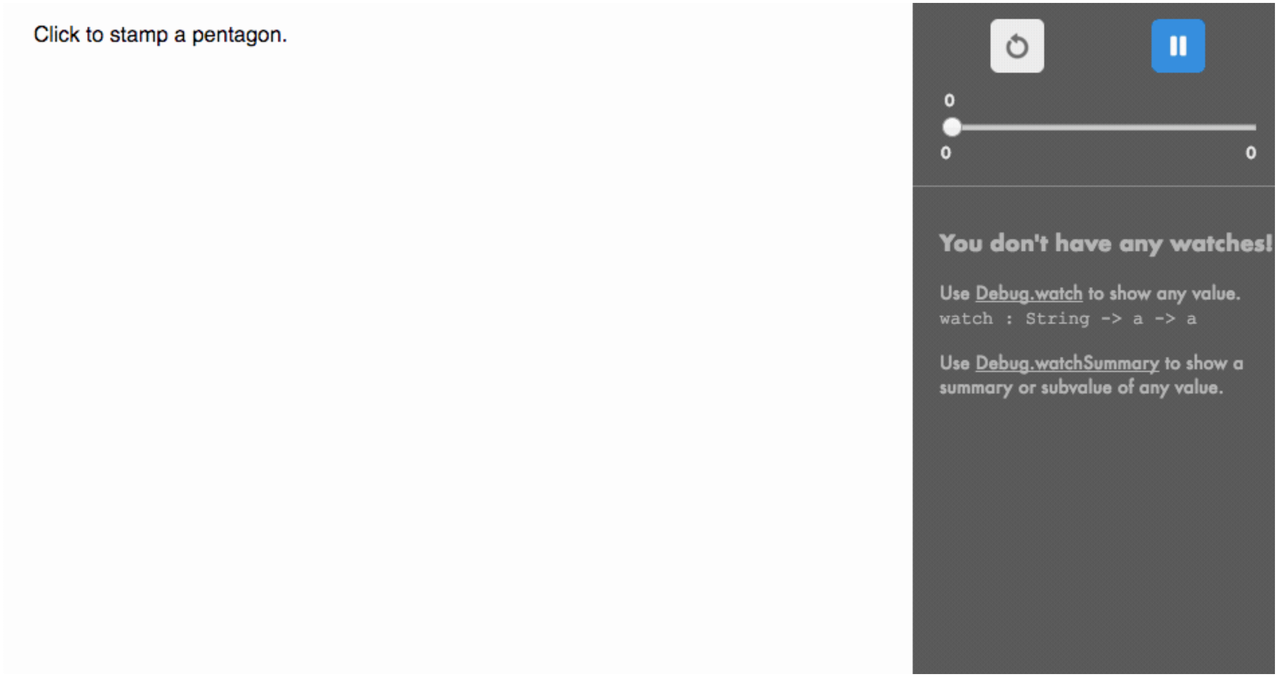
```
-- TYPE MISMATCH ---------------------------------------------- test.elm

The branches of this `if` produce different types of values.

 7|>    if n < 0 then
 8|>        "negative"
 9|>    else
10|>        n

The `then` branch has type:

    String

But the `else` branch is:

    number

Hint: These need to match so that no matter which branch we take, we always get
back the same type of value.

Detected errors in 1 module.
```

```
-- NAMING ERROR ---------------------------------------------------- test.elm

Cannot find variable `dificultNmae`

12|     text dificultNmae
          ^^^^^^^^^^^^^

Maybe you want one of the following?

    difficultName

Detected errors in 1 module.
```

Elm-package

Enforsed semantic versioning

```
Comparing exdis/elm-sample-package 1.0.0 to local changes ...
This is a MAJOR change.

------ Changes to module Sample - MAJOR ------

    Changed:
      - foo : Int → Int → Int
      + foo : Int → Int → Int → Int
```


Click to stamp a pentagon.
0
0
0
You don't have any watches!
Use Debug.watch to show any value.
watch : String -> a -> a
Use Debug.watchSummary to show a summary or subvalue of any value.

