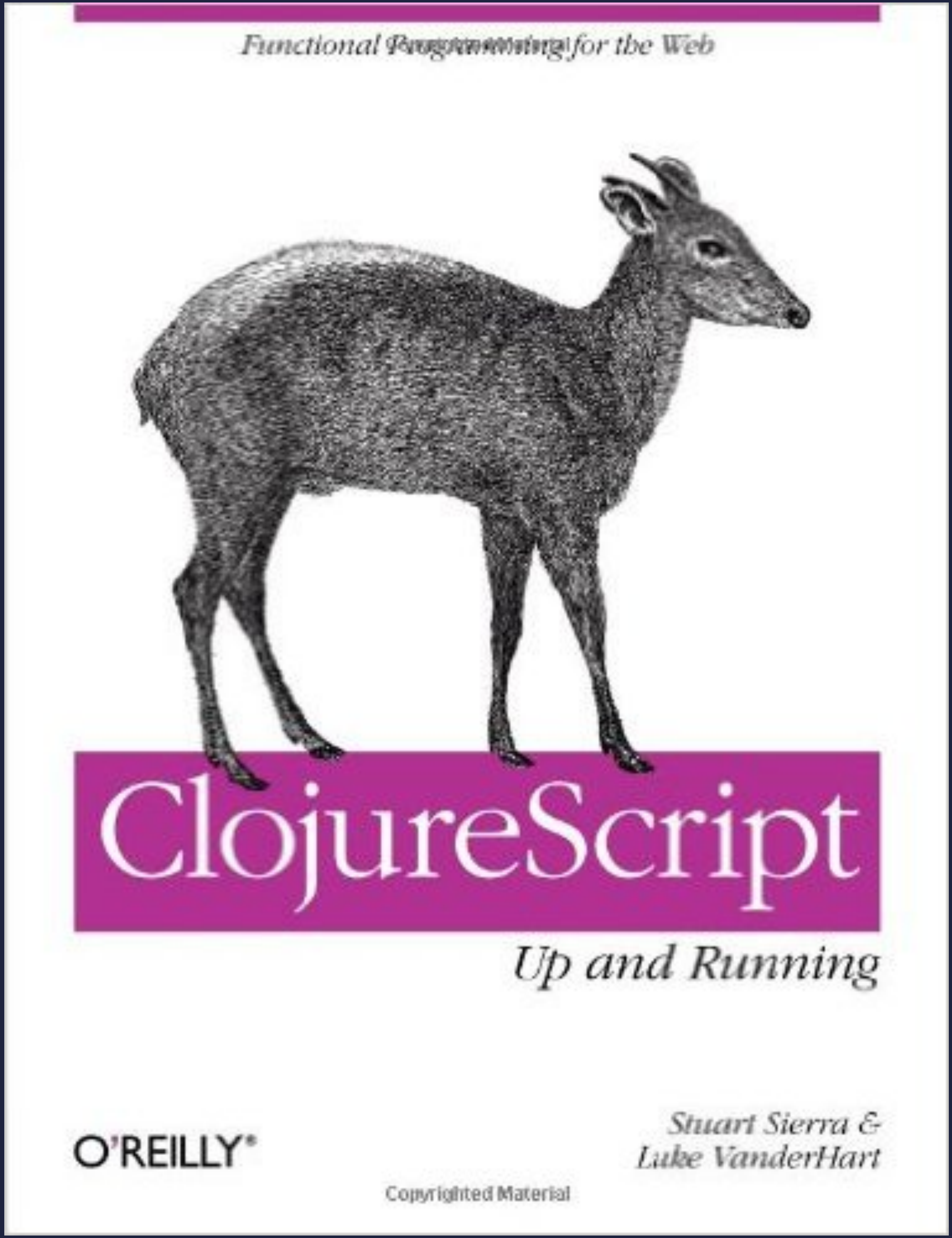
Land of ClojureScript

What do you get when you merge a rhino and an elephant?

UNSCRAMBLE

PYTE

> Programming is not about typing, it's about thinking.
>
> — Rich Hickey



Functional Programming for the Web

ClojureScript

Up and Running

Stuart Sierra &
Luke VanderHart

O'REILLY®

GITHUB STARS

ClojureScript: 5,112
Clojure: 4,591

DEMO

Hello World

NEW SCHOOL JAVASCRIPT

WHAT IS CLOJURESCRIPT?

