

2021 VS CLOJURESCRIPT. STILL ALIVE?

Alex Bykbaiev

ABOUT



Alex Bykbaiev, Frontend Engineer at Hopin.

AGENDA

- ClojureScript, what is it??
- Staff I like in ClojureScript
- There is always a small fly in the ointment
- How and why we use ClojureScript in production
- Some resources to learn more about ClojureScript
- Q&A

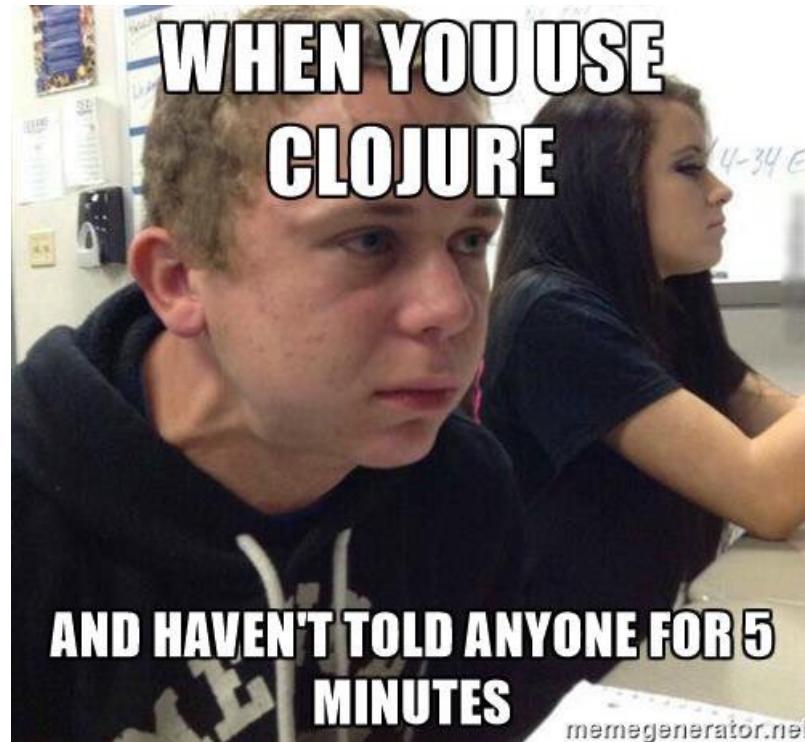
CLOJURESCRIPT, WHAT IS IT??

Clojure is a compiled, functional and dynamically typed language that targets multiple existing runtimes.

ClojureScript is a compiler for Clojure that targets JavaScript.



STAFF I LIKE IN CLOJURESCRIPT



MANAGEMENT OF CONTROL FLOW



REPL (READ-EVAL-PRINT LOOP)

The screenshot shows a code editor window with a dark theme. On the left, there is a code editor pane titled "user" containing the following Clojure code:

