

@ElmNetherlands



"KNOW YOUR CODE"

2017-03-08, Mats Stijlaart (@stil4m)

AVISI

- Pushed Elm
- Four teams use Elm
- Elm Guild

"Code is the Easy Part"

Language point of view

"Code is the Important Part"

Consumer point of view

"A TOOL THAT IS NOT THERE-ISH"

Idiomatic code

Code Quality

Syntax

Education

"Dreamland"

IDIOMATIC CODE

SYNTACTIC STRUCTURE

```
inc1 x y =  
  x + y
```

```
inc2 x y =  
  (+) x y
```

```
inc3 =  
  (+)
```

```
inc4 x =  
  (+) x
```

Which form would you use?

Which form should you use?

```
incAllByWith1 y xs =  
  List.map (\x -> y + x) xs
```

```
incAllByWith2 y xs =  
  List.map ((+) y) xs
```

But it is not that easy...

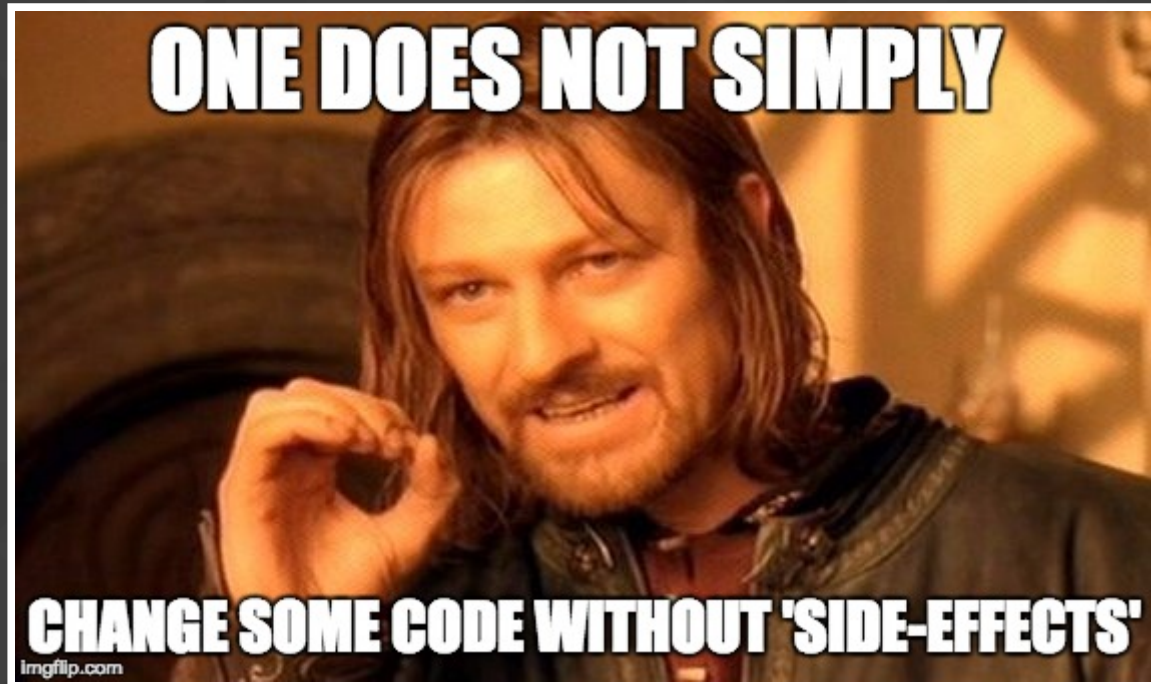
QUESTION

"What is idiomatic Elm code?"

It is not 42

I believe "it depends"

- Team qualities
- Complexity of code
- *Probably a bit more...*



```
incMaybe : Maybe Int -> Int
```

```
incMaybe x =
```

```
  case x of
```

```
    Just n ->
```

```
      n + 1
```

```
    Nothing ->
```

```
      42 * 42
```

```
incMaybe2 : Maybe Int -> Int
```

```
incMaybe2 x =
```

```
  x
```

```
  |> Maybe.map ((+) 1)
```

```
  |> Maybe.withDefault (42 * 42)
```

Seems the same right?

The value constructed for the `Maybe.withDefault` is
always computed!

```
var _user$project$Example$incMaybe = function (x) {  
  var _p0 = x;  
  if (_p0.ctor === 'Just') {  
    return _p0._0 + 1;  
  } else {  
    return 42 * 42;  
  }  
};
```

```
var _user$project$Example$incMaybe2 = function (x) {  
  return A2(  
    _elm_lang$core$Maybe$withDefault,  
    42 * 42,  
    A2(  
      _elm_lang$core$Maybe$map,  
      F2(  
        function (x, y) {  
          return x + y;  
        }  
      ))(1),  
    x));  
};
```

IDOMATIC CODE

- Not yet defined. I would like to start the discussion.
- Be consistent. Especially if you try to scale within a team.
- Be careful what you do. What seems the same, is not always the same.

QUALITY CODE

SOME QUESTIONS?

Do you use the Random package?

If you use any seed less than 53,668 and generate one bool, it will be True – if you're using core's Random module. More sophisticated statistical tests spot patterns in the "random" numbers almost immediately.

