WHY CLOJURESCRIPT?

LILY M. GOH

@LilyMGoh
lily@paren.com

(paren.com)

Why ClojureScript?

(paren)

What is ClojureScript?

(paren)

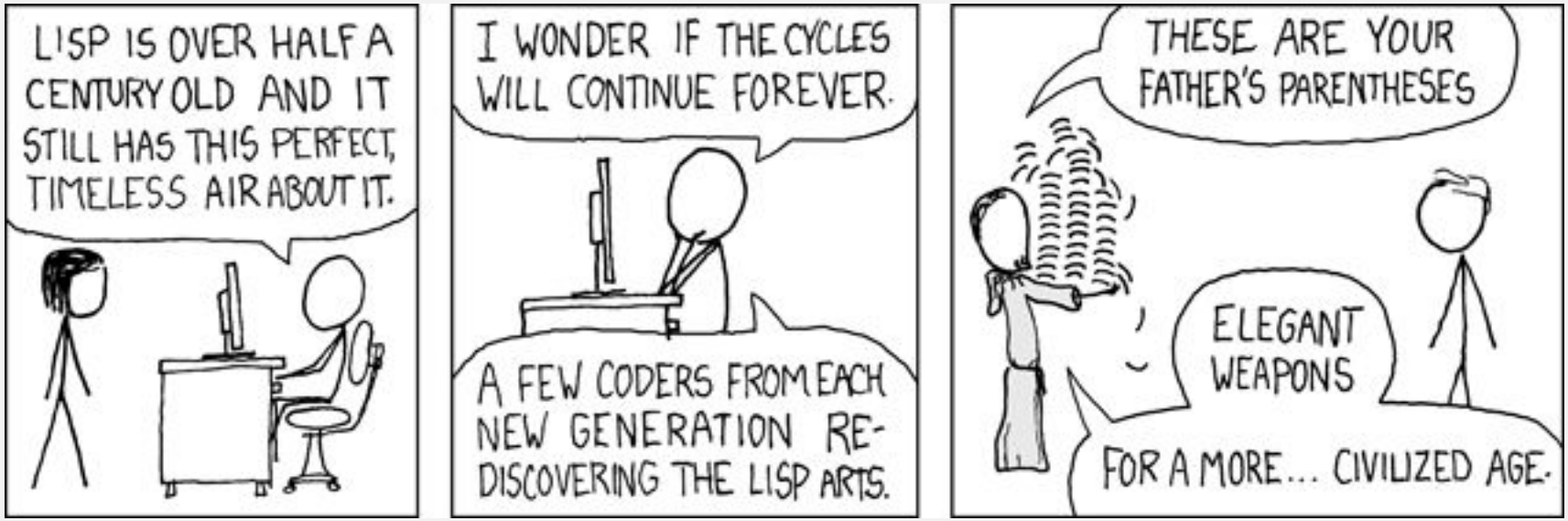
What is ClojureScript?

Lisp
Functional
Dynamically-typed
Compiled

(paren)

Lisp

(paren)



Mandatory XKCD

(paren)

LISt Processing

(paren)

Lisp in a Nutshell

JavaScript `func(arg1, arg2)`

ClojureScript `(func arg1 arg2)`

(paren)

Lisp in a Nutshell

```
(+ 1 2 3 4)
```

```
(if true
  "Yes"
  "No")
```

(paren)

Lisp is Homoiconic

(Code as Data)

(paren)

Lisp is Homoiconic

Code: `for, function, if, else, etc`
Data: `[], {}`

(paren)

Lisp is Homoiconic

```
(def my-list (list 1 2 3))
(map
 (fn [i] (+ i 1))
 my-list)
```

(paren)

Lisp is Homoiconic

Code as Data

Easy for Machine

Easy for Human

(paren)

Lisp is Extensible

(paren)

Macro

Function	Macro
Input Data	Input Raw Code
Evaluate Code	Manipulate Code
Runtime	Before Runtime

(paren)

Macro

```
(defmacro hello [name]
`(js/alert ~(str "Hello, " name)))

(macroexpand '(hello "New user"))
;=> (js/alert "Hello, New user")
```

(paren)

Compile v. Run time optimization

(paren)

Functional

(paren)


JavaScript
The Definitive Guide
O'REILLY
David Flanagan
Unearthing the Excellence in JavaScript
JavaScript:
The Good Parts
O'REILLY
YAHOO! PRESS
Douglas Crockford

(paren)

React is functional

(paren)