Benchmarks

Duration

Name	elm-v0.18.0-keyed	vue-v2.5.3-keyed	react-v16.1.0-redux-v3.7.2-keyed
create rows Duration for creating 1000 rows after the page loaded.	177.9 ± 7.2 (1.1)	169.2 ± 3.6 (1.0)	206.2 ± 9.0 (1.2)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	178.9 ± 10.9 (1.1)	161.8 ± 3.9 (1.0)	175.1 ± 4.2 (1.1)
partial update Time to update the text of every 10th row (with 5 warmup iterations) for a table with 10k rows.	99.4 ± 9.4 (1.0)	168.1 ± 7.4 (1.7)	97.8 ± 5.6 (1.0)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	11.0 ± 5.4 (1.0)	9.8 ± 2.5 (1.0)	10.1 ± 3.5 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	18.3 ± 4.6 (1.0)	21.8 ± 4.5 (1.2)	19.0 ± 4.4 (1.0)
remove row Duration to remove a row. (with 5 warmup iterations).	62.1 ± 5.6 (1.3)	52.5 ± 1.8 (1.1)	49.5 ± 1.8 (1.0)
create many rows Duration to create 10,000 rows	1636.4 ± 22.7 (1.1)	1521.4 ± 55.7 (1.0)	2048.5 ± 58.8 (1.3)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	271.8 ± 18.1 (1.0)	338.4 ± 10.3 (1.2)	300.7 ± 30.9 (1.1)
clear rows Duration to clear the table filled with 10.000 rows.	222.6 ± 4.7 (1.0)	240.9 ± 11.4 (1.1)	227.7 ± 8.6 (1.0)
startup time Time for loading, parsing and starting up	29.1 ± 1.3 (1.0)	48.4 ± 2.4 (1.7)	68.6 ± 2.3 (2.4)
slowdown geometric mean	1.05	1.17	1.17

Memory

Name	elm-v0.18.0-keyed	vue-v2.5.3-keyed	react-v16.1.0-redux-v3.7.2-keyed
ready memory Memory usage after page load.	3.7 ± 0.1 (1.0)	3.6 ± 0.1 (1.0)	4.2 ± 0.1 (1.2)
run memory Memory usage after adding 1000 rows.	7.6 ± 0.0 (1.1)	7.2 ± 0.0 (1.0)	8.5 ± 0.0 (1.2)
update eatch 10th row for 1k rows (5 cycles) Memory usage after clicking update every 10th row 5 times	7.7 ± 0.0 (1.1)	7.3 ± 0.0 (1.0)	9.6 ± 0.0 (1.3)
replace 1k rows (5 cycles) Memory usage after clicking create 1000 rows 5 times	7.8 ± 0.0 (1.1)	7.3 ± 0.0 (1.0)	11.0 ± 0.0 (1.5)
creating/clearing 1k rows (5 cycles) Memory usage after creating and clearing 1000 rows 5 times	4.1 ± 0.0 (1.1)	3.8 ± 0.0 (1.0)	5.8 ± 0.0 (1.5)

http://www.stefankrause.net/js-frameworks-benchmark7/table.html

Недостатки Elm

- Размер комьюнити
- Релизный цикл
- Вес рантайма (~100 Кб)
- Приватные пакеты
- Нет Server-side rendering
- Нет монад ¯_(ツ)_/¯

Roadmap

- Server-side rendering
- Tree shaking
- Code splitting
- Lazy loading

```
runtimeErrorsCount == 0?
```

```
runtimeErrorsCount == 0!
```

After 2 years and 200,000 lines of production @elmlang code, we got our first production runtime exception.

(We wrote code that called Debug.crash and shipped it. That function does what it says on the tin.)

In that period, our legacy JS code has crashed a mere 60,000 times.

–Richard Feldman

http://elm-lang.org/

http://package.elm-lang.org/

https://www.elm-tutorial.org/

http://elmprogramming.com/

Elm In Action (MEAP)

Elm for beginners

How to use Elm at work

Спасибо!

http://elm-lang.org/

http://package.elm-lang.org/

https://www.elm-tutorial.org/

http://elmprogramming.com/

Elm In Action (MEAP)

Elm for beginners

How to use Elm at work