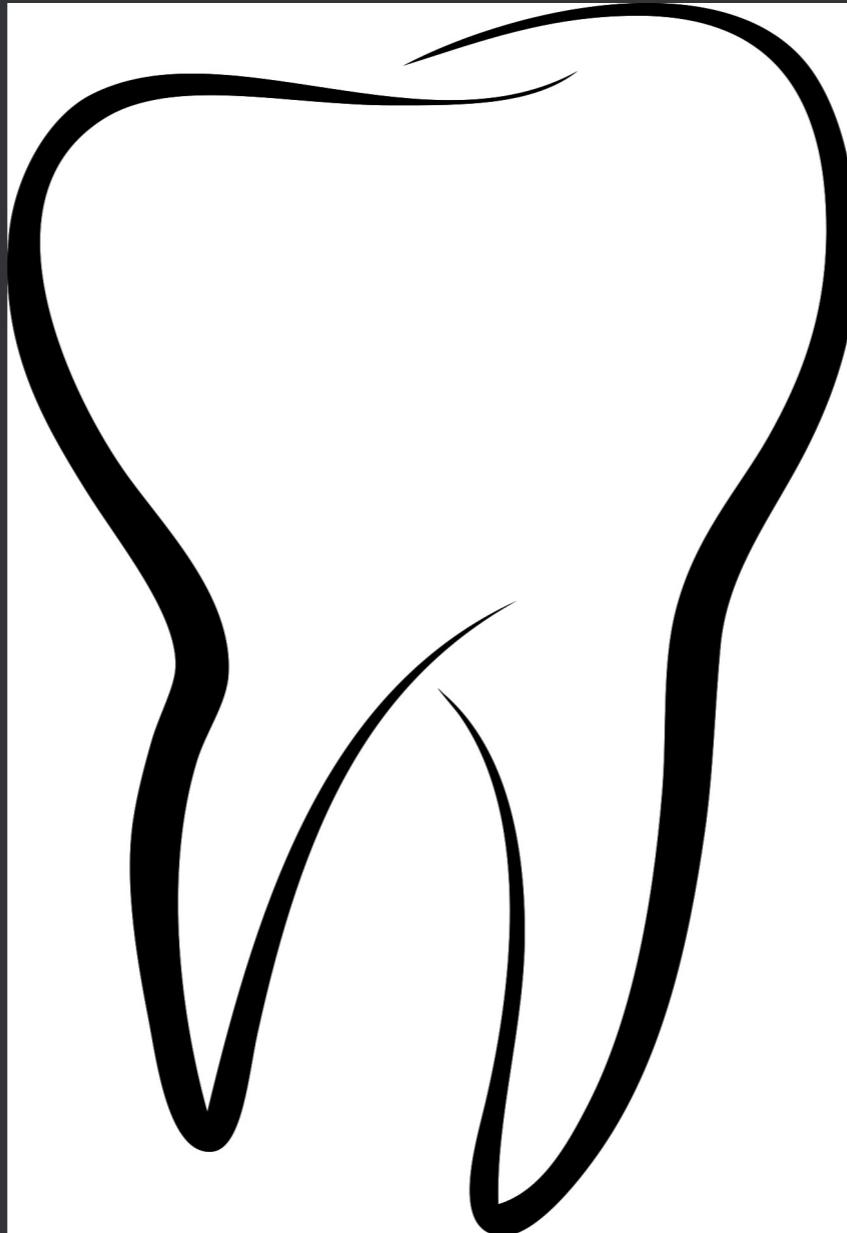


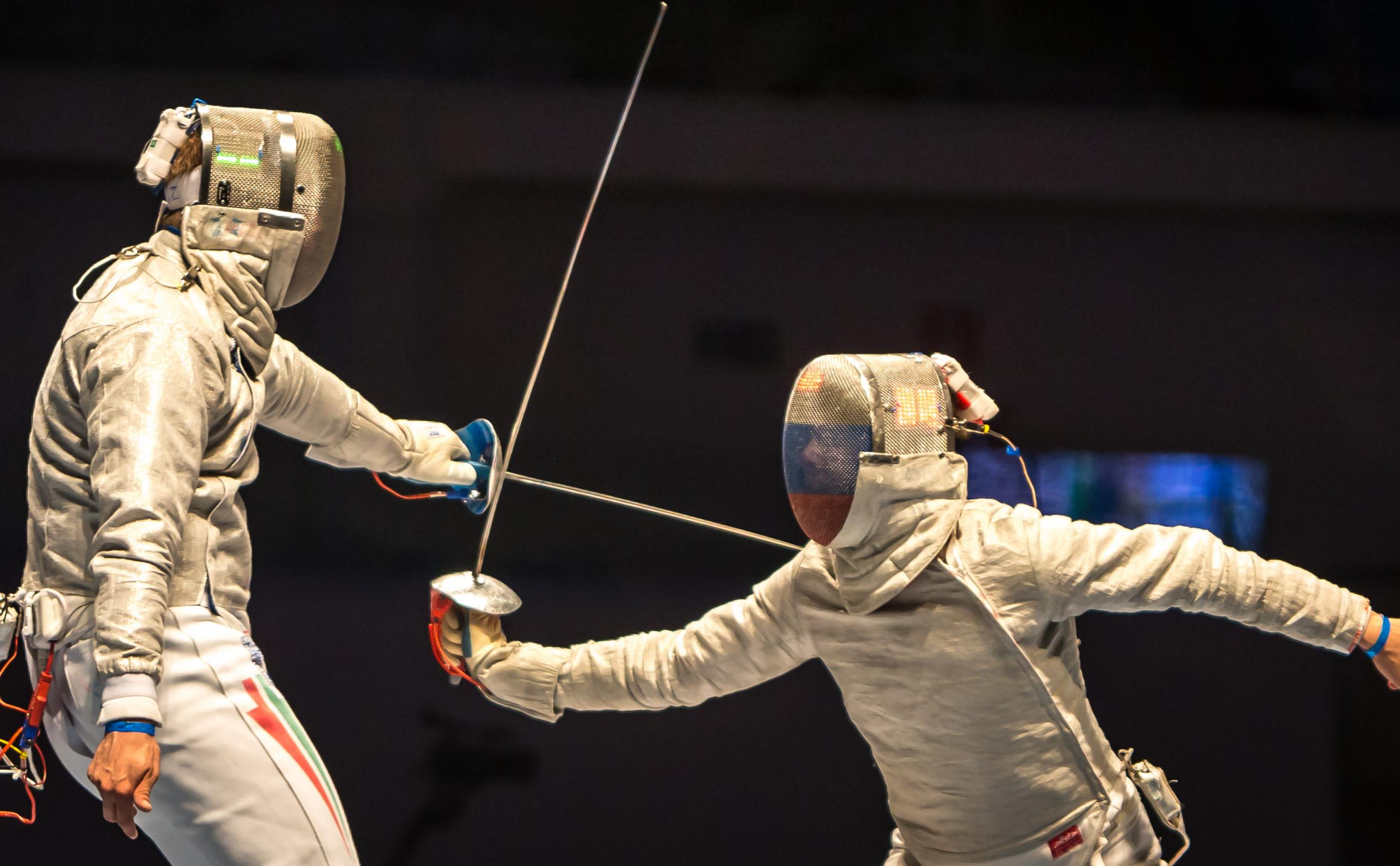
Code Defines Fun

Through Clojure and Functional Programming

Code Defines 😊



Code Defines



Code Defines



So many ways to have FUN

- Travels
- Sports
- Games
- Music
- Movies
- Books
- ...

Even just software related

- Solve hard problem
- Earn a lot of money
- Build product that people love
- And so on ...

