

ELM @ FUNCTIONAL ROTTERDAM

2016-11-17, Mats Stijlaart (@stil4m)

SHORT WHO AM I

- **Past:**
Bachelor Informatica HAN (Arnhem) 2013
- **Current:**
Software Developer at Avisi and Software Engineering
Master at UvA
- **Elm:**
Organizer of Elm Netherlands. Advocating Elm at
Avisi.

ELM



WHAT IS IT?

- ML Based
- Functional
- Pure
- Compiles to Javascript
- *"We can have nice things"*

WHAT IS IT NOT?

- No type classes
- No side effects

SYNTAX



PRIMITIVES

```
-- A comment
```

```
1           -- Int
3.14        -- Float
"Hello world" -- String
True        -- Bool
[1, 1, 2, 3, 5, 8] -- List
(1, "One")    -- Tuple as (Int, String)
```

```
-- Record
{ name = "Mats"
, age = 26
}
```

FUNCTIONS

```
increment : Int -> Int
increment x =
    x + 1
```

```
-- increment 5 == 6
```

```
addExclamationMarks : String -> Int -> String
addExclamationMarks input times =
    input ++ String.repeat times "!"
```

```
-- addExclamationMarks "hello" 3 == "hello!!!"
```


ALIAS

```
type alias StatusCode = Int
```

```
type alias Coordinate = (Float, Float)
```

RECORDS

```
type alias Person =  
  { name : String  
    , age : Int  
  }
```

```
mats : Person  
mats = { name = "Mats", age = 17 }
```

```
peter : Person  
peter = Person "Peter" 12
```

```
-- Will not compile  
johnDoe : Person  
johnDoe = { name = "John" }
```

TYPE

```
type TrafficLight
  = Green
  | Yellow
  | Red
```

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

CASE

```
canDrive : TrafficLight -> Bool
canDrive light =
  case light of
    Green -> True
    Yellow -> False
    Red -> False
```

```
canDrive : TrafficLight -> Bool
canDrive light =
  case light of
    Green -> True
    _ -> False
```

COOL STUFF

```
toCelcius : Float -> Float
```

```
toCelcius x =
```

```
  x
```

```
  |> (+) -32.0
```

```
  |> (/) 9
```

```
  |> (*) 5
```

COOL STUFF (2)

```
toCelcius : Float -> Float  
toCelcius =  
    (+) -32.0 >> (/) 9 >> (*) 5
```

COOL STUFF (3)

```
type alias Person = { name : String }
```

```
getNames : List Person -> List String
```

```
getNames x =  
    List.map .name x
```

```
getNames2 : List Person -> List String
```

```
getNames2 =  
    List.map .name
```

WHY?

**ARE YOU TELLING
ME**

**UNDEFINED IS NOT A
FUNCTION?**

memegenerator.net



*Runtime exceptions reduced to
(nearly) zero*

```
1 + "1"
```

```
1 + "1"  
  ^^^
```

(+) is expecting the right side to be a:

number

But the right side is:

String

Batteries included

- The Elm Architecture (TEA)
- No HTML and JS, just Elm

Designed to solve common problems.

- Package manager
- Formatting

We can assume certain things from our software, thus it would require less testing.



HTML


```
<div class="my-container">  
  <h1>Title</h1>  
  <ul>  
    <li>First</li>  
    <li>Second</li>  
    <li>Third</li>  
  </ul>  
</div>
```

```
<tag attrs*=" ">  
  <tag>TEXT</tag>  
  <tag attrs*=" ">  
    <tag attrs*=" ">TEXT</tag>  
    <tag attrs*=" ">TEXT</tag>  
    <tag attrs*=" ">TEXT</tag>  
  </tag>  
</tag>
```

```
<tag attrs*=" ">  
  CHILDREN*  
</tag>
```

```
-- ? : List Attribute -> List Html -> Html
-- text : String -> Html
```

```
div [ class "my-container"
      [ h1 [] [ text "Title" ]
        , ul []
          [ li [] [ text "First" ]
            , li [] [ text "Second" ]
            , li [] [ text "Third" ]
          ]
      ]
```

```
div [ class "my-container"]  
    [ h1 [] [ text "Title"]  
      , ul [] (List.map listItem items)  
    ]
```

```
items : List String  
items = ["First", "Second", "Third"]
```

```
listItem : String -> Html  
listItem x =  
    li [] [ text x]
```

THE ELM ARCHITECTURE

Building a simple counter

COMPOSING SOMETHING BIGGER

Example

NICE THINGS

- Elm Format
- Elm Package
- Elm Css

SOURCE

- <http://elm-lang.org>
- <https://github.com/isRuslan/awesome-elm>
- <http://www.elmweekly.nl/>
- Elm Lang Slack

QUESTIONS?

CODING DOJO