

closure

why & how

hey!

I'm Vassilis,
backend engineer @ Splunk (ex-Bugsense)

me and FP

“...you have about 2h to learn about LISP (scheme) syntax in order to start programming with our custom LISP-like language...”



closure

“Lisp for the 21st century”



Rich Hickey (creator of clojure)

Why did I write yet another programming language? Basically because I wanted:

- A Lisp
- for Functional Programming
- symbiotic with an established Platform
- designed for Concurrency

another series of “*why clojure?*”



LISP dialect

- To beat the averages (pg) :D
- Dynamically typed
- Homoiconicity (code as data, data as code, macros)
- Clear syntax
- Expressive
- REPL

Functional

- Pure Functions
- Immutability
- No Side Effects
- Clean code

JVM

- JVM → 5.5 billion JVM-enabled devices
(nuff said :D)

Concurrency

- No Locks
- Immutable data structures
- Language level concurrency primitives (atoms, refs, futures, agents, promises)
- Software Transactional Memory

what I like about clojure

- Exciting and fun
- Powerful standard lib
- Excellent external libs (clojurewerkz FTW)
- Great community
- Rich Hickey presentations

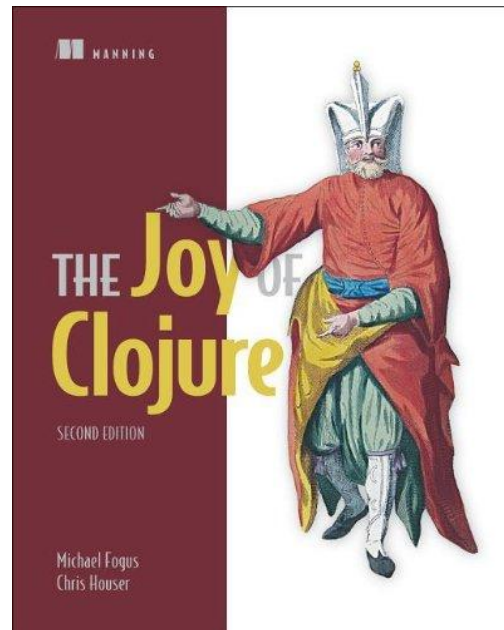
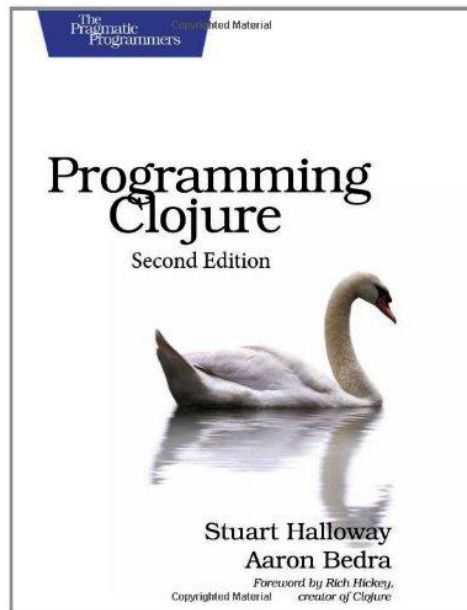
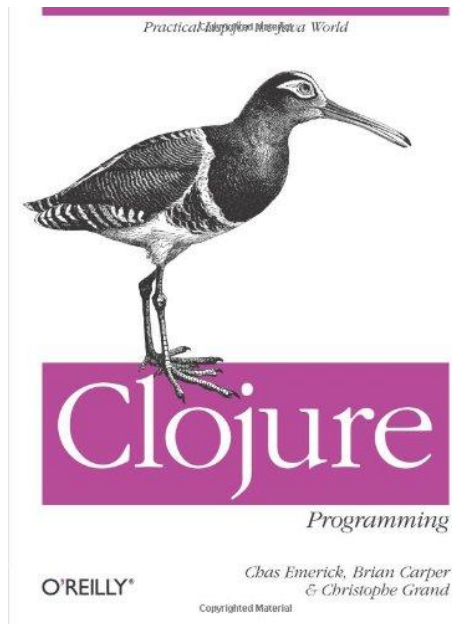
how to learn clojure?



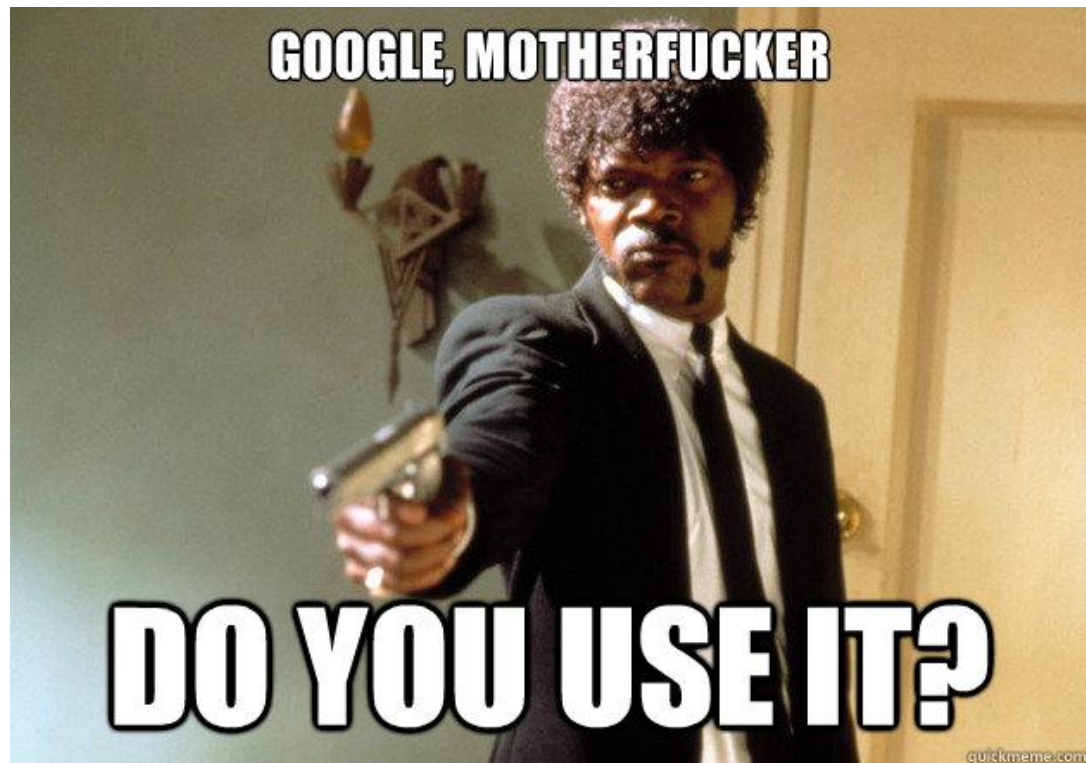
Many -awesome- tutorials

- <http://aphyr.com/tags/Clojure-from-the-ground-up>
- <http://www.braveclojure.com/>
- <http://hitchhikersclojure.com/>

Books



Google



Stack Overflow



personal advice :)

build something awesome using clojure!

thank you!