Clojure is a compiler for Clojure that targets JavaScript.

WHY CLOJURESCRIPT?

WELL DESIGNED

JAVASCRIPT THE GOOD PARTS

CLOJURESCRIPT THE AWESOME PARTS

NAMESPACES

```
(ns awesome.core)
```

MACROS

```
(defmacro unless
  [pred & body]
  `(if (not ~pred)
    (do ~@body) nil))

;; Compiles to
(macroexpand '(unless true (/ 1 0)))
; => (if (clojure.core/not true) (do (/ 1 0)) nil)')`)
```

GREAT LIBRARIES

CLOJURESCRIPT BASICS

FUNCTION CALLS

```
(+ 1 (* 2 3))
;; 7
```

NAMED FUNCTION

```
(defn square
  [x]
  (* x x))
```

NAMED FUNCTION

```
function square(x) {
  return x * x
}
```

ANNONYMOUS FUNCTION

```
(fn [x] (* x x))
```

ANNONYMOUS FUNCTION

```
function(x) { return x * x }
```

JAVASCRIPT INTEROP

```
;; method call
(. js/document (getElementById "app"))

;; get property
(.-value input)
```

LET

```
(let [x 1]
  (println x))
```

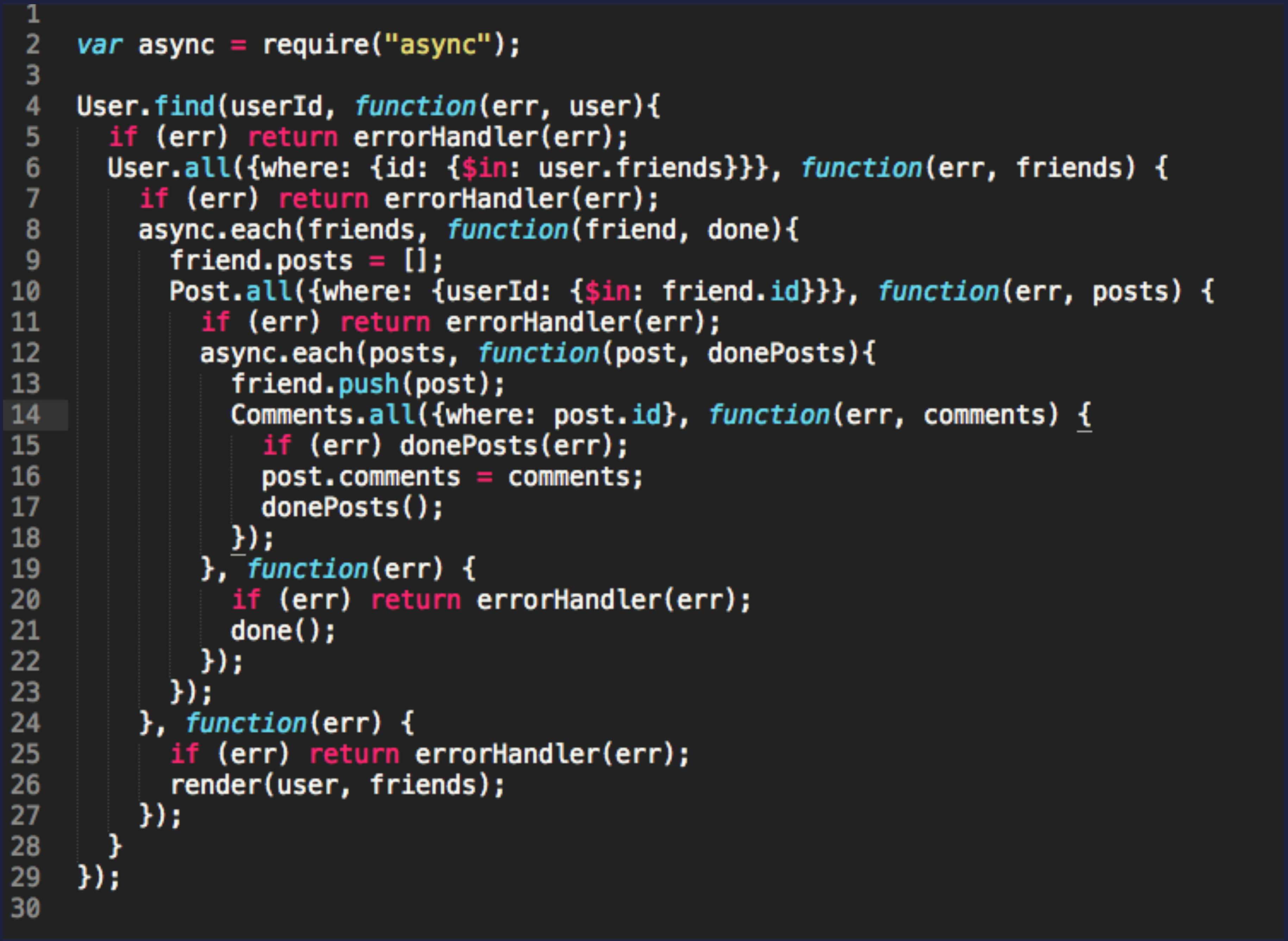
ATOM

```
(atom {})
```

ASYNCHRONOUS JAVASCRIPT

```
$('.submit').click(function(e){
  // add class to form
  // submit form
  // disable form so user can't keep clicking
});
```

ASYNCHRONOUS JAVASCRIPT



```
var async = require("async");

User.find(userId, function(err, user){
  if (err) return errorHandler(err);
  User.all({where: {id: {$in: user.friends}}}, function(err, friends) {
    if (err) return errorHandler(err);
    async.each(friends, function(friend, done){
      friend.posts = [];