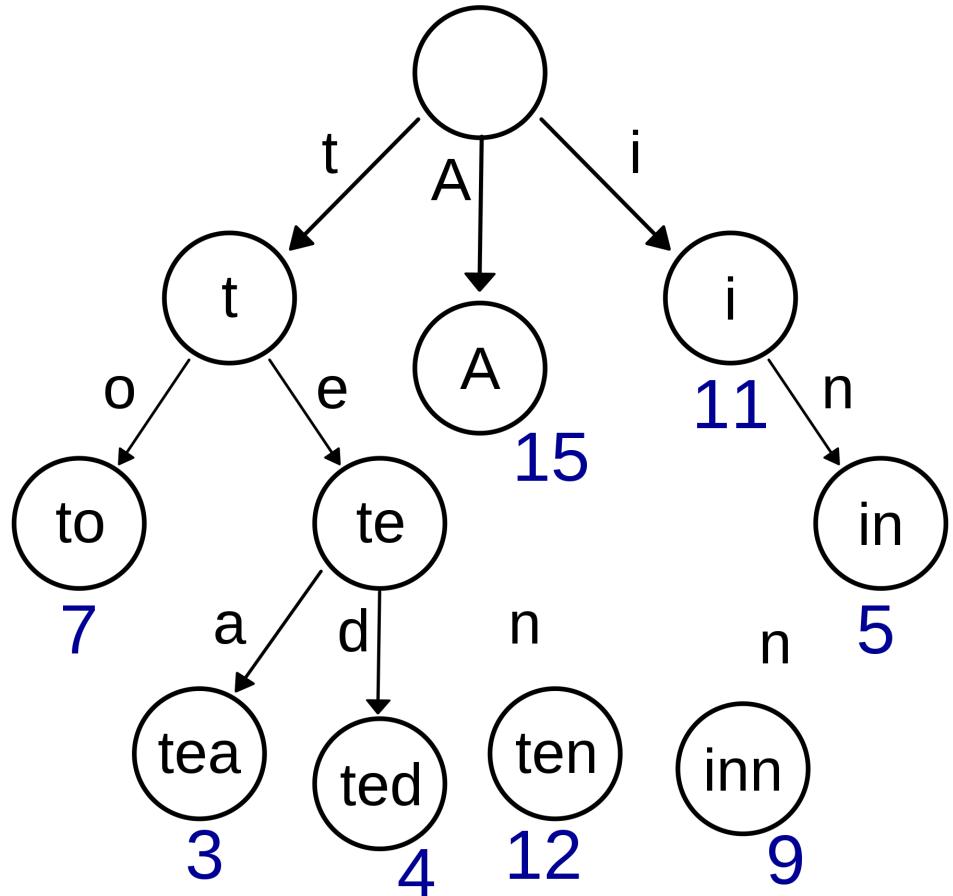
```
(ns user)
(defn square [x]
  (* x x))
```

To the right of the code editor is a "REPL Local: user" panel. This panel has a toolbar with various icons at the top, followed by a large, empty text area where the results of the evaluation would appear. At the bottom of the editor window, there is a status bar with the following information:

Compilation completed successfully in 2s 267ms (4 minutes ago) 5:1 LF: UTF-8: Git: master: Paredit: 1

COLLECTIONS

All collections in ClojureScript are persistent (immutability + structural sharing).



LIST

List provides efficient access to the first elements
(linked list under the hood).

```
1 (def list' '(1 2 3 4))
2
3 (list' 0)
4 => 1
5
6 (let [xs '(1 2 3)           ;;
      ys (cons 0 xs)] ;;
     (identical? xs (rest ys)))
7
8 => true
```

VECTOR

Vector provides efficient index access to its elements.

```
1 (def vec' [1 2 3 4])  
2  
3 (vec' 0)  
4 => 1
```

MAP

Map is a collection abstraction that allow you to store key/value pairs.

```
1 (def map' { :a 1 :b 2} )  
2  
3 (get map' :a)  
4 => 1  
5  
6 (map' :a)  
7 => 1  
8  
9 (:a map')  
10 => 1
```

SET

Set stores zero or more unique items of any type and is unordered.

```
1 (def set' #{1 2 3 10})  
2  
3 (set' 0)  
4 => nil  
5  
6 (set' 10)  
7 => 10
```

METAPROGRAMMING

Clojure is also a dialect of LISP (LISt Processing language) by design. The aim was to make the language more extensible.

Clojurescript allows you to write code that writes code (macros).

MACROS

```
1 (defmacro infix
2   [infixed]
3   (list (second infixed) (first infixed) (last infixed)))
4
5 (+ 1 1)
6 => 2
7
8 (infix (1 + 1))
9 => 2
```

THREADING MACROS

Thread-first macro

```
1 (def user {:first-name "Alex"
2                 :last-name   "Bykbaiev"} )
3 => {:first-name "Alex"
4       :last-name "Bykbaiev"}
5
6 (-> user
7       (update :full-name
8                 #(str (:first-name %)
9                         " "
10                            (:last-name %)))
11       (assoc :mental-age 54)))
12 => {:first-name "Alex"
13       :last-name   "Bykbaiev"
14       :full-name   "Alex Bykbaiev"
15       :mental-age 54}
```