Immutable.js

```
const { Map } = require('immutable')
const map1 = Map({ a: 1, b: 2, c: 3 })
const map2 = map1.set('b', 50)
map1.get('b') // 2
map2.get('b') // 50

// Convert to JS Object
map1.toObject()
map2.toObject()
```

```
(def map1 {:a 1 :b 2 :c 3})
(def map2 (assoc map1 :b 50)
(:b map1) ; 2
(:b map2) ; 50
```

(paren)

```
const user1 = {
 firstName: "John",
 lastName: "Doe"
}

const user2 = user;
user2.firstName = "Jane";


user1 === user2 // true
```

```
(def user1 {:first-name "John"
            :last-name "Doe"})

(def user2
 (merge
  user1
  {:first-name "Jane"}))


(= user1 user2) ;; false
```

(paren)

```
const styles = {
 container: {
   flex: 1,
   alignItems: 'center',
   justifyContent: 'center'
 },
 text: {
   fontSize: 32
 }
};
<View style={styles.container}>
 <Text style={styles.text}>
   Hello, World!
 </Text>
</View>
```

```
(def styles
 {:container {:flex 1
              :align-items "center"
              :justify-content "center"}
  :text {:font-size 32}})


[rn/view {:style (:container styles)}
  [rn/text {:style (:text styles)}
   "Hello, World!"]]
```

(paren)

Compiled

(paren)

Google Closure Compiler

(paren)

Google Closure Compiler

(paren)

Google Closure Compiler

```
function hello(name) {
 alert('Hello, ' + name);
}
hello('New user');
```

```
alert("Hello, New user");
```

(paren)

Google Closure Compiler Limitations

Not well-documented

Subset of JS only

Type annotation

Statically access property in object

(paren)

JS is the biggest compile-to-JS language

(paren)

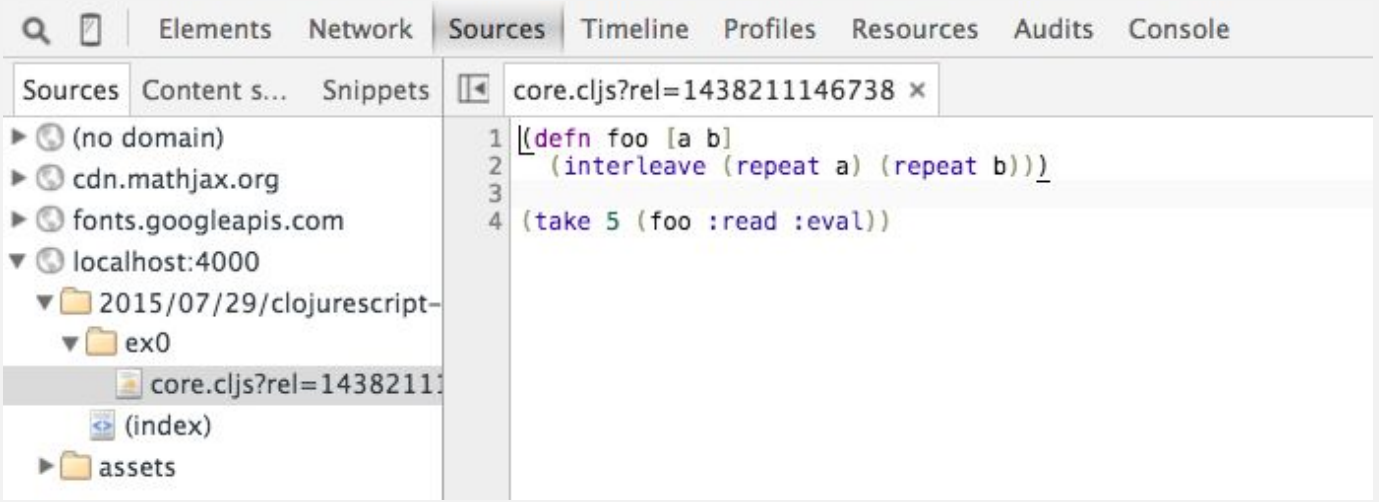
JS Interop

```
alert('Hello World');
```

```
(js/alert "Hello World")
```

(paren)