      Post.all({where: {userId: {$in: friend.id}}}, function(err, posts) {
        if (err) return errorHandler(err);
        async.each(posts, function(post, donePosts){
          friend.push(post);
          Comments.all({where: post.id}, function(err, comments) {
            if (err) donePosts(err);
            post.comments = comments;
            donePosts();
          });
        }, function(err) {
          if (err) return errorHandler(err);
          done();
        });
      });
    }, function(err) {
      if (err) return errorHandler(err);
      render(user, friends);
    });
  }
});
```

CORE.ASYNC

Clojure's implementation of Communicating Sequential Processes

CHAN

```
(chan)
```

PUT

```
(put! "value")
(>! "value")
```

TAKE

```
(take! "value")
(<! "value")
```

GO

```
(go)
```

WRITE SEQUENTIAL LOGIC

```
(go
  (println "Waiting...")
  (<! clicks-chan)
  (println "Got something")
)
```

DEMO

STATE & MUTATION

EQUALITY IN JAVASCRIPT

```
var bobFromYesterday = {name: 'bob'};
var bobFromThisMorning = {name: 'bob'};

bobFromYesterday == bobFromThisMorning // false
```

EQUALITY IN CLOJURESCRIPT

```clojure
(def bob-from-this-morning {:name 'bob'})
(def bob-from-yesterday {:name 'bob'})

(= bob-from-this-morning bob-from-yesterday) ;; true
```

CHANGING STATE IN JAVASCRIPT

```
var ages = [24 25 26]

process(ages) ;; function written by co-worker

console.log(ages)
// maybe [24 25 26]
```