THREADING MACROS

Thread-first macro

```
1  (der user {:_first-name "Alex"
2                  :last-name "Bykbaiev" } )
3 => {:_first-name "Alex"
4      :last-name "Bykbaiev" }
5
6  (-> user
7      (update :full-name
8                  #(str (:first-name %)
9                      " "
10                     (:last-name %)) )
11      (assoc :mental-age 54) )
12 => {:_first-name "Alex"
13      :last-name "Bykbaiev"
14      :full-name "Alex Bykbaiev"
15      :mental-age 54}
16
```

THREADING MACROS

Thread-first macro

```
9          " "
10         (:last-name %) )))"
11   (assoc :mental-age 54)))
12 => {:first-name "Alex"
13     :last-name "Bykbaiev"
14     :full-name "Alex Bykbaiev"
15     :mental-age 54}
16
17 (assoc (update user
18               :full-name
19               #(str (:first-name %)
20                     " "
21                     (:last-name %) )) )
22               :mental-age
23               54)
```

THREADING MACROS

Thread-last macro

```
1  (def numbers [1 2 3 4 5 6 7 8 9 0])
2
3  (take 2 (filter odd? (map inc numbers)))
4 => (3 5)
5
6  (->> numbers
7      (map inc)
8      (filter odd? )
9      (take 2))
10 => (3 5)
```

THREADING MACROS

Thread-as macro

```
1 (def points {:vals [1 2 3 4 5 6 7 8 9 0]} )  
2  
3 (take 2 (filter odd? (map inc (get points :vals))))  
4 => (3 5)  
5  
6 (as-> points $  
7   (get $ :vals)  
8   (map inc $)  
9   (filter odd? $)  
10  (take 2 $))  
11 => (3 5)
```

THREADING MACROS

Thread-some macros

```
1 (def user { :name "Alex" })
2
3 (-> user
4     (get :mental-age) ;; => nil
5     (str " years"))
6 => " years"
7
8 (some-> user
9     (get :mental-age)
10    (str " years")))
11 => nil
```

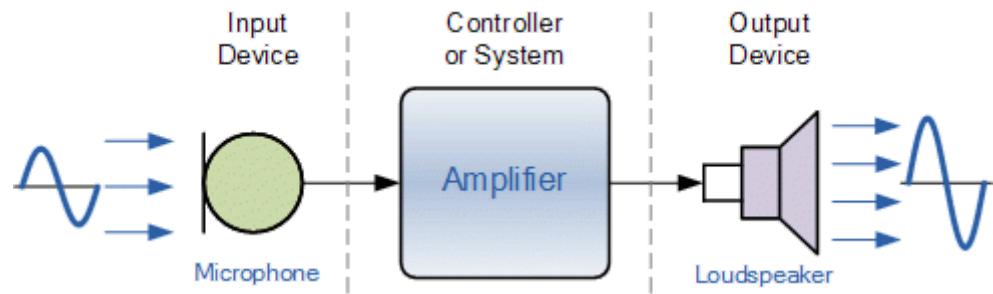
THREADING MACROS

Thread-cond macros

```
1 (def month-of-birth "Jan")
2
3 (def motivating-Mars? true)
4
5 (def equalizing-Saturn? false)
6
7 (cond-> []
8   (= month-of-birth "Jan") (conj "Capricornus")
9   (= month-of-birth "Feb") (conj "Aquarius")
10  ; ; ...
11  motivating-Mars? (conj "You would be unstoppable as"
12  equalizing-Saturn? (conj "Strike a balance between"
13 => [ "Capricornus" "You would be unstoppable as the moon for"
```

TRANSDUCERS

A transducer is a composable higher-order reducer. It takes a reducer as input, and returns another reducer.



```
1 (def xs [1 2 3 4 5 6 7 8 9 0])
2
3 (->> xs
4     (map inc)
5     (filter odd? )
6     (take 2))
7 => (3 5)
8
9 ; ; map = transform => reducer => reducer
10
11 (def tr (comp (map inc)
12                 (filter odd? )
13                 (take 2)))
14
15 (reduce (tr conj) [] xs)
```

```
8
9  ;; map = transform => reducer => reducer
10
11 (def tr (comp (map inc)
12                  (filter odd? )
13                  (take 2)))
14
15 (reduce (tr conj) [ ] xs)
16 => [3 5]
17
18 (into [ ] tr xs)
19 => [3 5]
20
21 (transduce tr conj [ ] xs)
22 => [3 5]
```

