Clojure: LISP Reincarnated on the JVM (as a four-year-old wonder)

Dublin GTUG, 30th Aug 2011.

Kevin Noonan (@kevin_noonan)

"All problems in computer science can be solved by another level of indirection."
- David Wheeler.

The primary target platform of a programming language is wetware: the brains of developers.

This is especially true of a new language.

Contents

introduction
basic syntax
control flow
data types
working with Java
sequence library
concurrency
advanced features
web development
wrapping up

introduction

Hello Dublin GTUG

A command-line 'Hello World':

```
(println “Hello Dublin GTUG”)
```

A graphical 'Hello World' (with a Swing popup dialog):

```
(javax.swing.JOptionPane/showMessageDialog
    nil "Hello Dublin GTUG")
```

quick start

On Ubuntu 9.x (or later):

```
sudo apt-get install rlwrap clojure
```

No matter your platform, pick a plugin for your editor or IDE:

http://disclojure.org/where-to-start/

quick start on Mac OS X

You must have Java 1.5 or later installed & JAVA_HOME set:

http://www.oracle.com/technetwork/java/javase/downloads/jdk6-jsp-136632.html

Install ClojureX:

```
git clone git://github.com/citizen428/ClojureX.git
```

```
cd ClojureX
```

```
bin/create_symlink
```

ClojureX also works on OpenSolaris, Linux & Windows (under Cygwin),
with various installation instructions.

quick start on Windows

You must have Java 1.6 installed:

http://www.oracle.com/technetwork/java/javase/downloads/jdk6-jsp-136632.html

Download the installer for “cljr”

http://incanter.org/downloads/cljr-installer.jar

```
java -jar cljr-installer.jar
```

add `c:\.cljr\bin` to your system `PATH`

(“cljr” also runs on other platforms with similar steps to install.)

clojars

http://clojars.org/

Repository for Open Source projects in Clojure.

Like PyPy to Pythonistas... or RubyForge for Rubyists.

Integrates with Leiningen, Maven & Ivy.

Use Clojars to look up the syntax to pull down the code you want.

Leiningen

Like Rake for Ruby. Or Ant for Java.

Offers dependency management & build tool.

Lein is built on Maven (don't dismiss it even if you hate Maven!)

I recommend installing Apache Maven (along with Lein):
http://maven.apache.org/

Alternatives to Lein: Maven itself, Gradle (Groovy's build-tool).

To install Leiningen

download "lein" script:

http://github.com/technomancy/leiningen/raw/stable/bin/lein

```
chmod 755 ./lein
```

```
./lein self-install
```

Sample Leiningen file

Simplest (generated) Leiningen build & dependency file:

```
(defproject sample "1.0.0-SNAPSHOT"
  :description "FIXME: write"
  :dependencies [[org.clojure/clojure "1.2.0"]
                 [org.clojure/clojure-contrib "1.2.0"]])
```

To install Cake

Cake: all the goodness of Leiningen with persistent JVMs for speed!

If you're a Rubyist:

```
gem install cake
```

If you're a Mac user:

```
brew install Cake
```

Otherwise:

https://github.com/flatland/cake/wiki/Installation

Summary of installation tips

First, ensure you have Java 1.5 or Java 1.6 installed.

On Mac or Linux (or other UNIX), install Cake.

On Windows, install Leiningen.

Either Cake or Leiningen can automatically download the Clojure jars (and most other libraries you may need).

From a Gekko

If Gordon Gekko were a software mogul,
he might have said:

"Never get emotional about software."

(with apologies to Oliver Stone)

With that in mind...

Dirty little secrets

Clojure is a LISP

Most hard-core Clojurists use Emacs

Clojure is just a JAR-file (Java library).

To use Clojure effectively: you must know (some) Java & make use of Java libraries.

Clojure *can* be used on several platforms, but it's mature enough *today* only on the standard JVM (& on JavaScript).

Clojure is outmatched for Web-development by Python+Django or Ruby-on-Rails.

Bright sides

You don't have to be a LISP bigot!

There's a plugin for any editor/IDE you like.

It's easy to deploy Clojure in a Java environment.

Many Clojure libraries are available. (Clojure is rising up the pop charts on Github.)

