

Practical ReasonML

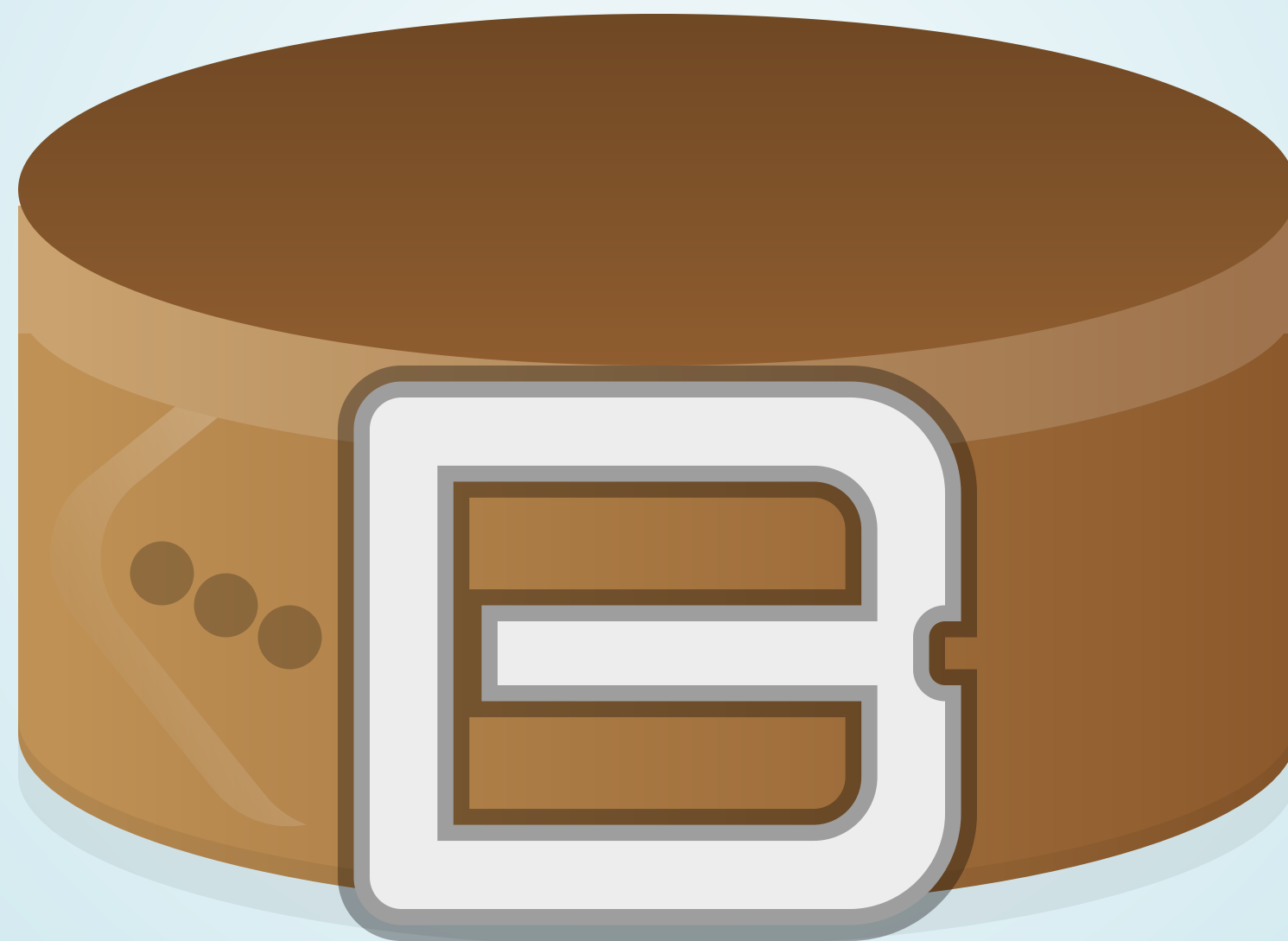
 codecentric

@MarcoEmrich

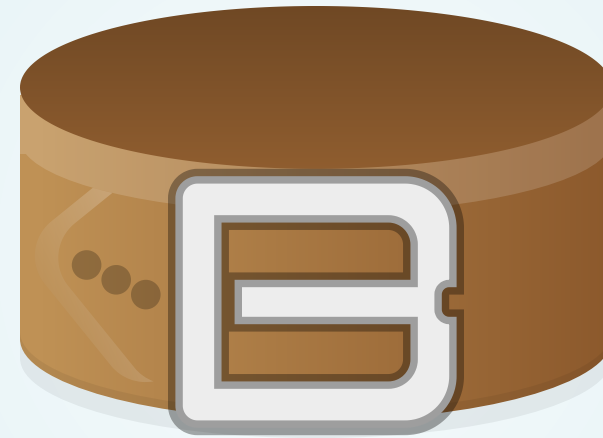


BS

BUCKLESCRIPT



REASONML / OCAML

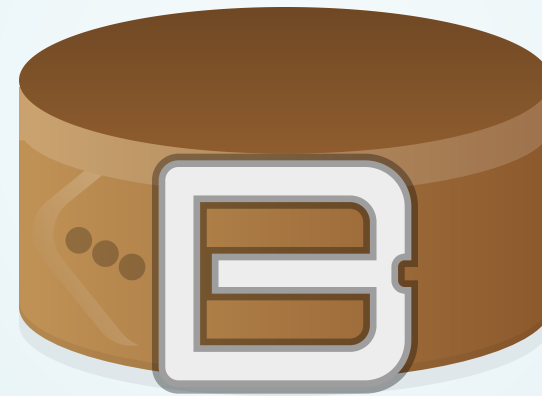


JAVASCRIPT

REASONML / OCAML



BSB-NATIVE



NATIVE / BACKEND / MICRO KERNEL / REACT NATIVE

TRANSPILE TO JS



BLOOMBERG



- 176 Offices
- 19000 Employees

HELLO WORLD

```
npm -g i bs-plattform  
bsb -init sample -theme basic-reason
```

DEMO

COMPUTER OR HUMAN?

stolen from Sean Grove

WRITTEN BY A MACHINE? (1/2)

```
var Int_map = require("./int_map.js");
function test() {
  var m = /* Empty */0;
  for(var i = 0; i <= 10000000; ++i) {
    m = add(i, i, m);
  }
  for(var j = 0; j <= 10000000; ++j) {
    find(j, m);
  }
  return /* () */0;
}
```

WRITTEN BY A MACHINE? (2/2)

```
var Pervasives = require("bs-plattform/lib/js/pervasives");
var Http       = require("http");

var hostname = "127.0.0.1";

function create_server(http) {
  var server = http.createServer(function (_, resp) {
    resp.statusCode = 200;
    resp.setHeader("Content-Type", "text/plain");
    return res.end("Hello World\n");
  });
  return server.listen(3000, hostname, function () {
    console.log("Server running at http://") +
    ...
  });
}
```

SOLUTION

GENERATES

IDOMATIC JS

REASON

?

TypeScript





DEADLINES?

TIME PRESSURE?

ENEMIES OF THE TYPE-SYSTEM

```
: any
```

```
@ts-ignore
```

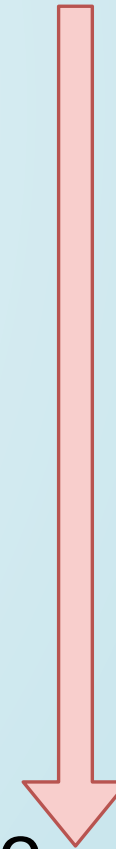
AnyScript™

JS-TRANSPILER LANGUAGES

Purity Enforcement /
Type Coverage



Interoperability with the
JavaScript Ecosystem





ELM

INDUSTRIAL

16232

x 1000g

5 lt.