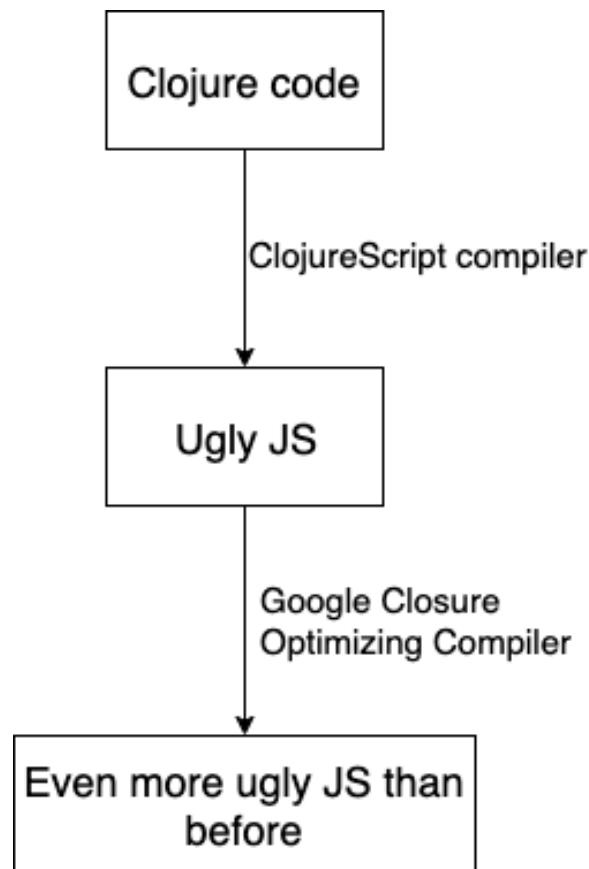
TRANSDUCERS

Details:

Returns a lazy sequence of the first `n` items in `coll`. Returns all the items if there are fewer than `n`.

Returns a stateful transducer when no collection is provided.

PERFORMANCE



```
1 (def xs [1 2 3 4 5 6 7 8 9 0])
2
3 (->> xs
4     (map inc)
5     (filter odd? )
6     (take 2))
```

```
1 cljs.user.xs = new cljs.core.PersistentVector(
2     null, 10, 5, cljs.core.PersistentVector.EMPTY_NODE, [(1
3 )];
4 cljs.core.take.call(
5     null,(2),cljs.core.filter.call(
6         null,cljs.core.odd_QMARK_,cljs.core.map.call(
7             null,cljs.core.inc,cljs.user.xs
8             )
9         )
10 );
```

ECOSYSTEM

- Leiningen ~ package.json (dependencies + scripts)
- shadow-cljs ~ webpack
- Reagent/Om/Rum - React wrappers
- Re-frame - Redux on steroids



Reagent



THERE IS ALWAYS A FLY IN THE OINTMENT



BUNDLE SIZE

Module: :main [JS: 474.49 KB] [GZIP: 124.56 KB]