Given time, Clojure will be a viable option
in other environments, like .NET.

You can do web-development in Clojure today - or combine (say) Python+Django in the Web-tier with Clojure in the middle tier.

Big ideas

LISP: second-oldest language (after Fortran)

functional language

"symbiant" on JVM

dynamic language

All data is immutable by default!

The future is multi-core

Jargon

A “pure function” has no side-effects.

A “higher-order function” accepts functions as arguments.

Lazy sequences: evaluated only as needed (which enables Clojure to handle infinite sequences).

REPL: Read-Eval-Print Loop,
just like an interactive Python (or Ruby) interpreter.

STM: Software Transactional Memory, for concurrency.

basic syntax

Essential syntax

```
(fn arg1 arg2 argn)
```

```
(fn1  argX  (fn2 argY))
```

Commas are ignored in Clojure, they're just whitespace!

read from right-to-left

data types

Forms: the make-up of Clojure

Boolean	`true, false`
Character	`\b`
Keyword	`:key, :something`
List	`(“apple”, “pear”, “orange”)`
Map	`{:user “Jim”, :pass “theword”}`
Nil	`nil`
Number	`1.23, 9`
Set	`#{“alpha”, “beta”, “gamma”}`
String	`“Fine day for a stroll”`
Symbol	`java.lang.String`
Vector	`[9  7  5]`

control flow

Conditionals

```
(def danger true)

(if danger (println “Sound the alarm!”))



(def cost 10)

(cond (cost < 100)(println “cheap”)
      (cost >= 100) (println “expensive”))
```

Iteration

```
(dotimes [counter 3]
  (println "Loop run " (inc counter)))
```

```
Output:
Loop run 1
Loop run 2
Loop run 3
```

Avoid explicit loops.
They're not idiomatic in Clojure.

Instead use the sequence library.
Or use recursion.

working with Java

Working with java

import a Java library:

```
(import '(java.io.File))
```

create a Java object:

```
(String.)
```

call a method on a Java object:

```
(.nextInt rnd)
```

sequence library

Seq-able (say "SEEK-a-bull") data

All Clojure collections & all Java collections

Java arrays and strings

Directory trees in a filesystem

Trees of XML data

Input & output streams

Regular expression matches

Any class that implements Java's "Iterable" interface.

Simple Seq examples

Seq the file-system

```
(import '(java.io File))

(.listFiles (File. "."))



(map #(.getName %)
 (seq (.listFiles (File. "."))))



(count (file-seq (File. ".")))
```

concurrency

Reference types

A reference type is a mutable reference to immutable data

Four reference types: var, ref, atom, agent.

Reference types can hold any kind of data.

Dereference (read the value it contains) with the @ macro

A ref uses Software Transactional Memory.

Simple STM example with ref

```
(def forecast (ref "sunny with scattered
showers"))

@forecast

Output: "sunny with scattered showers"


(dosync (ref-set forecast "torrential rain
and thunder-storms"))

@forecast

Output: "torrential rain and thunder-storms"
```

Multi-process concurrency

NB: all of the foregoing refers to concurrency
within a single JVM.

Amit Rathore's Swarmiji for multi-process concurrency
(across several hosts):
http://github.com/amitrathore/swarmiji

And of course, Java has umpteen libraries for
SOAP, XMPP, CORBA, Terracotta etc.

advanced features

Macros

Code that generates code.

Two phases:
"macro expansion"
execution.

For example: a macro for “unless”

```
(defmacro unless [expr form]
   (list 'if expr nil form))
```

An “unless” macro in action

```
(def danger false)

(unless danger (println "Open the door."))
```

expands to:

```
(if danger nil (println "Open the door."))
```

output:

"Open the door”

Simple Multimethod example

The method to call is determined by the “class” function:

```
(defmulti gen-welcome-msg class)
```

```
(defmethod gen-welcome-msg User [x]
    (println "Hello " (.name x))
```

```
(defmethod gen-welcome-msg Admin [x]
(println "Red carpet treatment for you "
                (.name x))
```

Using a multimethod

Assuming we've already created a User object for "Jim"...