Package mgold/elm-random-pcg

Have you ever used `Regex.regex` with a variable?

elm-lang/

[elm-lang/animation-frame](#) ... 1.0.1 — Overview
Somewhat low-level API for smooth animations with `requestAnimationFrame`.

[elm-lang/core](#) ... 3.0.0 ... 4.0.5 ... 5.1.1 — Overview
Elm's standard libraries

[elm-lang/dom](#) ... 1.1.1 — Overview
DOM helpers for managing focus and scrolling.


```
import Regex

myPattern =
  Regex.regex "[+]"

matcher : String -> String -> Bool
matcher x y =
  Regex.contains (Regex.regex x) y
```

*We need something that helps us
"know stuff" about our code, and
encode this in tooling, CI and IDEs.*

SOME 'SMELLY' CODE

```
elm-package.json  
src/Models.elm  
src/Decoders.elm  
src/Rest.elm  
src/MyView.elm
```

```
module Models exposing (..)
```

```
...
```

```
type Person =  
    Person String Int
```

```
...
```

```
module Decoders exposing (..)
```

```
...
```

```
decodePerson : Decoder Person
```

```
decodePerson =
```

```
...
```

```
module MyView exposing (..)

...

viewPerson : Person -> Html msg
viewPerson (Person name age) =
    div
        [ if age >= 18 then
            text "Can buy beer"
          else
            text "Can't buy beer"
        ]
```

```
module MyView exposing (..)

...

viewPerson : Person -> Html msg
viewPerson person =
    div
        [ if canBuyBeer person then
            text "Can buy beer"
          else
            text "Can't buy beer"
        ]

canBuyBeer : Person -> String
canBuyBeer (Person _ age) =
    age >= 18
```


The code elements have 'distance'

```
viewPerson <-- small --> canBuyBeer  
canBuyBeer <-- big --> type Person
```

Wouldn't we like:

```
viewPerson <-- big --> canBuyBeer  
canBuyBeer <-- small --> type Person
```

'BETTER' CODE

```
module Person exposing (..)

type Person =
  ...

canBuyBeer : Person -> Bool
...

decode : Decoder Person
...
```

Can we automatically recognize 'bad' design?

WHAT ELSE WOULD WE NEED?

- Dead code
- Big functions
- Duplications (within project and ecosystem)
- Dependency graphs
- *And whatever helps you*

CODE QUALITY

- Hard to know if you do it 'right'.
- We can improve on this with the appropriate tools.
- For now we have to do it by eye.

EDUCATION

ELM-ANALYSE

<https://github.com/stil4m/elm-analyse>

It is all about improving your code.

Learn by example: Where to put parens?

```
max x y =  
  if (x < y) then  
    y  
  else  
    x
```

```
reverseName record =  
  { record | name = (String.reverse record.name) }
```


Redefining variables

```
update model =  
  let  
    model = { model | counter = model.counter + 1 }  
  in  
    if model.show then  
      { model | model.updated = True }  
    else  
      model
```

Unused stuff

Continuous Integration

DREAMLAND

AST

- Analyse structure of code.
- Fix code
- Rewrite code

ELM ON THE JVM?

```
package nl.stil4m;

import java.lang.Function1;
import java.lang.Function2;
import java.lang.Function3;

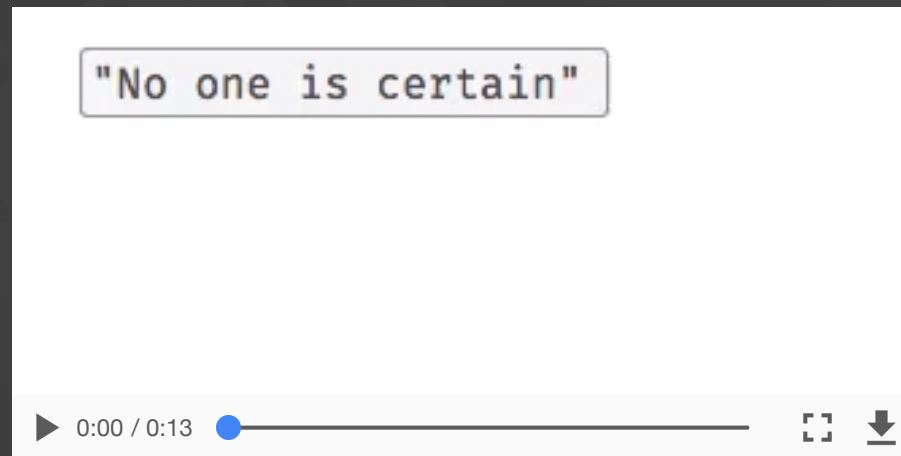
public class MaybeMod {

    public static <A> Function2<A, Maybe<A>, A> withDefault() {
        return (Function2<A, Maybe<A>, A>) (a, aMaybe) -> {
            if (aMaybe instanceof Just) {
                Just<A> j = (Just) aMaybe;
                return j.arg0;
            } else {
                return a;
            }
        };
    }
}
```

MUTATION TESTING?

```
incIfOdd n =  
  case n % 2 == 0 of  
    True ->  
      n  
  
    False ->  
      n + 1
```

UNISON?



<http://unisonweb.org/2015-05-07/about.html>

WHATEVER YOU CAN THINK OF...

FUTURE

- There is a lot of work to do
- Join discussions
- Call for these tools

@stil4m

Twitter / Slack / GitHub