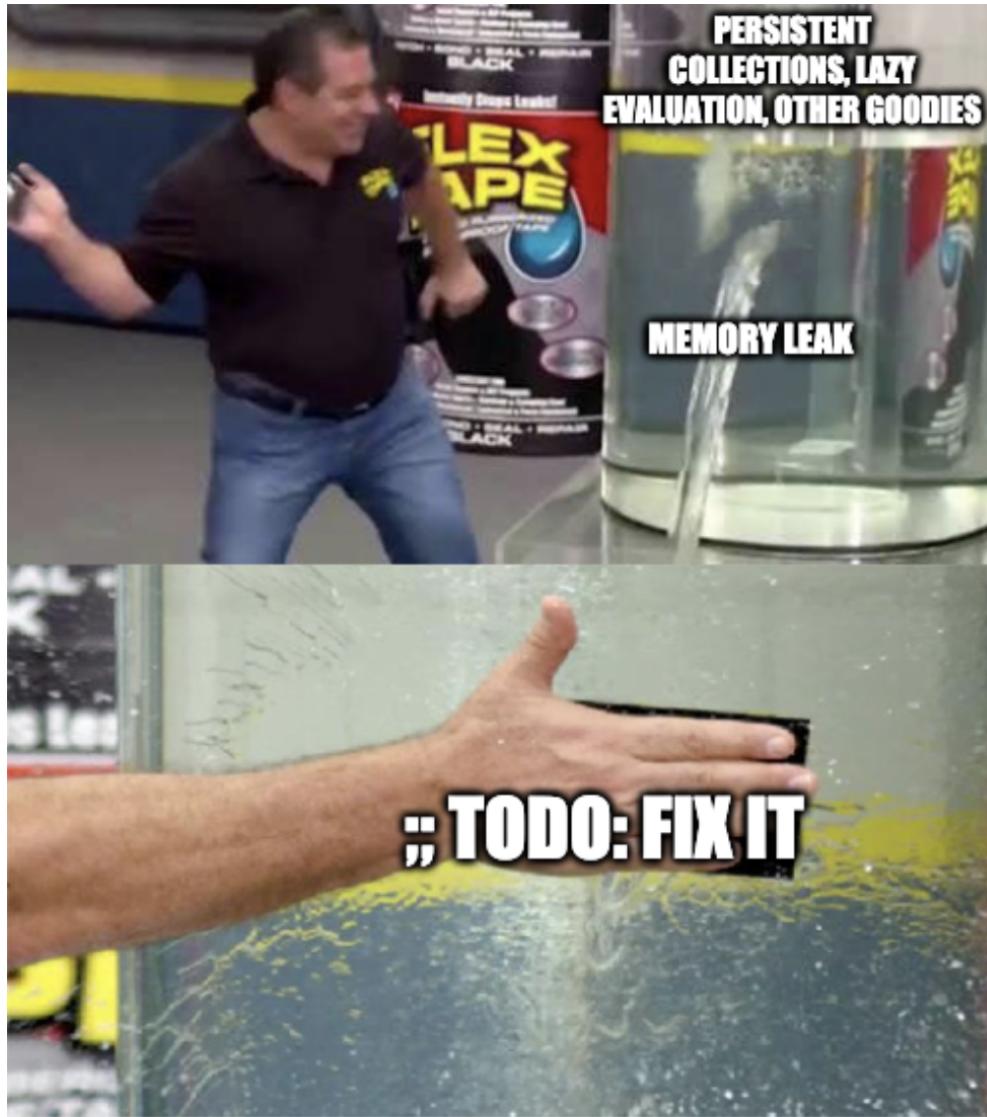
Group	Optimized	%
+ org.clojure/clojurescript @ mvn: 1.10.879	192.3 KB	40.6 %
+ react-dom @ npm: 16.9.0	111.25 KB	23.5 %
+ src	36.83 KB	7.8 %
+ org.clojure/google-closure-library @ mvn: 0.0-20201211-3e6c510d	36.55 KB	7.7 %
+ reagent @ mvn: 0.9.1	19.75 KB	4.2 %
+ org.clojure/tools.reader @ mvn: 1.3.6	16.03 KB	3.4 %
+ cljs-ajax @ mvn: 0.7.3	15.82 KB	3.3 %
+ re-frame @ mvn: 0.10.7	13.53 KB	2.9 %
+ bidi @ mvn: 2.1.5	8.8 KB	1.9 %
+ react @ npm: 16.9.0	6.57 KB	1.4 %
+ scheduler @ npm: 0.15.0	5.44 KB	1.1 %
+ Generated Files	3.14 KB	0.7 %
+ kibu/pushy @ mvn: 0.3.8	2.83 KB	0.6 %
+ com.cognitect/transit-js @ mvn: 0.8.874	2.15 KB	0.5 %
+ object-assign @ npm: 4.1.1	978	0.2 %
+ day8.re-frame/http-fx @ mvn: v0.2.0	940	0.2 %
+ com.cognitect/transit-cljs @ mvn: 0.8.269	348	0.1 %