Source Map



Credit: David Nolen

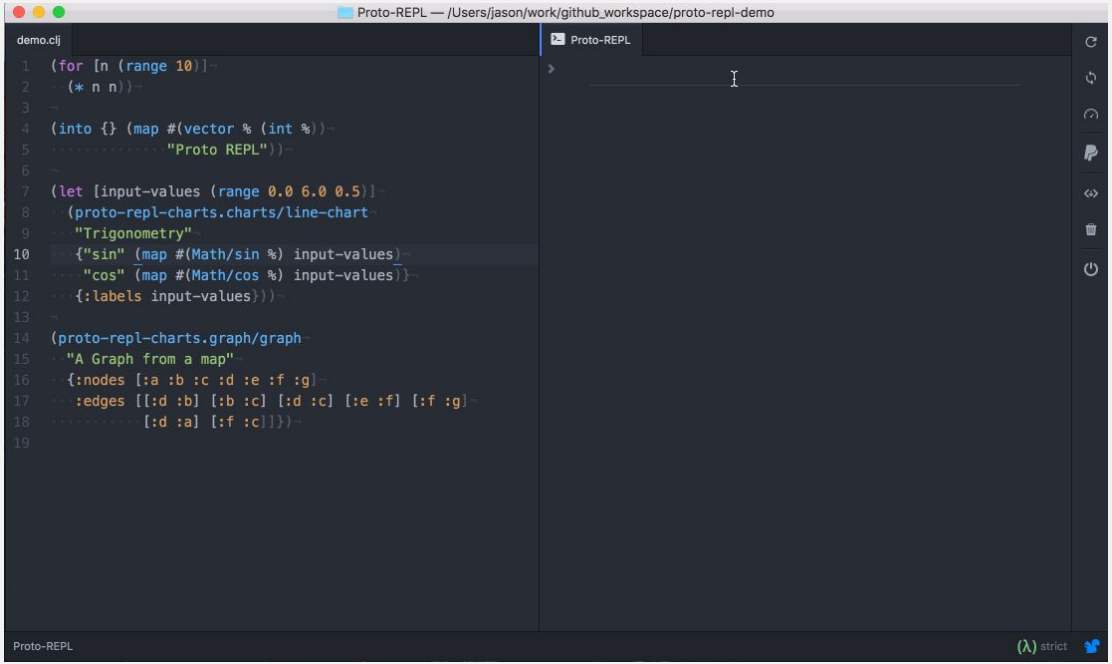
(paren)

REPL Driven Development

Read Eval Print Loop

(paren)

REPL Driven Development



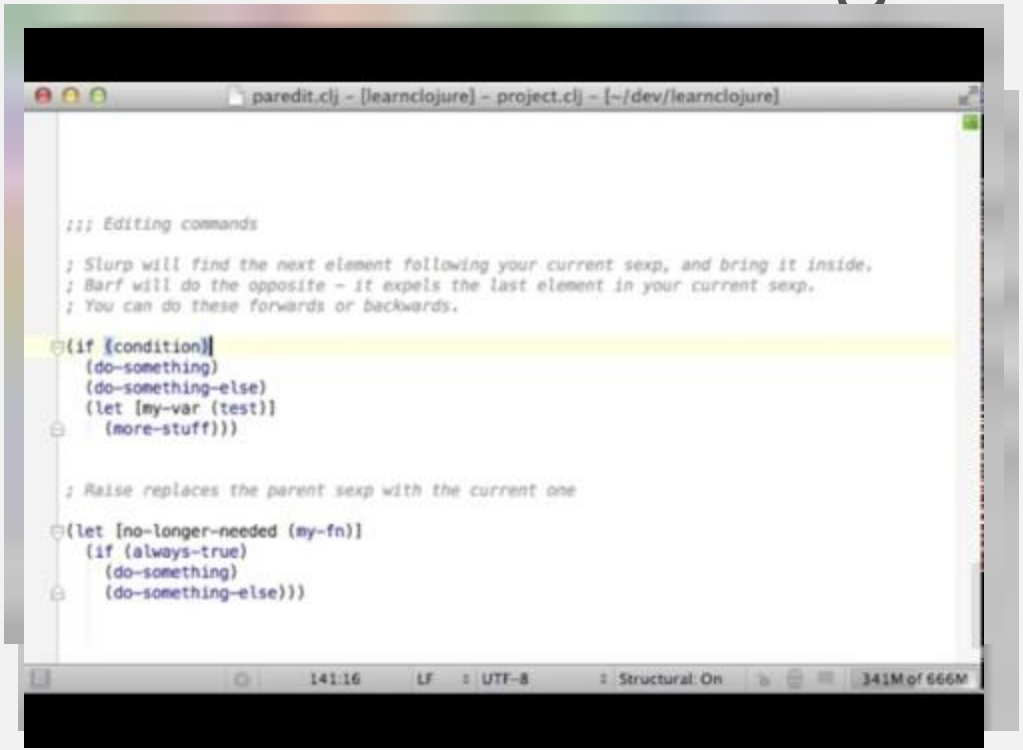
Credit: Proto REPL

(paren)

How to get started?

(paren)

Structural Editing



Credit: cursive-ide.com

(paren)

How to get started?

Structural Editing

Clojure for the Brave and True

“Simple Made Easy” talk

(paren)

hello@paren.com

LILY M. GOH

lily@paren.com

@LilyMGoh

(paren)