ADOR CREM

Virginia
Evolutio

DOR CREMA

Virginia

Virginia
LAVALOZAS CONCENTRADA

Hit
FUERA

SAFETYGEN

JS

DEFEQUIM

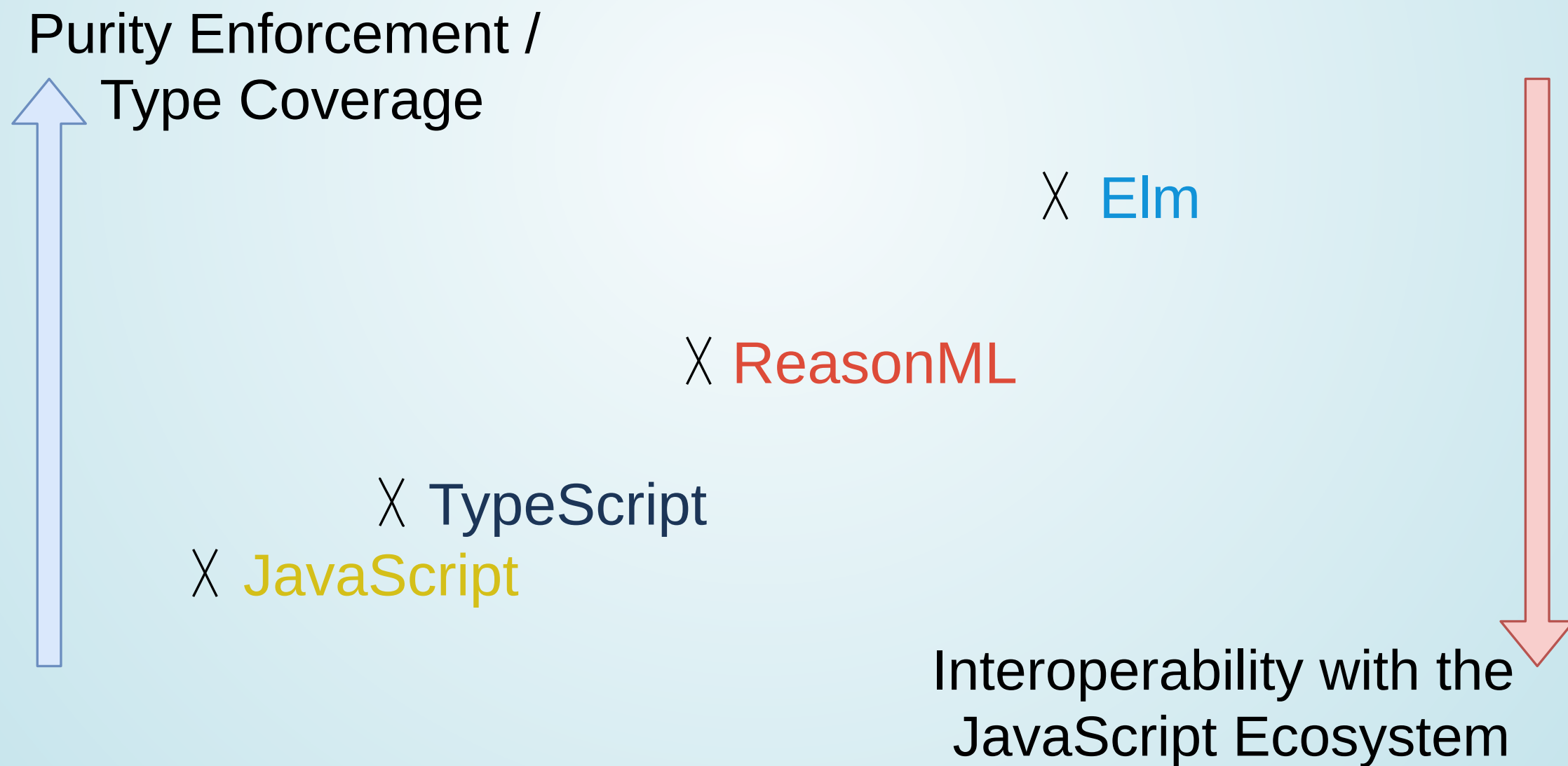
DESTRUCTO DE FECA

5 L

ELM



JS-TRANSPILER LANGUAGES



A collection of various tools is arranged on a dark wooden surface. The tools include a hammer with a wooden handle, a hammer with a metal handle, a scythe, a pair of pliers, a pair of work gloves, a pair of axes, a utility knife, a flashlight, a metal cup, a metal bell, and a circular metal object with a logo. The text "CHOOSE BEST TOOL FOR THE JOB" is overlaid in the center in a white, bold, sans-serif font.

CHOOSE
BEST TOOL
FOR THE JOB

Need a lot of crazy JS-Libraries?

Need no JS-Libraries at all?

Somewhere in between?

REASON

THREE WISHES





BENEFIT OF
FUNCTIONAL
PROGRAMMING

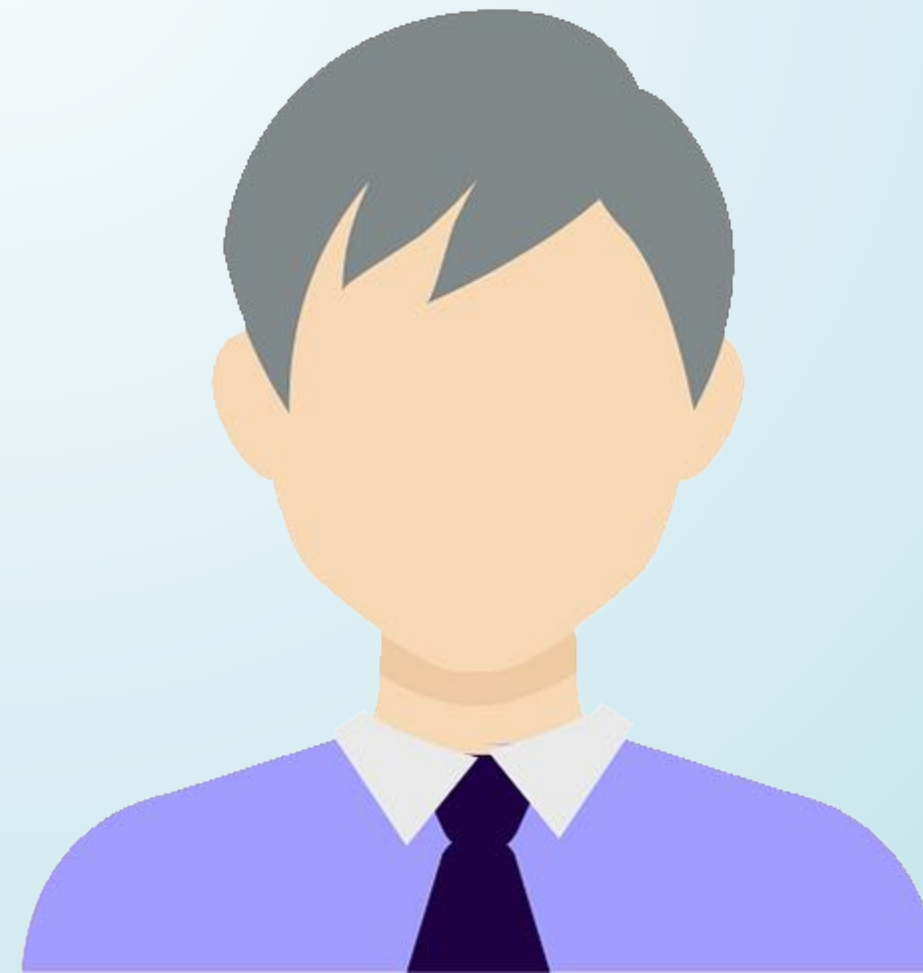
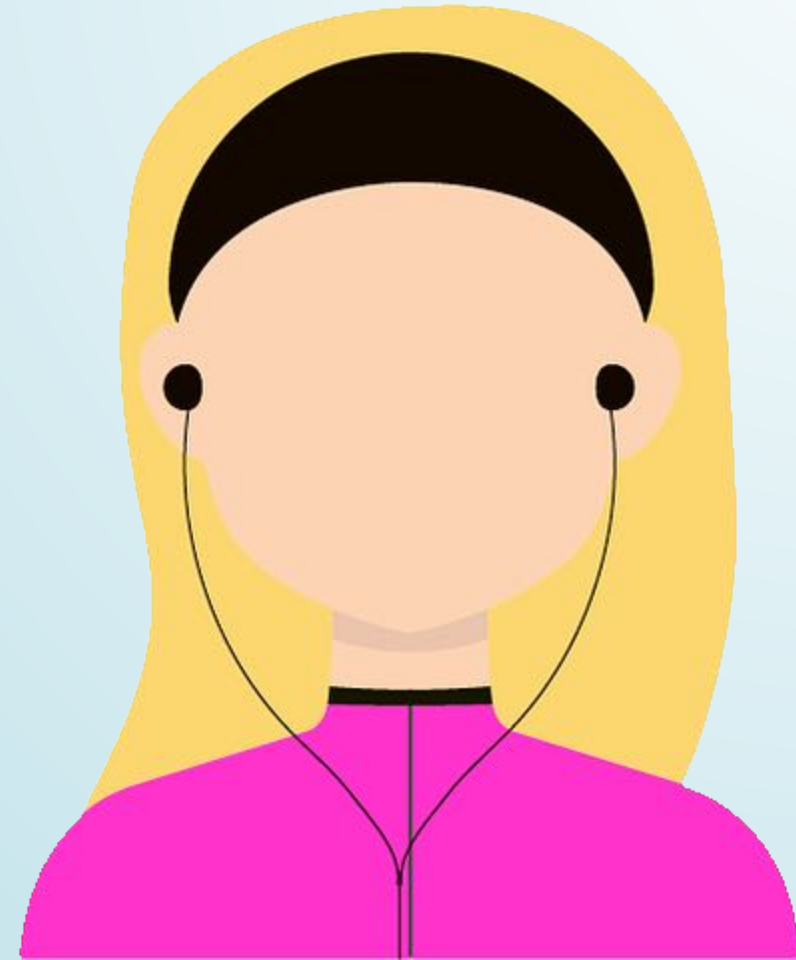
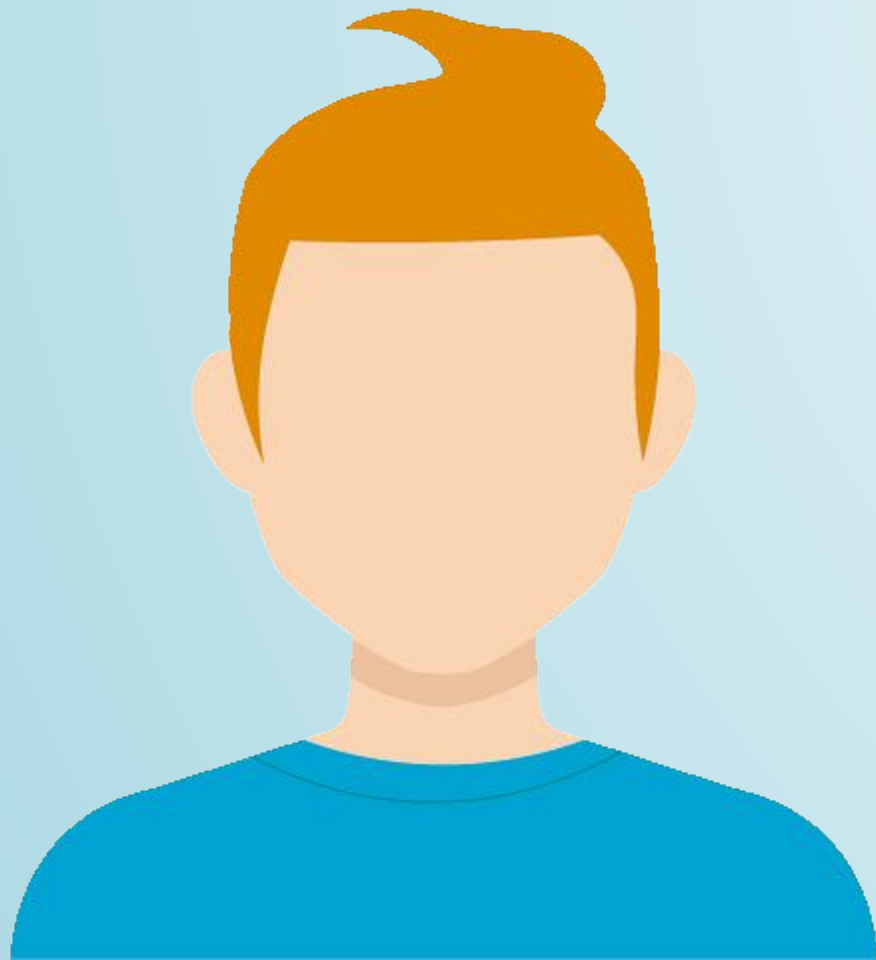
... (IN THE FRONT END)