Usage:

```
(gen-welcome-msg user-jim)
```

Output: "Hello Jim"

Remember: we can *dispatch* based on *any* function, not just “class”.

Far more general and powerful than polymorphism in OO languages.

web development

Compojure

Compojure is the leading web-framework.

Inspired by (& comparable to) "Sinatra" for Ruby.

Can be used with “Ring” middleware.

Can be run on Google App Engine.

To install with Leiningen, add these line to project.clj...

```
[compojure "0.6.5"]
[ring/ring-jetty-adapter "0.3.11"]
```

..then run: `lein deps`

Sample Compojure code

```
(ns testweb.core
   (:use compojure.core, ring.adapter.jetty)
    (:require [compojure.route :as route]))

  (defroutes main-routes
    (GET "/" [] "<h1>Hello Dublin GTUG")
    (route/not-found "<h1>Page not
found</h1>"))

  (run-jetty main-routes {:port 8080})
```

Options for Web templating

Enlive (DSL)

http://wiki.github.com/cgrand/enlive/getting-started

(DSL means “domain-specific language”. In this case, it means writing web-page markup in Clojure code.)

Fleet

http://github.com/Flamefork/fleet

More like (say) PHP or JSPs: embed what you want in HTML.

Run your blog on Clojure

Leading European Clojurian, Lau Jensen's blogging software:

http://github.com/LauJensen/bestinclass.dk

Gary Burd's blogging software
(A word of warning: a challenge to build.)

Twister: Google App Engine web-framework
http://github.com/garyburd/twister

"gburd-blog": blog application built on Twister (no longer online)
~~http://github.com/garyburd/gburd-blog~~

Clojure on App Engine

“App Engine Magic”:
https://github.com/gcv/appengine-magic
(Lots of helpful utilities and abstractions for Clojure on GAE)

My appeal to Google's powers-that-be:
provide native support for Clojure concurrency on App Engine.

If you care, please click the "star" on this issue on Google's site:

http://code.google.com/p/googleappengine/issues/detail?id=2498
(And comment on the issue.)

wrapping up

Other platforms

Clojure on .NET (in beta):
http://github.com/richhickey/clojure-clr

ClojureScript
Clojure-to-JavaScript compiler:
https://github.com/clojure/clojurescript

(Don't confuse with *Closure*, Google's JavaScript optimiser.
Take a hint: talk to the Search team; don't pollute the google-space!)

Brief history

1958 LISP created by John McCarthy

1995 Java released by SUN, having been created by James Gosling.

Oct 2007 Rich Hickey releases Clojure, after years of private work.

Aug 2011 current version: 1.3 beta 2

State of Clojure

Vibrant global community.

Five published books.

Clojure.org: active Open Source development.

Clojure.com: established for training and consultancy

"State of Clojure" survey results 2011, by @cemerick: http://cemerick.com/2011/07/11/results-of-the-2011-state-of-clojure-survey/

Notable Open Source projects in Clojure

Crane

Manage distributed processes on Hadoop and Amazon EC2

https://github.com/getwoven/crane

Incanter

"...statistical computing and graphics environment for the JVM"

http://github.com/liebke/incanter

Pallet

"provision and maintain compute nodes"

(on public Cloud providers like Amazon EC2)

