St. Louis
Clojure

ClojureScript Reagent Tutorial

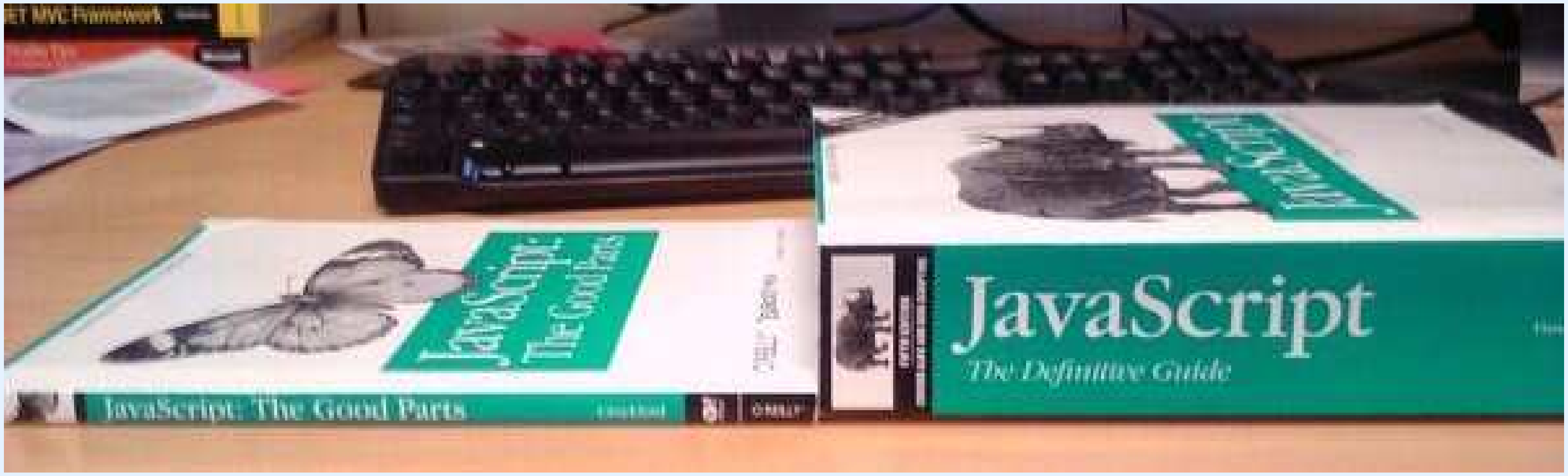
Christopher Mark Gore

cgore.com

Tuesday, May 17, AD 2016

We write Clojure at The Climate Corporation, and we're hiring! Come work with us!

Why ClojureScript? Isn't JavaScript good enough?



ClojureScript lets us use Clojure, a real lisp, in place of JavaScript.

ClojureScript versus JavaScript – Namespaces

JavaScript has no native namespacing.

ClojureScript's namespacing works the same as in Clojure.

One namespace:

```
(ns my.library)
...
```

Including another namespace:

```
(ns my.library
  (:require [other.library :as other]))
...
```

ClojureScript versus JavaScript – no variable hoisting

This actually does something in JavaScript other than raise an error, which is probably not what you want:

```
function printName() {
  console.log('Hello,␣' + name);
  var name = 'Bob';
}
```

ClojureScript versus JavaScript – destructuring binds

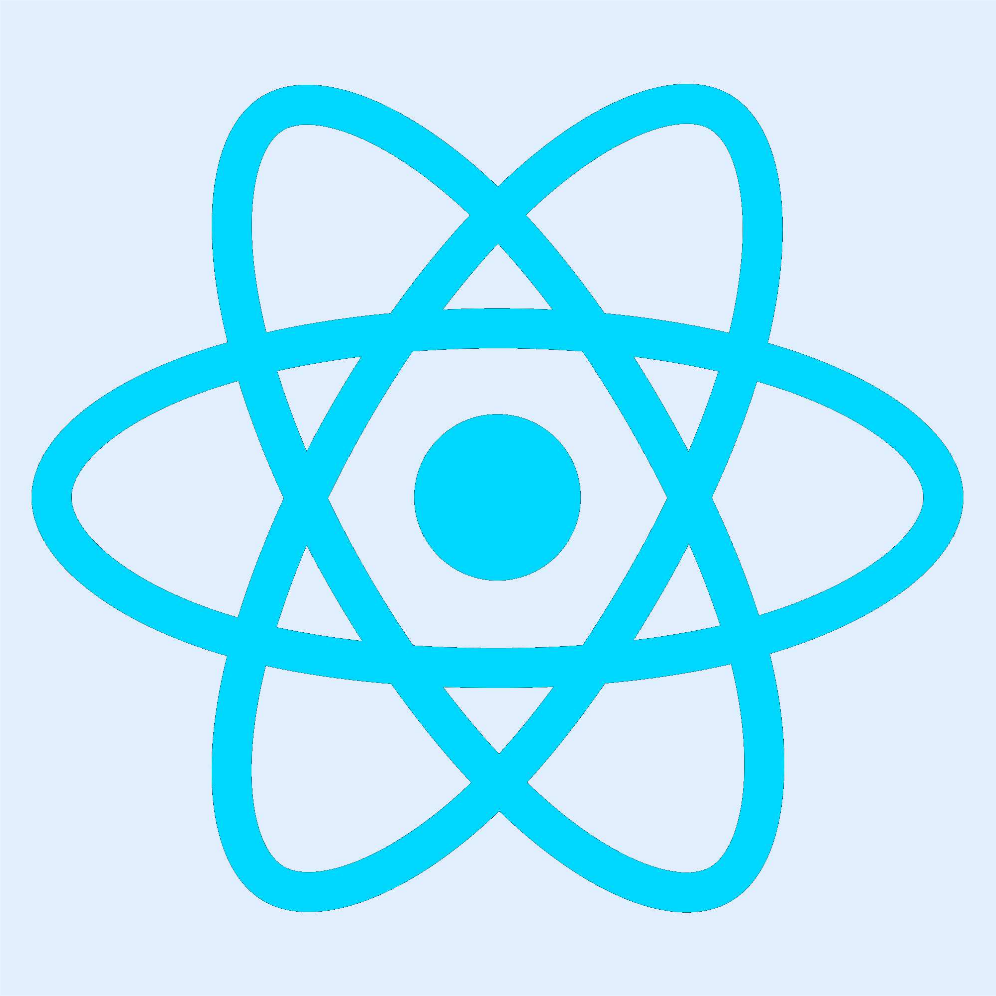
```clojure
(def m {:first "Bob"
        :middle "J"
        :last "Smith"})

(let [{:keys [first middle last]} m]
  ...)

(def color [255 255 100 0.5])

(let [[r g _ a] color]
  ...)
```

ClojureScript versus JavaScript – arbitrary keys for hash maps

```
// JavaScript -- Only string keys allowed
var m = {
  "foo": 1,
  "bar": 2
};
```

```
;; ClojureScript -- Arbitrary keys allowed
(def m { [1 2] 3
        #{1 2} 3
        '(1 2) 3 })
```

Reagent is a wrapper around React.js for ClojureScript.



Why do we want to use React.js?

- React.js comes out of Facebook originally, but is used everywhere now.
- React.js dates back to 2011.
- React.js is good at creating user interfaces.
- If you are used to MVC, React.js is the **V**.

Conclusion

Questions?