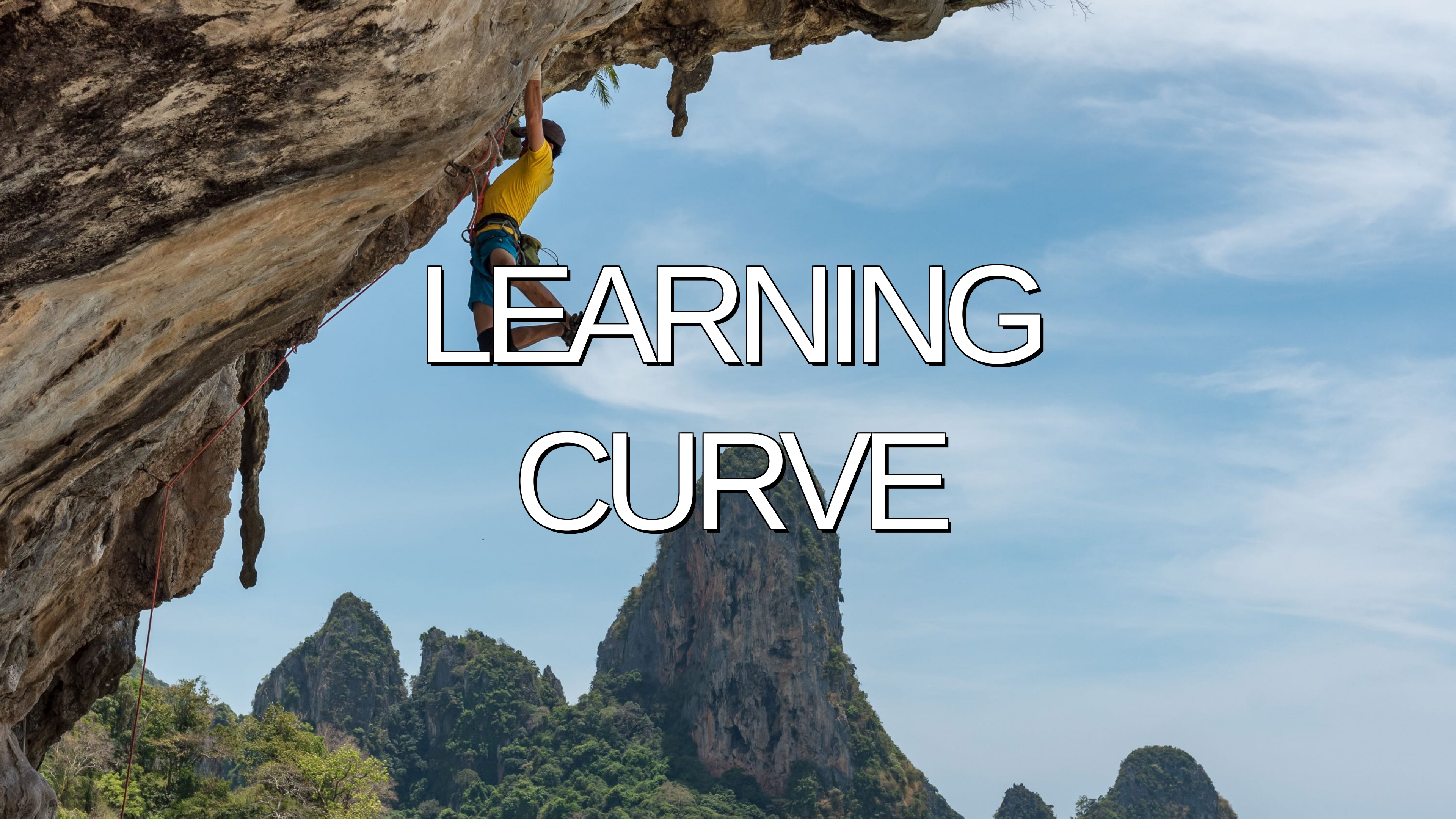


PURESCRIPT, ELM,
HASKELL ...


BUT...

... CONVINCE COLLEAGUES





LEARNING CURVE

A dark, atmospheric scene featuring pumpkins, lit candles, and a hand holding a skull, with the text "CURSE OF THE MONAD" overlaid. The scene is dimly lit, with the primary light source being several lit candles. The candles are of various sizes and are scattered across the foreground and midground. The background is filled with dark, tangled branches and leaves, creating a sense of a dense, possibly outdoor setting. In the center, a hand is visible, holding a small, light-colored skull. The overall mood is mysterious and ominous.

**CURSE OF THE
MONAD**

Zygohistomorphic prepromorphisms

Used when you really need both semi-mutual recursion and history and to repeatedly apply a natural transformation as you get deeper into the functor. Zygo implements semi-mutual recursion like a zygomorphism. Para gives you access to your result à la paramorphism.

```
import Control.Morphism.Zygo
import Control.Morphism.Prepro
import Control.Morphism.Histo
import Control.Functor.Algebra
import Control.Functor.Extras

zygoHistoPrepro
  :: (Unfoldable t, Foldable t)
  => (Base t b -> b)
  -> (forall c. Base t c -> Base t c)
  -> (Base t (EnvT b (Stream (Base t)) a) -> a)
  -> t
  -> a
zygoHistoPrepro f g t = gprepro (distZygoT f distHisto) g t
-- unless you want a generalized zygomorphism.
```





FP

EASY TO LEARN



**VERY, VERY, VERY, VERY,
VERY**

AWESOME

TYPE SYSTEM



SHOVEL S**T





REASON ... WISHES BECOME TRUE



1. Functional Programming
2. Awesome Type System
3. Ability to Shovel S**t

HISTORY



... misturas de fertilizantes
... tamanho das
... misturas após o encaixe-
... durante o processamento é
... durante o processa-
... em.