RealWorld Comparison 2020

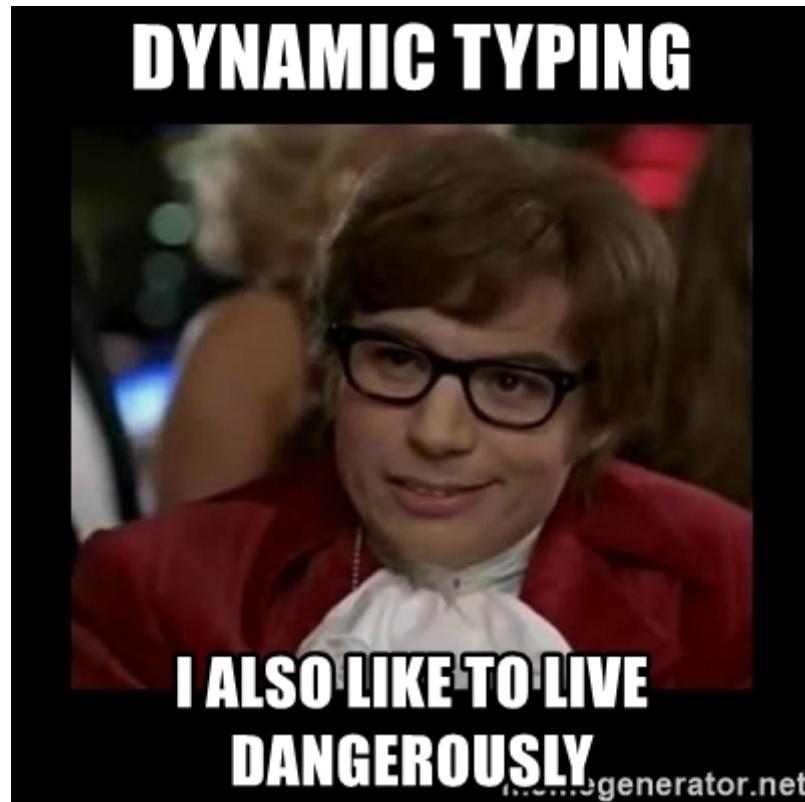
PERFORMANCE



MEMORY



DYNAMIC TYPE SYSTEM



SUPPORT

Closed ✓ Fixed

Pinned fields x

Click on the ⚡ next to a field label to start pinning.

Details ^

Assignee	 Unassigned
Reporter	 [REDACTED]
Labels	ft performance
Approval	Ok
Patch	Code
Priority	 Major
Affects versions	Release 1.5
Fix versions	Release 1.10.2

Created 30 May 2012, 11:55 Updated 11 December 2020, 21:09 Resolved 11 December 2020, 21:09

 Configure

clojurists together

NO POPULAR SOURCE OF KNOWLEDGE

JS - please, read the documentation, there are also learnjavascript, tonns of "Learn JS in 10 mins", video courses, books, blog posts etc.

ClojureScript - hmm... we have Cheatsheet, maybe some API docs, 2-3 books (it's difficult to find, but you should try!)

JOB OPPORTUNITIES

It's difficult both to find a job and to hire



JAKE-CLARK.TUMBLR

HOW AND WHY WE USE CLOJURESCRIPT IN PRODUCTION

The screenshot shows the EDU Conference 2020 virtual platform. At the top, there's a navigation bar with the conference logo, date (July 14-16, 2020), and a message: "Your session starts in 10 minutes." Below the navigation is a sidebar with links: Town Hall, Community, Schedule, Speakers, Sponsors, Exhibitors, News, and About. The main content area features a banner for "EDU CONF 2020" with a photo of three people. A welcome message for "Joe" is displayed, followed by a description of the conference's purpose and a "Read more" link. Below this are stats: 19 Speakers, 23 Sessions, and 12 Sponsors. A "Platinum Sponsor" section is shown with a green logo. On the right side, there's a "Community" section listing profiles of attendees like Amy Hawkins, Ashley Jones, Darlene Steward, Jane Andrews, Jorge Robertson, Regina Cooper, Richard Colman, Rosemary Edwards, Ryan Peterson, Ted Flores, Theresa Pena, and Dianne Robertson. A "Distance Learning Best Practices" session is highlighted, featuring speakers Ryan Peterson and Ashley Jones. At the bottom, a "Up Next" section is partially visible.