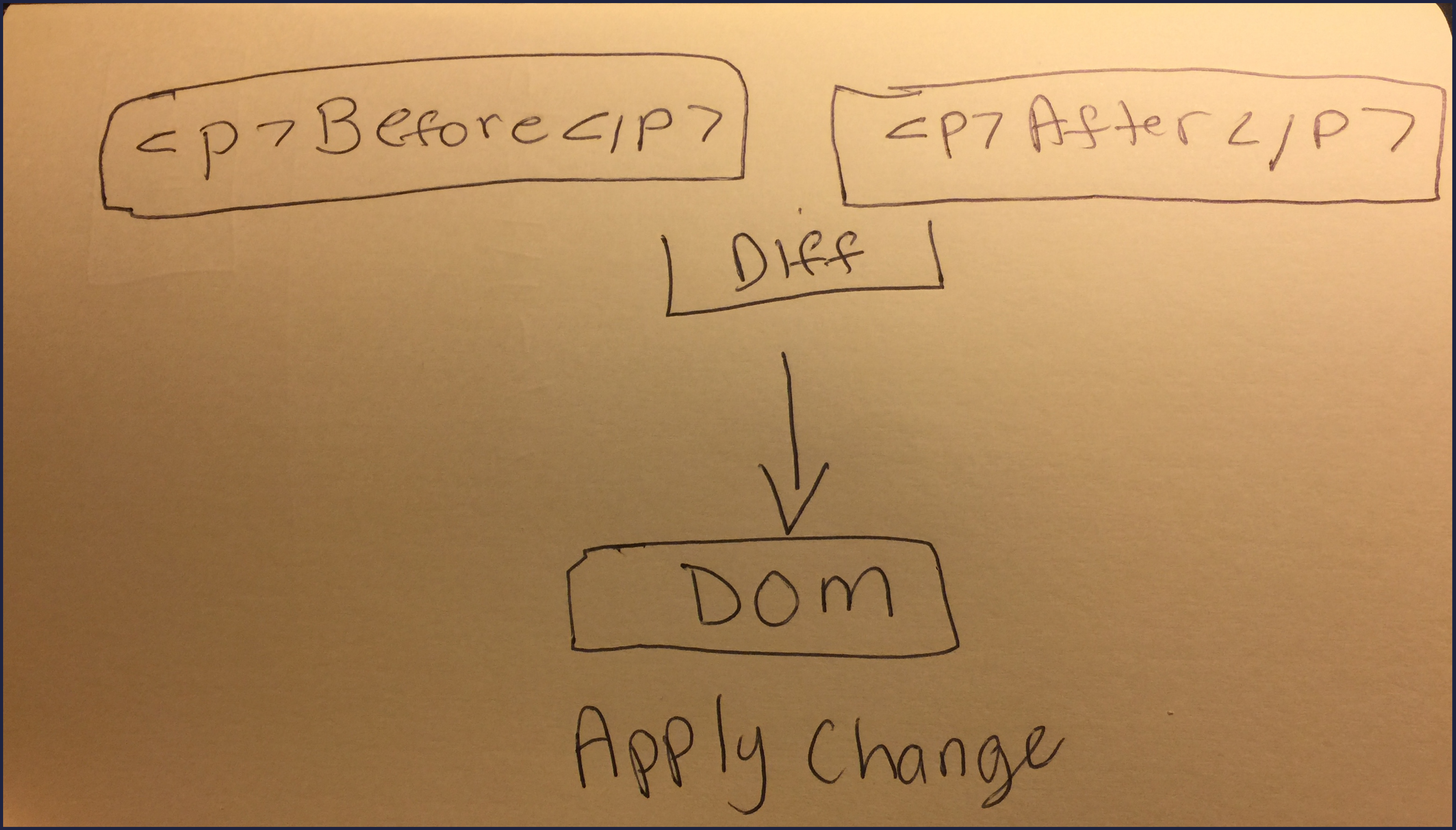
IMMUTABILITY IN CLOJURESCRIPT

```
(def ages [24 25 26])

(process ages) ;; function written by co-worker

(println ages)
;; [24 25 26]
```