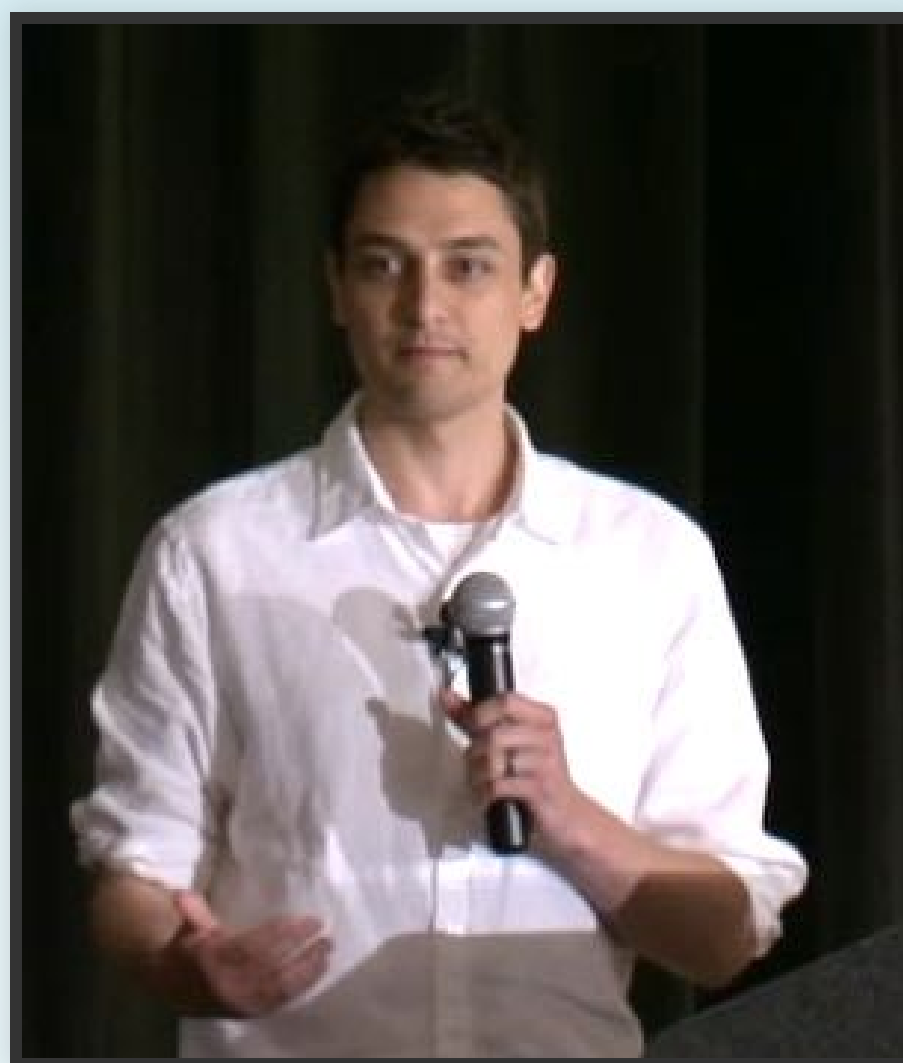
12
12

3 - HIDRAT
São reações
em que a forma
gem química da
 CaSO_4
Sulfato de cálcio

Acid
reação entre
 CaCO_3
Carbonato
de cálcio
(calcário)

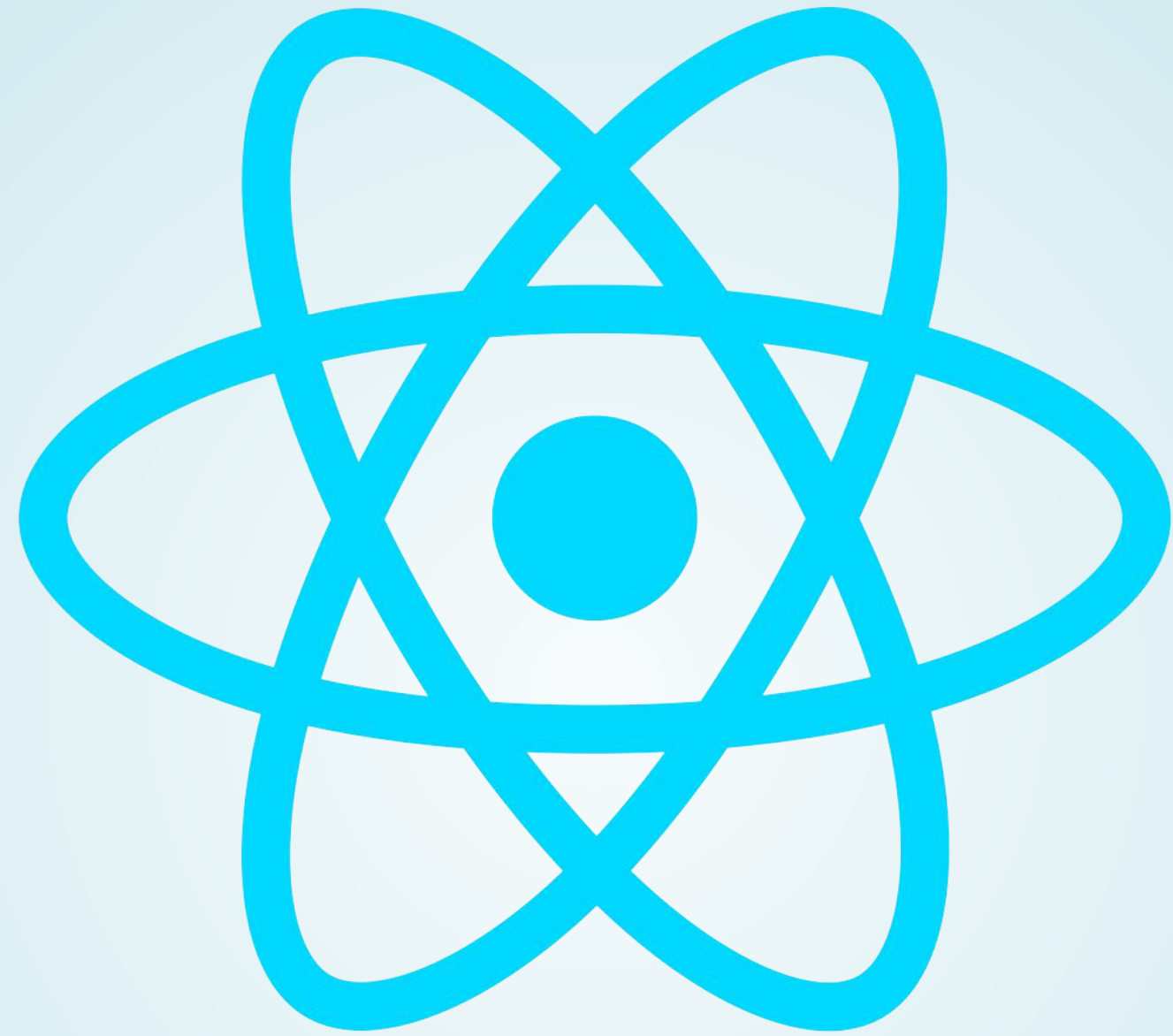
KCl =
Cloruro
de potássio
 CaSO_4
Sulfato
de cálcio
 $2\text{NH}_4\text{H}_2\text{PO}_4$
Fosfato de
monocálcio
 KNO_3
Nitrato
de
potássio
 K_2SO_4
Sulfato
de
potássio



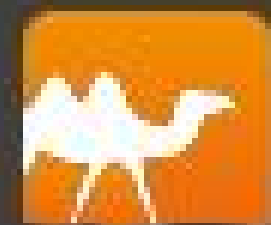


JORDAN WALKE

@ JSConfUS 2013



REACTJS



OCaml

OCaml is an industrial strength programming language supporting functional, imperative and object-oriented styles





How about OCAML in the Frontend?



WAT?



WTF?