I just want to share one perspective of fun
with code

The Joy of Clojure

Thinking the Clojure Way

Michael Fogus
Christopher J. Houser



MANNING



Aspects of FUN

- Feedback & Interactivity
- Confident
- Powerful
- Front-end
- Innovation

Clojure

- Functional language
- Mainly compile to JVM byte code and JavaScript
 - Hosted, embraces host language
- Data oriented

```
(println "Hello, " "World")
```

Function

```
(defn say-hello [name]  
  (println "Hello," name))  
  
#(println "Hello," %)
```

Data structures

```
[ "Orange" "Apple" "Microsoft" ]
```

```
{1 "Monday" 2 "Tuesday" 3 "Wednesday" 4 "Thursday"  
5 "Friday" "6" "Saturday" :7 "Sunday" }
```

```
#{"Pen" "Apple" "Apple Pen"}
```

Aspects of FUN

- *Feedback & Interactivity*
- Confident
- Powerful
- Front-end
- Innovation

REPL

- Read, eval, print, loop
- Inherited from Lisp

Demo

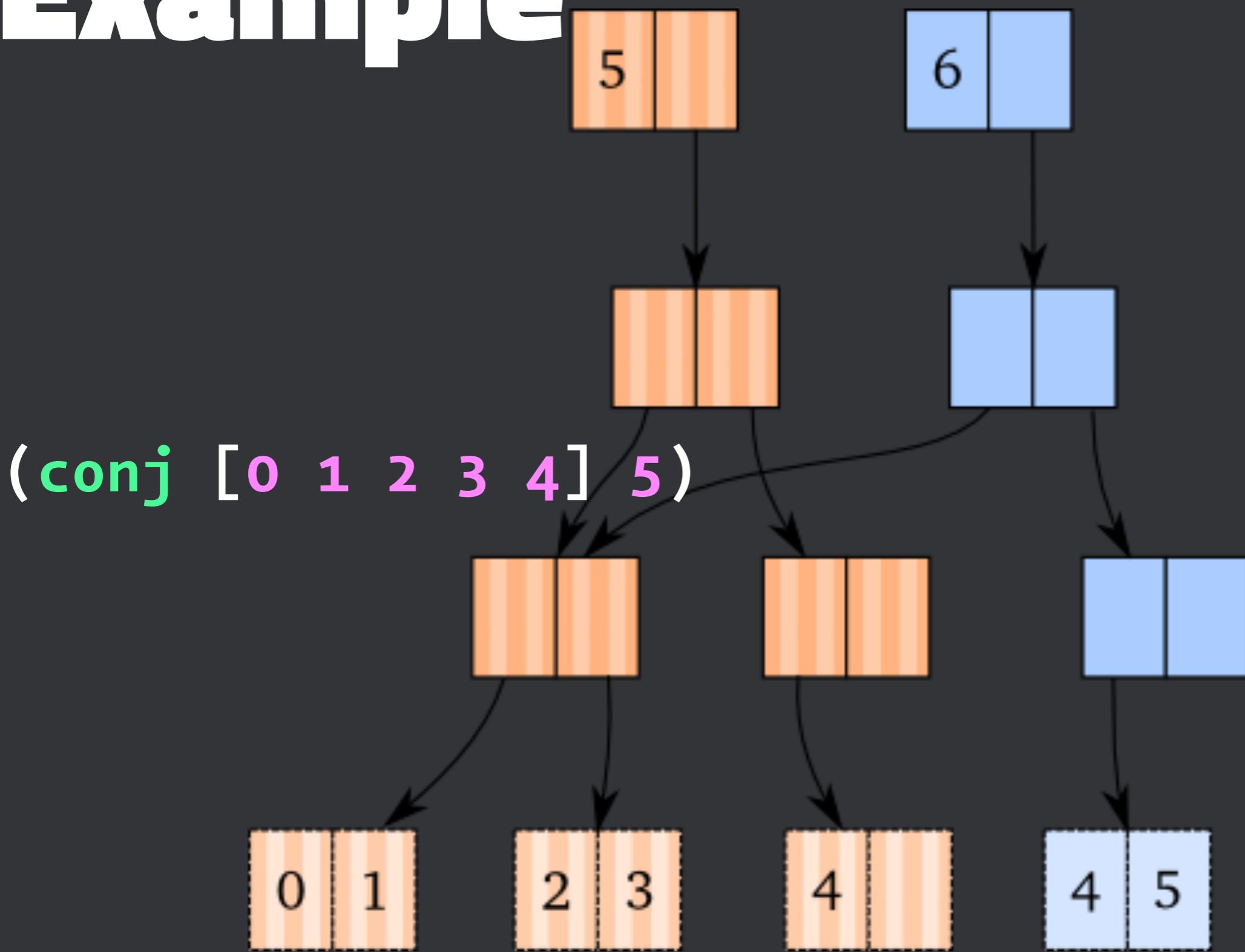
Aspects of FUN

- Feedback & Interactivity
- *Confident*
- Powerful
- Front-end
- Innovation

Confident

- Immutable & Persistent data structure
 - Any operation on a data structure only efficiently creates a new one
 - No more defensive copy
 - Simple concurrency
- Small & stable core
 - Macro

Example*



* <http://hypirion.com/musings/understanding-persistent-vector-pt-1>

```
(get {::a 1 ::b 2} :a)
(assoc {::a 1 ::b 2} :c 3)

(get-in {::a {::b {::c {::d 5}}}} [:a ::b ::c ::d])
(assoc-in {::a {::b {::c {::d 5}}}} [:a ::b ::c ::d] 6)

(update-in {::a {::b {::c {::d 5}}}} [:a ::b ::c ::d] inc)
(update-in y [:a ::b ::c ::d] #(+ % 1))
```

Collection functions

```
(group-by :name [{:name "David" :age 36}  
                  {:name "Michael" :age 40}  
                  {:name "David" :age 25}])
```

Some data structures are also function

```
(filter even? (range 20))
```

```
(#{3 8 6} 6)                      (#{3 8 6} 5)
```

```
(filter #{3 8 6} (range 10))
```

```
(filter (comp #{3 8 6} :id)
        [{:id 2 :name "Adam"}
         {:id 3 :name "Albert"}