REACT

A JavaScript library for building user interfaces

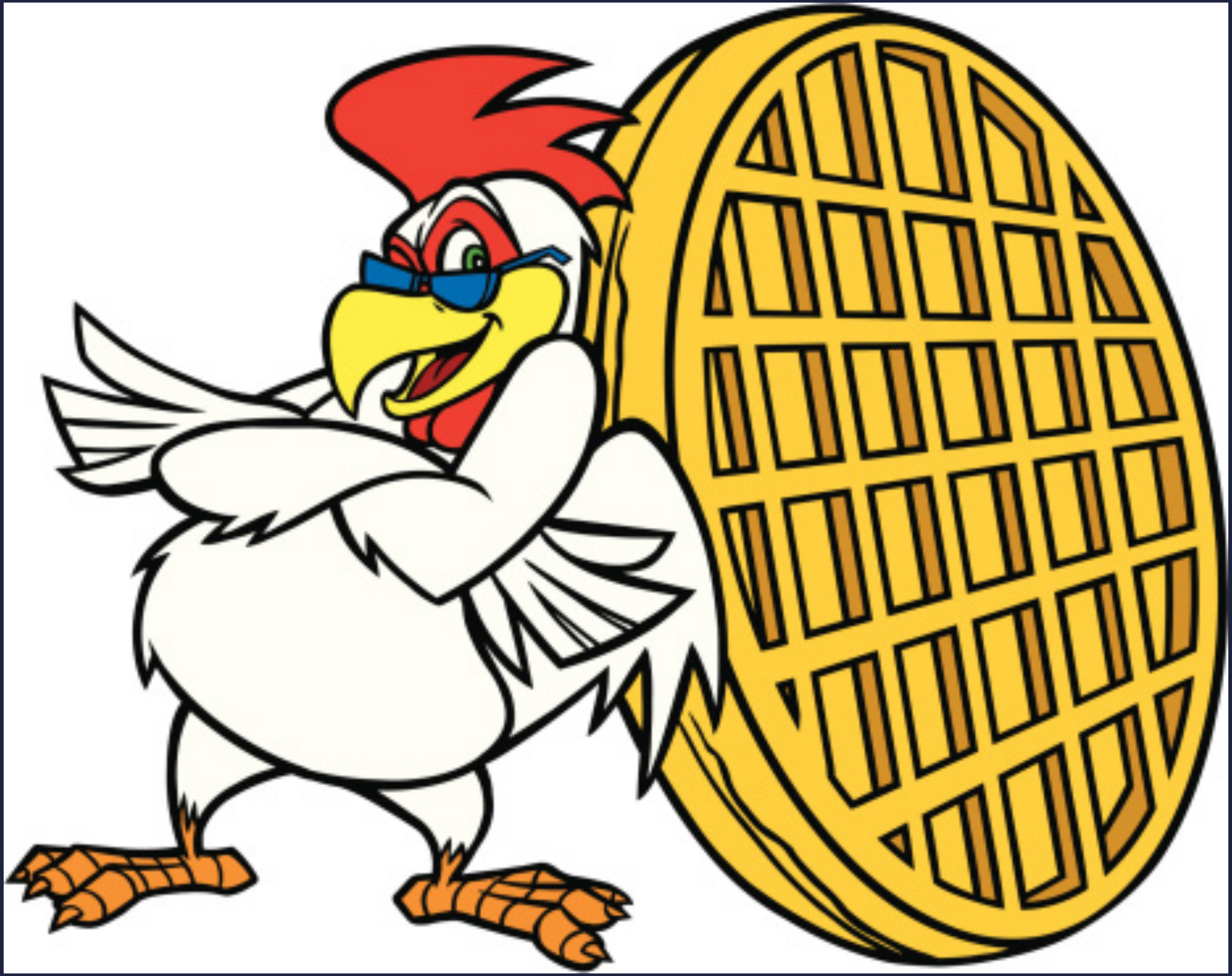
REACT



OM

A ClojureScript interface to Facebook's React.

OM + REACT



DEMO

TRANSDUCERS

Composable transformations

```clojure
(defn is-admin?
[record]
  (= "admin" (record :role)))
```

```
(defn full-name
  [record]
  (let [first-name (record :first-name)
        last-name (record :last-name)]
        (str first-name last-name)))
```

```
(defn admin-full-names
  (comp
    (filter is-admin?)
    (map full-name)))
```