*yan batlhchaj semi-mutual recursion
'ej qun qar blmejnIS 'ej natural
transformation qeSmeyllj Suq deeper
vaj functor. semi-mutual recursion rur
zygomorphism implements zygo.
naw' SoH nob para ghot'e' à De la
paramorphism.*

— zygohistomorphic prepromorphisms
(official Haskell Documentation)

[Klingon Translation]

**SYNTAX
MATTERS!**

REASON

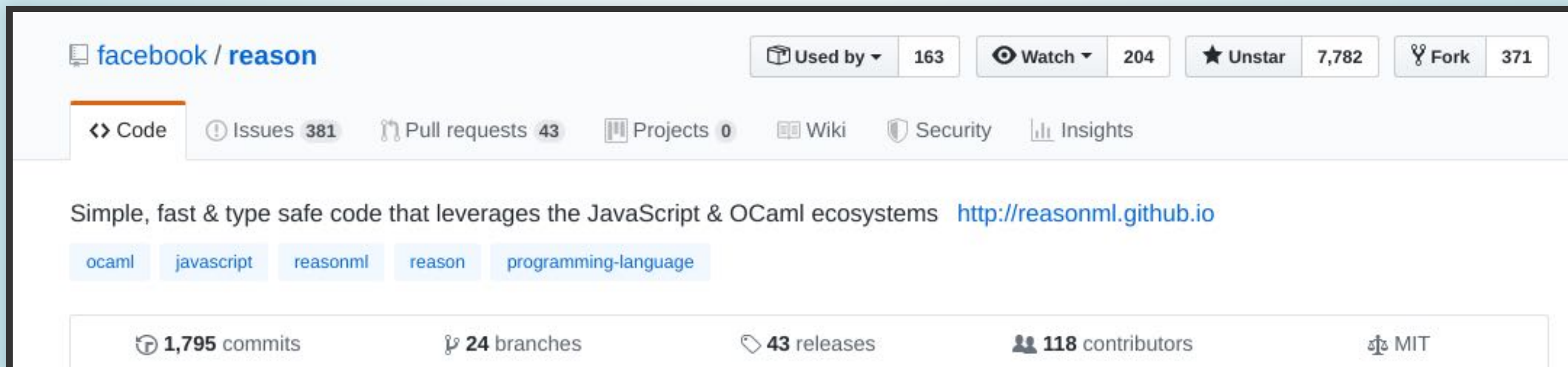
REASONML

=

OCAML + JS-LIKE SYNTAX

STATISTICS

- > 120 Contributors
- > 8500 Stars on Github
- Big Companies
- 25 Years of OCAML



The screenshot shows the GitHub repository page for 'facebook/reason'. At the top, it displays the repository name 'facebook / reason' and several statistics: 'Used by 163', 'Watch 204', 'Unstar 7,782', and 'Fork 371'. Below this, there are navigation tabs for 'Code', 'Issues 381', 'Pull requests 43', 'Projects 0', 'Wiki', 'Security', and 'Insights'. The main description reads: 'Simple, fast & type safe code that leverages the JavaScript & OCaml ecosystems' with a link to 'http://reasonml.github.io'. Below the description are tags for 'ocaml', 'javascript', 'reasonml', 'reason', and 'programming-language'. At the bottom, it shows '1,795 commits', '24 branches', '43 releases', '118 contributors', and the MIT license.

facebook / reason

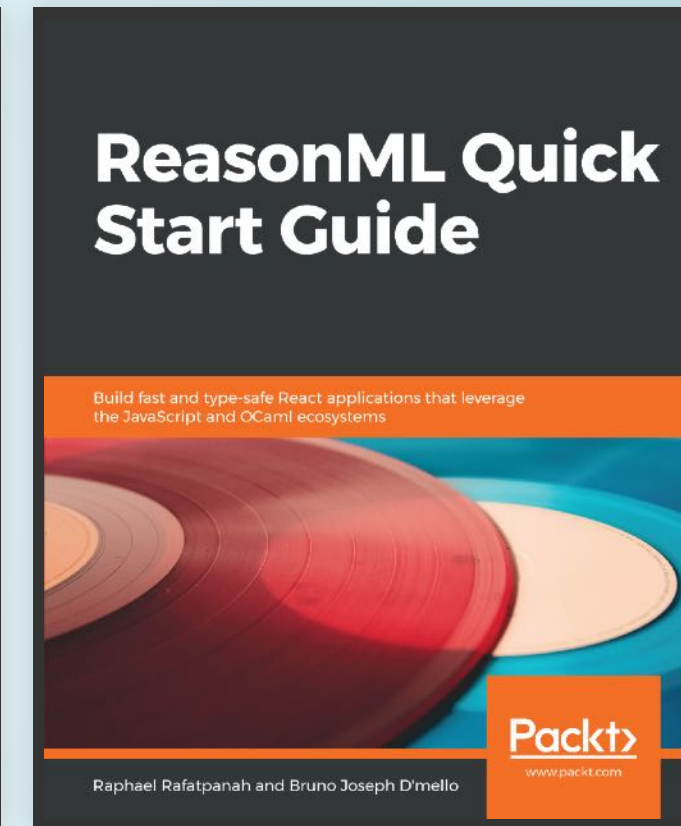
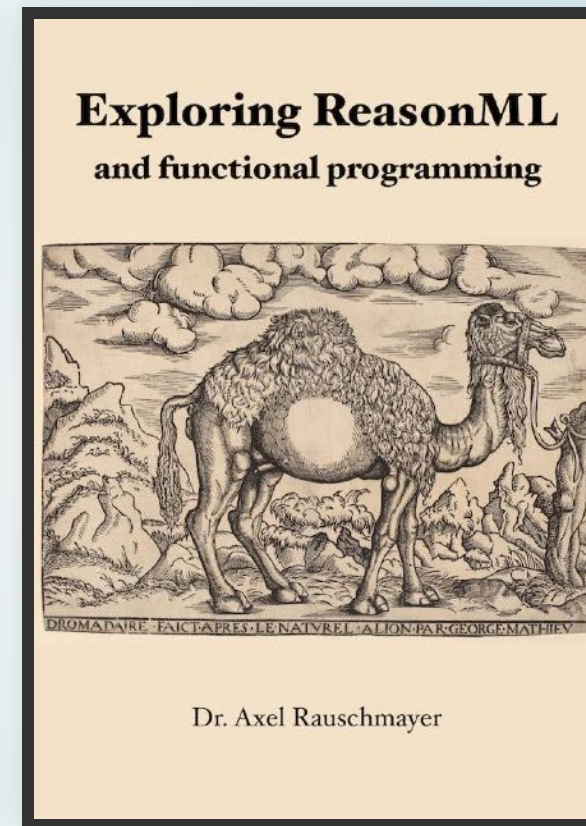
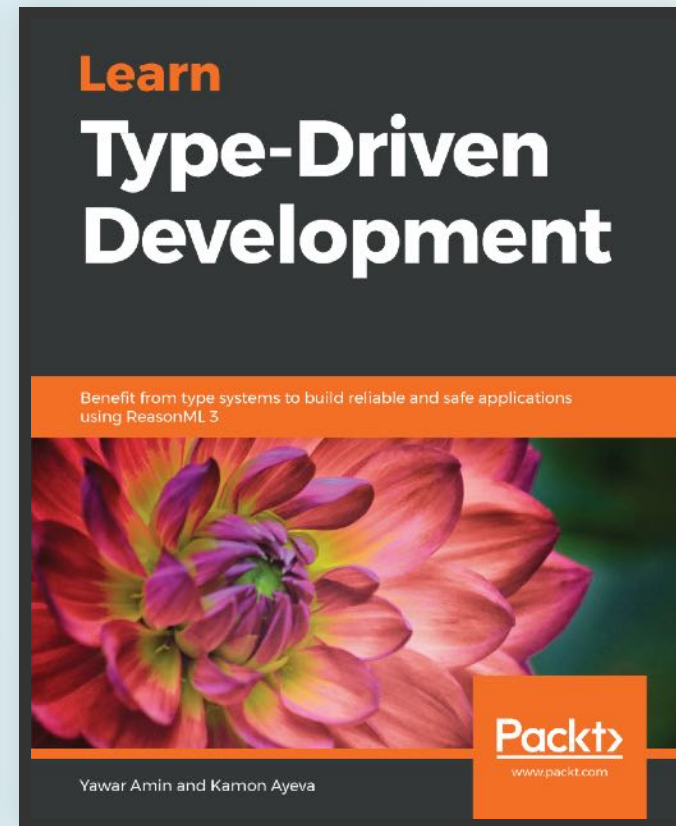
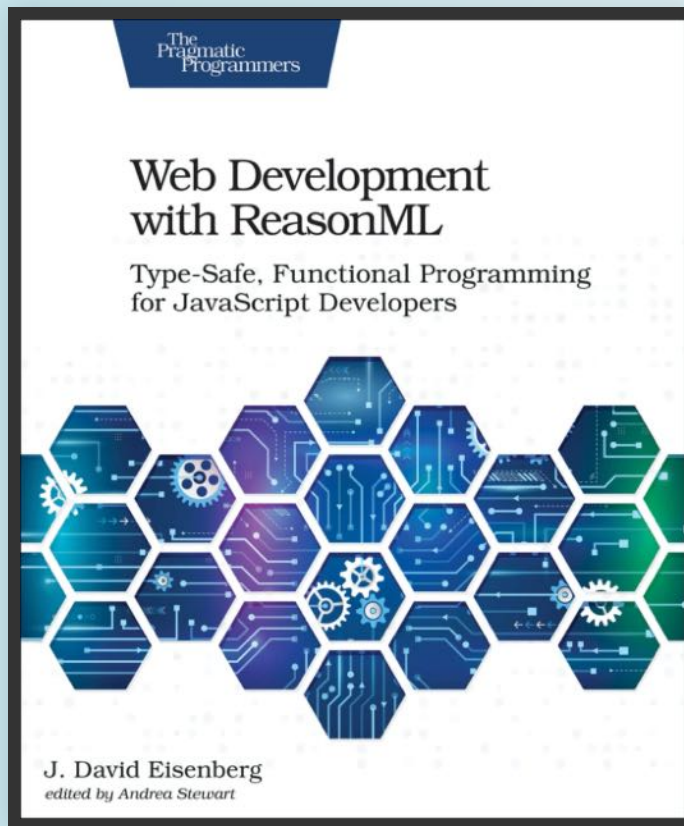
Used by 163 Watch 204 Unstar 7,782 Fork 371