         {:id 4 :name "Alex"}])
```

Demo

Macro

```
Map charNumber = new HashMap();  
charNumber.put("A", 1);  
charNumber.put("B", 2);  
charNumber.put("C", 3);  
  
(doto (java.util.HashMap.)  
  (.put "A" 1)  
  (.put "B" 2)  
  (.put "C" 3))
```

Demo

Aspects of FUN

- Feedback & Interactivity
- Confident
- *Powerful*
- Front-end
- Innovation

Powerful

- Concurrency
- Leverage power of host platform
 - All JVM and JS libraries available
- Extensibility

Concurrency support with ease

- CRUD app handles by database
- Many primitives, but we don't have time so I pick the most important one
- Atom

Atom

- Mutable reference to immutable data
- Get -> Calculate -> Check -> Set or Retry
- Fearless multithread programming

```
(def a (atom {:_x 1 :_y 2}))
```

```
@a
```

```
(swap! a assoc :_x 3)
```

Demo

Extensibility

- core.async
 - Steal from Go
 - Implemented as a library
 - Macro

Demo

Aspects of FUN

- Feedback & Interactivity
- Confident
- Powerful
- *Front-end*
- Innovation

ClojureScript

- Compiles Clojure to JavaScript
- Should work with all the JS runtimes, more popular in browser
 - Node.js, React Native, Electron, Google Spreadsheet
- Uses Google Closure compiler / library
 - Behind Gmail
 - Dead code eliminations
 - Rich set of libraries

Demo ClojureScript app

Aspects of FUN

- Feedback & Interactivity
- Confident
- Powerful
- Front-end
- *Innovation*

Innovation

- Keep up latest great ideas
 - Go style CSP
 - React
- Stable core and backward compatibility in mind
 - Macro & Rich Hickey
- Keep up with future platforms
 - Hosted design

Few last points

- Learn library not syntax
- Small community achieving big things
- One person can achieve more things

FB: Clojure in Thai
thank you

Credits

- <https://pixabay.com/en/tooth-dentist-bite-zahntechnik-1501321/>
- <https://www.pexels.com/photo/2-man-playing-sparring-90772/>
- <https://www.pexels.com/photo/relaxation-relax-cats-cat-96428/>
- <http://www.joyofclojure.com/the-book/>