COMMAND LINE TOOLS

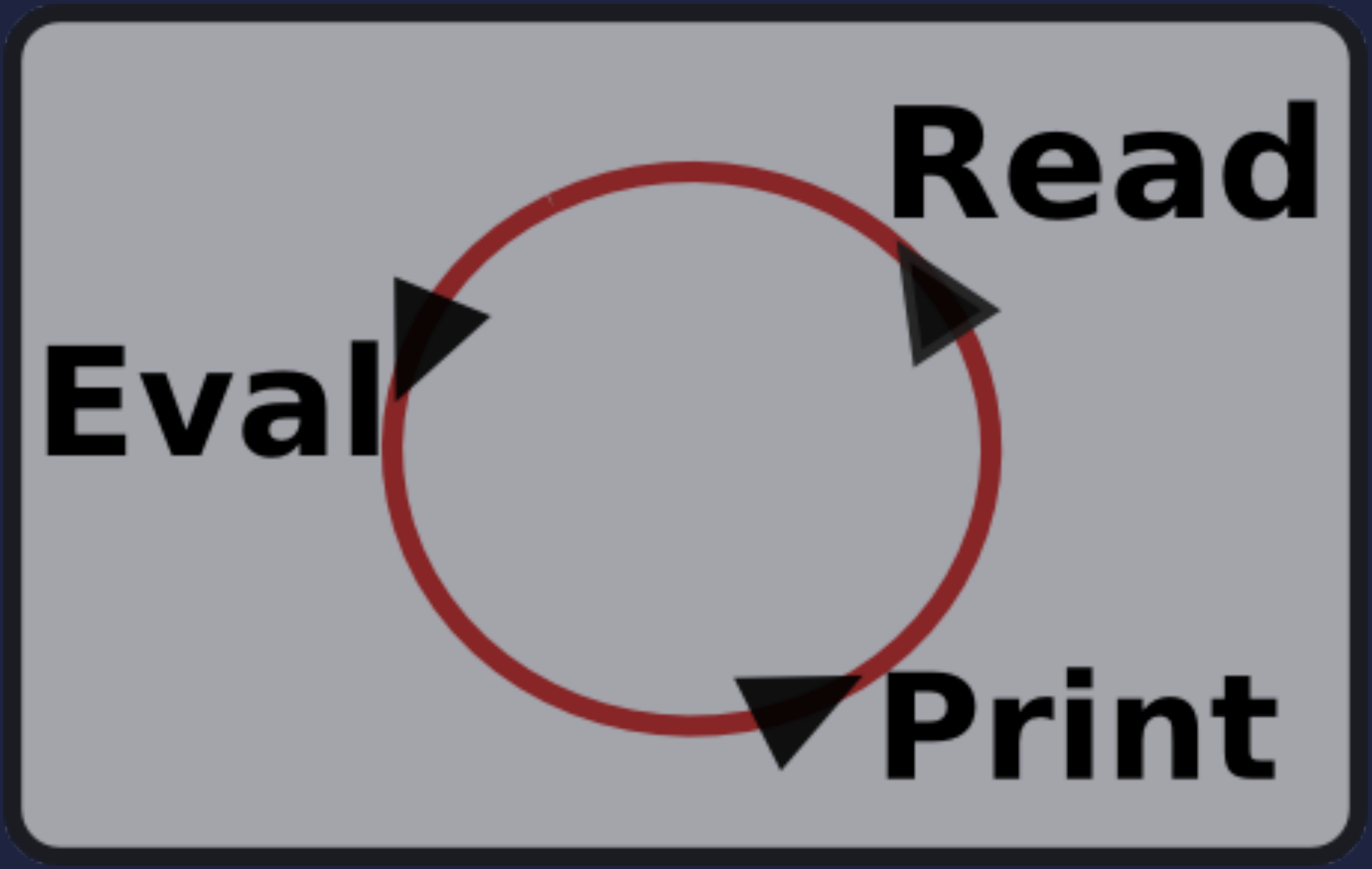
Demo

DEBUGGING

SOURCE MAPS

FIGWHEEL

BROWSER REPL



INTEGRATING WITH JAVASCRIPT

EXTERNS

```
var User = {};
User.fullName = function() {};
js/User
```

CLJSJS

CLJSJS: http://cljsjs.github.io/

SUMMARY

It's ok not to use JavaScript

HELPFUL LINKS

- http://clojurescript.net/
- http://funcool.github.io/clojurescript-unraveled/
- https://github.com/circleci/frontend
- https://www.youtube.com/user/ClojureTV
- http://swannodette.github.io/
- https://github.com/iamjarvo/pplz

LETS PAIR PROGRAM :)

- Jearvon Dharrie
- Twitter: @jearvon
- Podcast: http://turing.cool