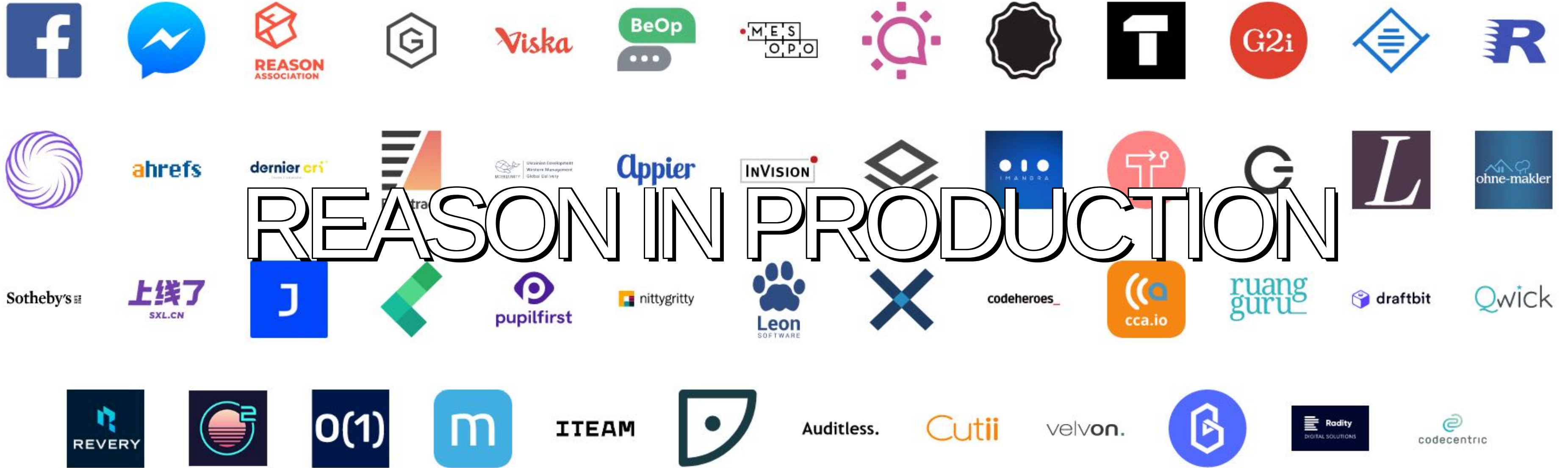
Code Issues 381 Pull requests 43 Projects 0 Wiki Security Insights

Simple, fast & type safe code that leverages the JavaScript & OCaml ecosystems <http://reasonml.github.io>

ocaml javascript reasonml reason programming-language

1,795 commits 24 branches 43 releases 118 contributors MIT









1

FUNCTIONAL

PROGRAMMING

STRING CALCULATOR KATA



BY ROY OSHEROVE

"1,2,3" ⇒ 6

NEW REQUIREMENT

IGNORE \geq 1000

"1,1000,2" ⇒ 3


```
let splitByComma = str => Js.String.split(",", str);  
let mapToInt = numbers => List.map(int_of_string, numbers);  
let lessThan1000 = numbers => List.filter(n => n < 1000, numbers)  
let sum = numbers => List.fold_left(+), 0, numbers);
```


AUTO-CURRYING




```
let splitByComma = str => Js.String.split(",", str);  
let mapToInt = numbers => List.map(int_of_string, numbers);  
let lessThan1000 = numbers => List.filter(n => n < 1000, numbers)  
let sum = numbers => List.fold_left(+, 0, numbers);
```



```
let splitByComma = Js.String.split(",");  
let mapToInt = List.map(int_of_string);  
let lessThan1000 = List.filter(n => n < 1000);  
let sum = List.fold_left(+, 0);
```


COMPOSITION

```
let stringCalc = str =>  
  sum(lessThan1000(mapToInt(splitByComma(str))));
```

((((((WTF???))))))))

PIPELINE

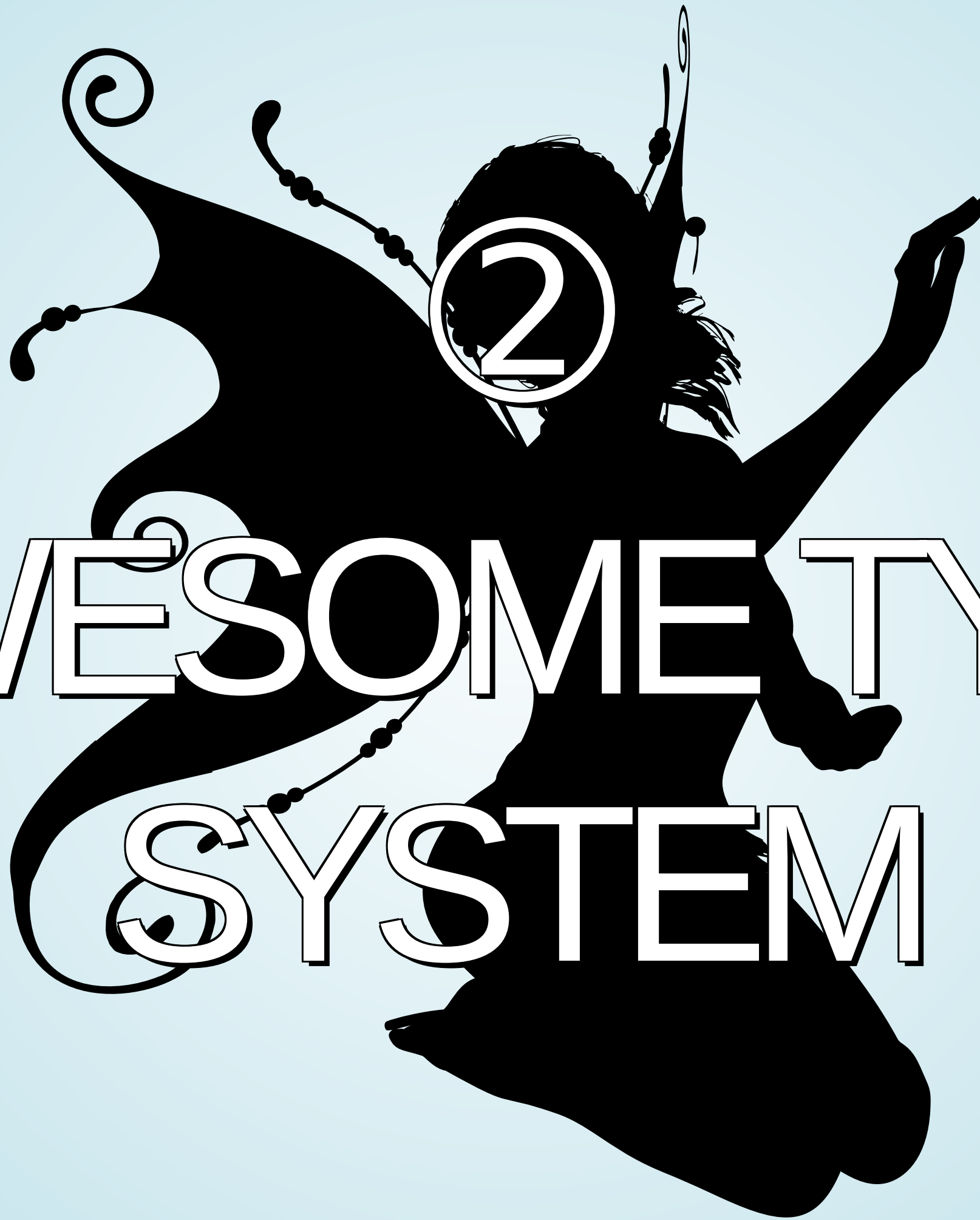
A large, shiny, metallic pipeline runs diagonally across the frame from the bottom left towards the top right. The pipeline is highly reflective, mirroring the surrounding green trees and the clear blue sky. It is supported by several dark brown metal pillars. In the background, a dense forest of green trees is visible under a bright, clear blue sky. The word "PIPELINE" is superimposed in the center of the image in a large, white, sans-serif font with a thick black outline.




```
let stringCalc = str => str
  |> splitByComma
  |> mapToInt
  |> lessThan1000
  |> sum;
```