EDU Conference 2020
July 14-16, 2020

Your session starts in 10 minutes.

Tutorial Enter Practice Room

Community

Everyone Leaderboard

Amy Hawkins
Consultant @ Learning Systems Inc.

Ashley Jones
Product Manager @ Anchor Systems

Darlene Steward
Provost @ Nebraska State University

Jane Andrews
Admissions @ ASU

Jorge Robertson
Director @ Highland Systems

Regina Cooper
Project Director @ Digital Promise

Richard Colman
Founder @ Remote EDUcate

Rosemary Edwards
Head of Programs @ NY Public Schools

Ryan Peterson
Head of Programs @ University of Souther...

Ted Flores
Professor @ Seleg School of Business

Theresa Pena
Product Manager @ Pedagog Systems

Distance Learning Best Practices

Ryan Peterson Ashley Jones

Up Next

Democratizing Access to Financial Aid

10:45am – 11:15am

Dianne Robertson

Powered by attendify

WE ALL WRITE CLOJURE

When you are the **Frontend** developer,
but also the **Backend** developer



CLOJURESCRIPT IS AN AWESOME TOOL TO MAKE THINGS FAST

**6 MONTHS TO RELEASE
DATE BUT WE GOT CLJS APPROVED**



SHIT, BRO, THAT'S ALL YOU HAD TO SAY

PERFORMANCE

The screenshot displays a digital conference interface for "EDU Conference 2020" from July 14-16, 2020.

Left Sidebar (Schedule):

- Schedule:** Mon, 14 | Tue, 15 | Wed, 16
- 10:00: Intro: Distance learning in 2020 (10:00am - 10:10am)
- 10:00am - 11:00am: Distance Learning Best Practices
- 10:45am - 11:15am: Democratizing Access to Financial Aid
- 11:00am: Workshop: Campus Leadership
- 11:30am - 11:45am: Workshop: Integrating International Students
- 2:00pm: Campus Life 3.0 (2:00pm - 3:00pm)

Middle Section (Session Details):

You are a speaker | **Tutorial** | **Enter Practice Room**

This session is scheduled for **July 14, 2020 at 10:00 am**

Distance Learning Best Practices

July 14, 2020, 10:00am - 11:00am

Tags: Distance Learning, Digital

Speakers

Ryan Peterson Head of Programs University of Southern...	Mitch Cooper Account Executive Anchor Systems	Joe Williams Marketing Director Acme Inc.
---	--	--

Right Sidebar (User Activity):

Joe Williams (Acme Inc.)

What's on your mind?

Regina Cooper (Project Director @ Digital Promise) 5m ago: This is a great session, so many actionable insights!

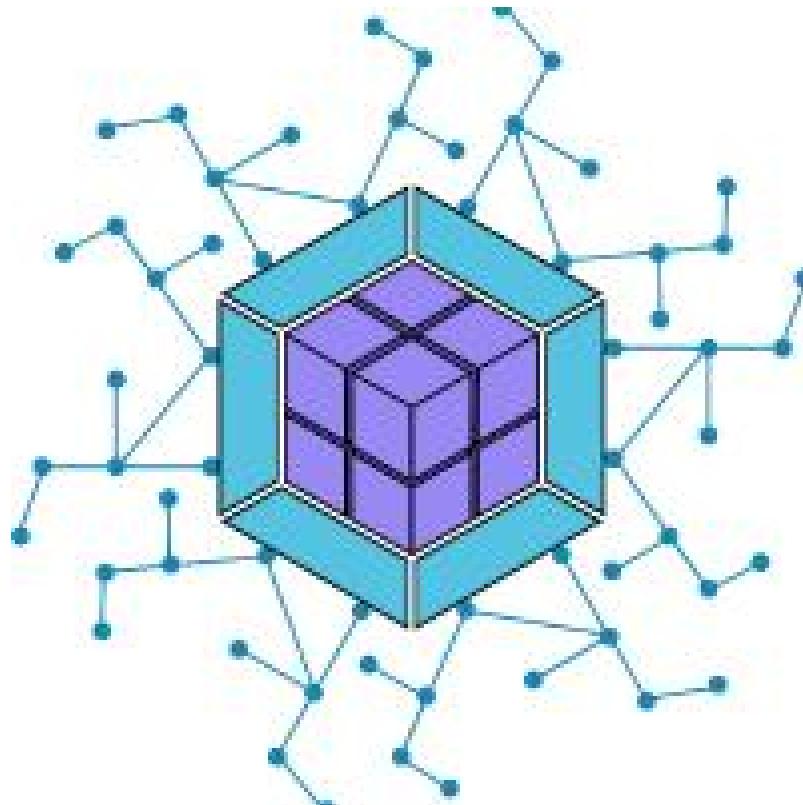
Amy Hawkins (Consultant @ Learning Systems Inc.) 10m ago: I love Ashley's idea about fostering remote participation and engagement!