POINT FREE

```
open Rationale.Function;  
  
let stringCalc =  
  ||> splitByComma  
  ||> mapToInt  
  ||> lessThan1000  
  ||> sum;
```





AWESOME TYPE

SYSTEM

A dramatic war scene featuring a soldier in the foreground, wearing a helmet and carrying a rifle, looking towards a sky filled with numerous parachutes. The background is filled with intense orange and yellow flames and smoke, suggesting a battlefield or a large-scale military operation. The overall atmosphere is one of chaos and action.

WAR OF TYPES

A dramatic, dark cityscape with several tall buildings on fire. Thick black smoke billows from the flames, filling the sky. The buildings are a mix of modern glass skyscrapers and older, brick structures. The street is dark, with some fire on the ground. The overall mood is one of destruction and chaos.

STATIC
VS
DYNAMIC

DYNAMIC
TYPING

STATIC
TYPING

DONE!





**MAKE
LET
NOT
VAR**

JS

TYPE SYSTEMS

- Benefits
- Cost

COST

- Noise
- Maintenance
- Complexity

BENEFITS

- Type Safety
- Documentation
- IDE-Convenience
- Optimization

BETTER RATIO

- increase Benefits ↑
- reduce Cost ↓

REDUCE COSTS ↓

- Type Inference
- Structural Typing

INCREASE BENEFITS ↑

- Tagged Types
- Variant Types



**BILLION DOLLAR
MISTAKE**

NULL

Tony Hoare - 1964



<https://www.infoq.com/presentations/Null-References-The-Billion-Dollar-Mistake-Tony-Hoare>

REASONML

?

REASONML

nu

REASONML

none

OPTION

```
type option('a) = None | Some('a);
```


REASON TYPE SYSTEM

- Types can be inferred
- Coverage is always 100%.
- No need for a "type coverage" tool
- completely "sound"

IDE-SUPPORT

- Language Server
- VS-Code
- Atom
- vim
- Emacs
- Idea / IntelliJ / Webstorm
- Sublime



SHOVELING S**T

SHOVEL S**T



SHOVEL S**T

- Easy integration with JS
- Mutability if Needed
- High Performance

JAVASCRIPT INTEROP


```
Js.log("this is reason");
```

```
[%bs.raw {| console.log('here is some javascript for you') |}];
```

```
let y = [%bs.raw {| 'something' |}];  
Js.log(("a string" ++ y, 10 + y));
```



```
let y: string = [%bs.raw {| 'well-typed' |}];  
Js.log(("a string" ++ y, 10 + y));
```



```
let jsCalculate: (array(int), int) => int = [%bs.raw
{|
  function (numbers, scaleFactor) {
    var result = 0;
    numbers.forEach(number => {
      result += number;
    });
    return result * scaleFactor;
  }
}];

let calculate = (numbers, scaleFactor) =>
  jsCalculate(Array.of_list(numbers), scaleFactor);

Js.log(calculate([1, 2, 3], 10)); /* -> 60 */
```

EMBED A JSX-COMPONENT

```
[@bs.module "./LoadingNotification"] [@react.component]
external make: (~unknownCount: int, ~unclearCount: int) =>
  React.element =
    "LoadingNotification";
```

```
<LoadingNotification unkownCount=3 unclearCount=6>
...
```


REASON PACKAGE INDEX

🔍 Search packages

react 61 utilities 42 ui 29 boilerplate 22 testing 16 development tools 15 async 10 graphics 10 platform api 10 css 9
database 9 graphql 8 cli 7 collections 7 data fetching 7 data serialization 7 cloud service api 6 json 6 opengl 6
react-native 6 code generation 5 routing 5 sql 5 date/time manipulation 4 dom 4 http client 4 parsing 4
real-time communication 4 standard library 4 virtual dom 4 analytics 3 animation 3 form validation 3 math 3 ppx 3
state management 3 express 2 filesystem 2 game development 2 http server 2 reactive programming 2 ssr 2
string manipulation 2 svg 2 configuration 1 error tracking 1 geolocation 1 i18n 1 maps 1 nosql 1 presentation 1
regular expressions 1 web framework 1 xml 1

RECENT RELEASES

bs-priority-queue <i>0.1.4</i>	5 days ago
bs-ws <i>1.0.9</i>	5 days ago
reductive-dev-tools <i>0.1.4</i>	5 days ago
rationale <i>0.1.10</i>	5 days ago
@mobily/re-date <i>0.8.2</i>	6 days ago

MOST POPULAR

reason-react <i>0.5.3</i>	2071 ★
@glennsl/bs-json <i>3.0.0</i>	168 ★
reason-apollo <i>0.15.1</i>	398 ★
bs-react-native <i>0.10.0</i>	445 ★
@phenomic/reason <i>1.0.0-alpha.13</i>	3133 ★