OUR STACK

Backend: Ring, Compojure (routing), Aleph, HugSQL (DB requests), prismatic/schema (declarative data description and validation), custom RPC protocol

Frontend: Reagent/Rum (React wrappers) + re-frame/own state management solution (analogue of

HOW DO WE LIVE IN JS WORLD?

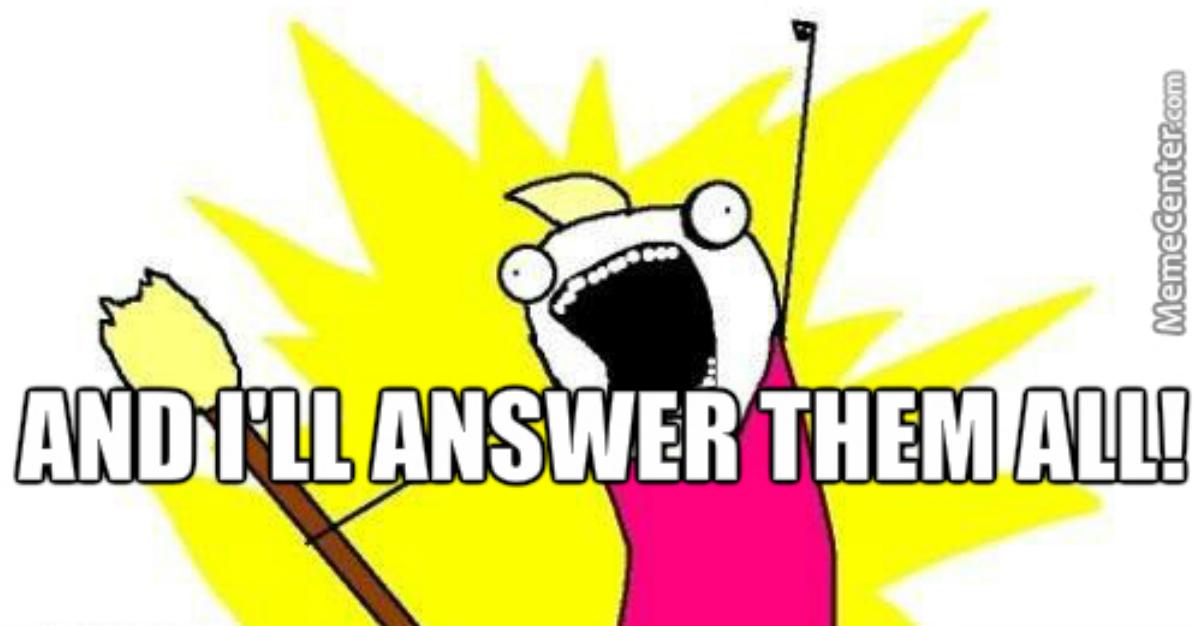


SOME RESOURCES TO LEARN MORE ABOUT CLOJURESCRIPT

- ClojureScript Unraveled [book](#)
- Clojure for the brave and true [book](#)
- The joy of Clojure [book](#)
- Why I chose ClojureScript over JavaScript [post](#)

Q&A

ASK ALL THE QUESTIONS



(MemeCenter.com)

THANKS FOR ATTENTION!