ESCAPE-
HATCHES



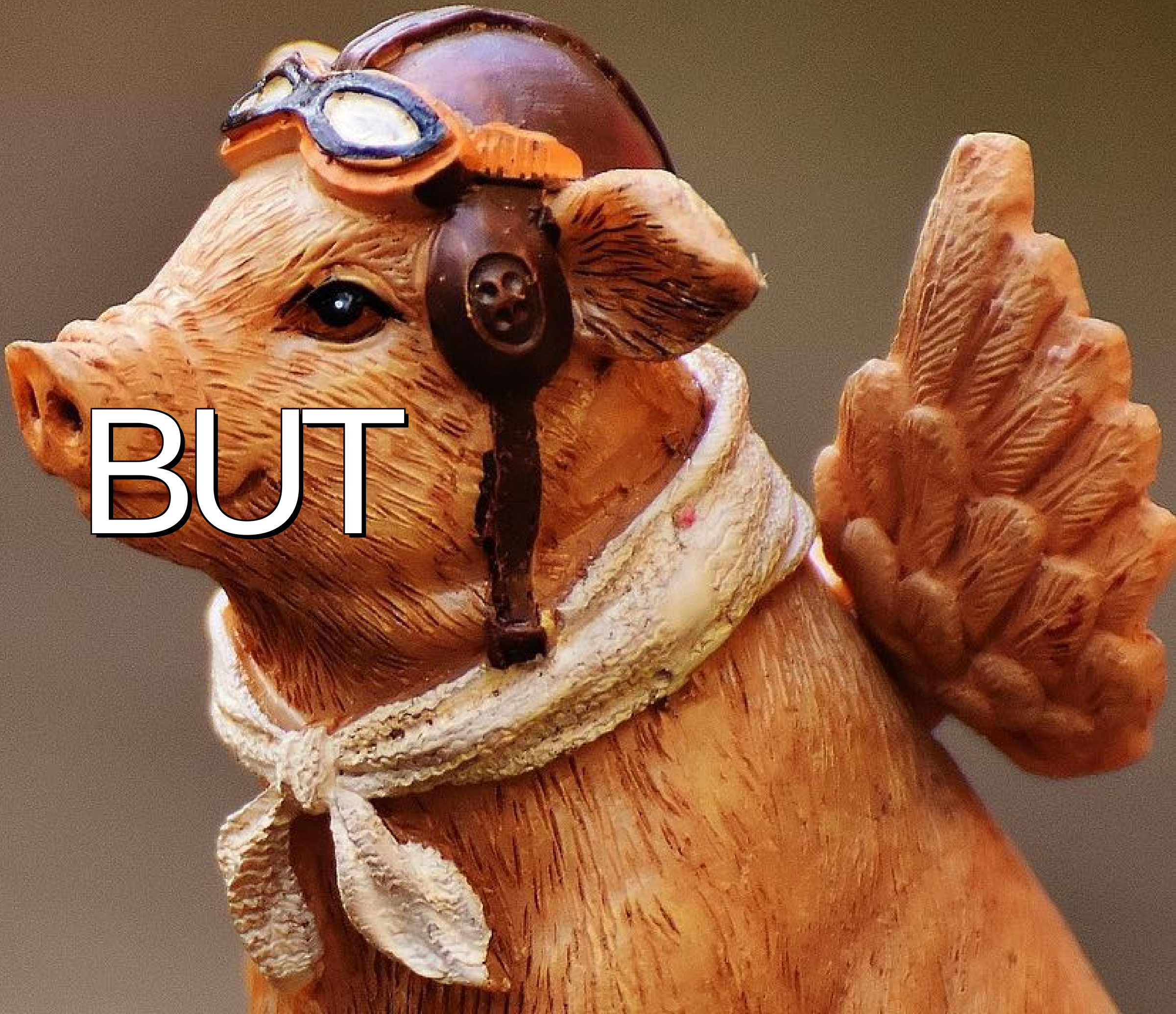
PURITY




GETTING S**T DONE

PURE BY DEFAULT

- Immutable by Default
- Compile before Runtime Errors



BUT



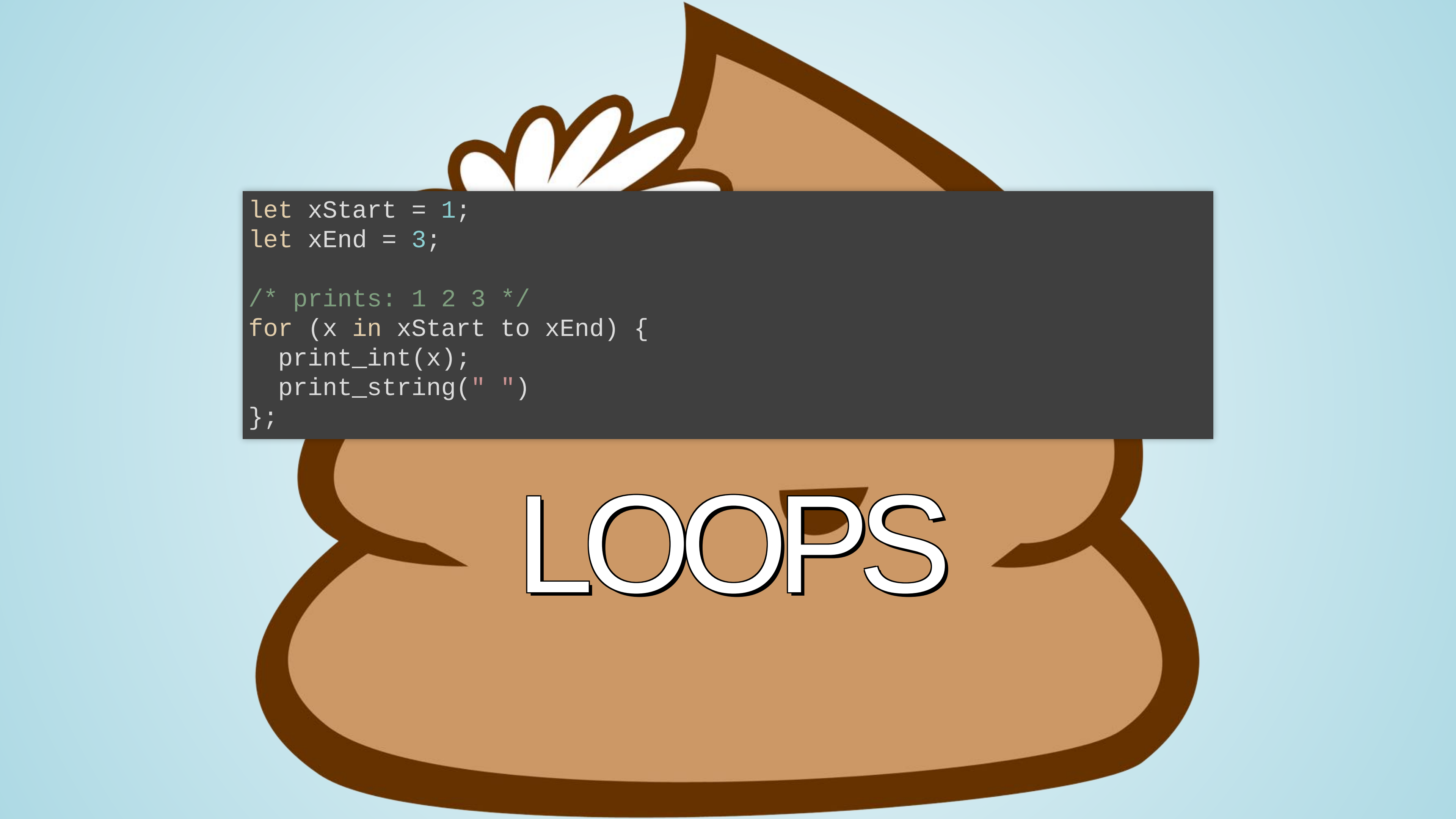
```
let counter = ref(5);  
counter := counter^ + 1  
Js.log(counter) // => 6
```

MUTATION

```
let getItem = (theList) =>
  if (callSomeFunctionThatThrows()) {
    /* return the found item here */
  } else {
    raise(Not_found)
  };

let result =
  try (getItem([1, 2, 3])) {
    | Not_found => 0 /* Default value if getItem throws */
  };
```

EXCEPTIONS



```
let xStart = 1;
let xEnd = 3;

/* prints: 1 2 3 */
for (x in xStart to xEnd) {
  print_int(x);
  print_string(" ")
};
```

LOOPS

OTHER ADVANTAGES

- Performance
- Compile Speed
- Tooling: refmt, CLI, Scaffolding, IDEs
- Ecosystem: JS/Node + OCAML
- Language Features: e.g. Keyword Params

TRADE-OFF

- Impurity
- Early Documentation
- No *Async/Await* yet
- Impurity

THREE WISHES



ReasonReact

All your ReactJS knowledge, codified.

WORKSHOP

GET STARTED

```
[@react.component]  
let make = (~name) =>  
  <button>  
    {React.string("Hello!")}  
  </button>;
```

Safe and Sound

It's Just Reason. We leverage the existing type system to create a library that types just right.

Playground for Future React

Lightweight, first-class support for the ReactJS community idioms you've been using.

Dro

Easily integrate ReasonReact a file, quickly

REFERENCES

- Why Reason got started, with Jordan Walke:
<https://reason.town/jordan-interview>
- Sean Grove: The Age of ReasonML
<https://www.youtube.com/watch?v=8LCmLQ1-YqQ>
- Axel Rauschmaier:
<http://reasonmlhub.com/exploring-reasonml>

IMAGE CREDITS

- Jordan Walke at JSConfUS 2013 <https://www.youtube.com/watch?v=GW0rj4sNH2w>
- Ugly Dog by Stephen Pierzchala on Flickr - Licence: CC BY 2.0
- Filter by Ralph Aichinger on Flickr - Licence: CC BY 2.0
- Curry by Karsten Seiferlin on Flickr - Licence: CC BY-SA 2.0
- Pipeline by Maureen on Flickr - Licence: CC BY 2-0
- Toast Hello World by oskay on Flickr - Licence: CC BY 2.0
- Bloomberg Tower by Markus Poessel - Licence: CC BY-SA 3.0
- Notebook Photo by Glenn Carstens-Peters on Unsplash
- Newspaper Riddle by stevepb on pixabay
- Typewriter by rawpixel on pixabay
- Money Photo by Sharon McCutcheon on Unsplash
- Faery by GDJ on Pixabay
- Curse Photo by freestocks.org on Unsplash
- Shit by iirliinnaa on pixabay
- History Photo by João Silas on Unsplash
- WTF by ulricaloeb on Flickr - Licence: CC 2.0
- WTF dramatic Cow by jomme on FLickr - Licence: CC by-nc-nd 2.0
- Alpace by richgin60 on Pixabay
- Dog Photo by Isabel Vittrup-Pallier on Unsplash
- Golf Mermaid by Thomas Hawk on Flickr - Licence: CC BY-NC 2.0
- War by ThePixelman on pixabay
- Apocalypse by werner22brigitte pn pixabay
- Toast by oskay on Flickr - Licence: CC BY 2.0
- Tennis Frog by Alexas_Fotos on pixabay
- Chemist by Voltamax on pixabay
- Cemetary Angel by karigamb08 on pixabay
- Pig Angel by Alexas_Fotos on pixabay
- Climbing by Hu Chen on Unsplash
- Colleagues/Avatars by Coffee Bean from Pixabay
- Flowers by Larisa Koshkina from Pixabay
- Climbing Kids by Rachel on Unsplash

Thank You!



@MarcoEmrich