http://pallet.github.com/pallet/

Programming Clojure

The Pragmatic Programmers

Stuart Halloway

Foreword by Rich Hickey
author of Clojure

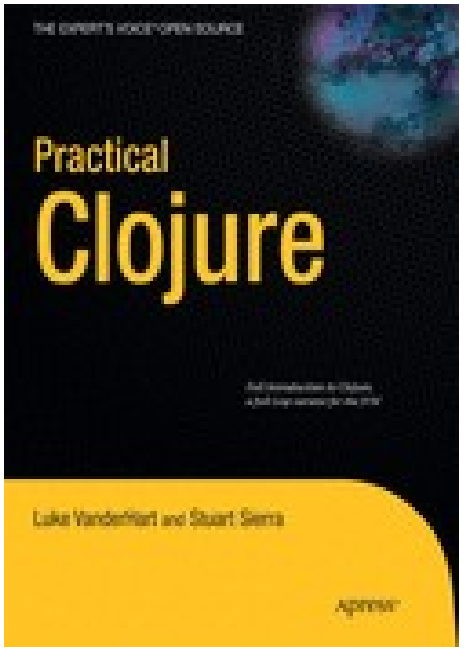
Edited by Susannah Davidson Pfalzer

May 2009

http://pragprog.com/titles/shcloj/programming-clojure

@stuarthalloway

Practical Clojure

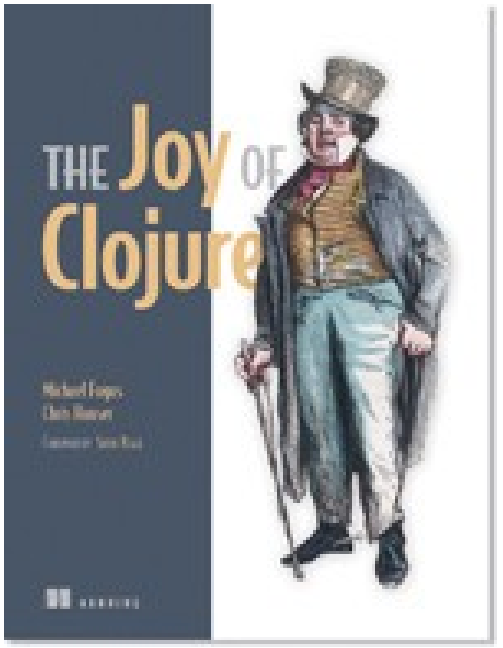


May 2010

http://apress.com/book/view/1430272317

Luke VanderHart & @stuartsierra

The Joy of Clojure

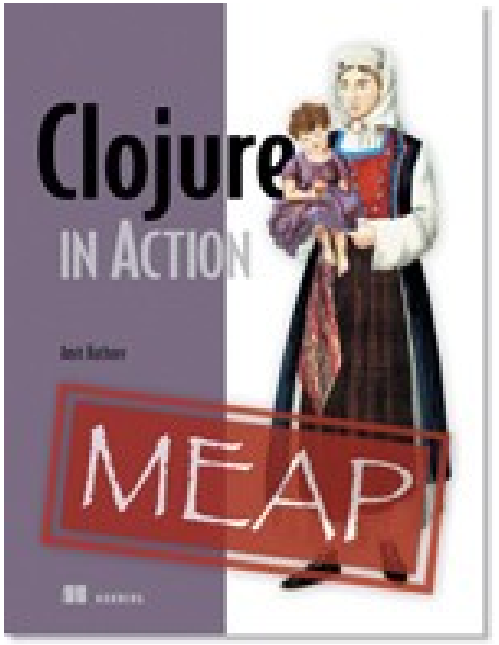


March 2011

http://www.manning.com/fogus/

@fogus & @chrishouser

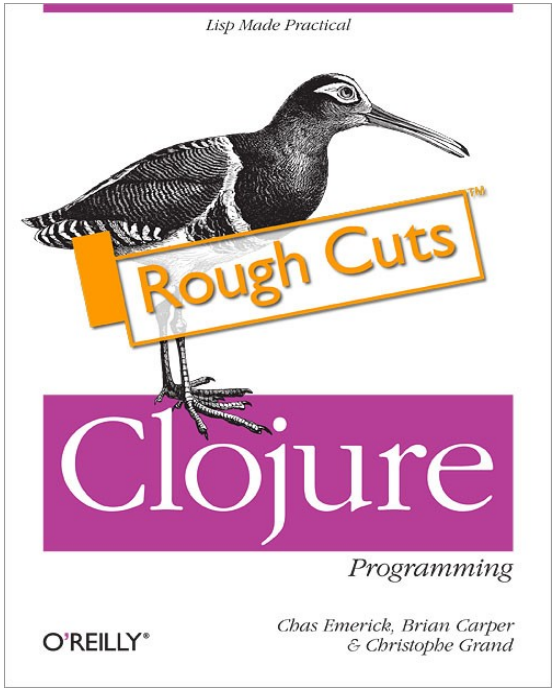
Clojure in Action



(Available in Acrobat format)
http://www.manning.com/rathore/

@amitrathore

Clojure Programming



(Available in Acrobat format)
http://oreilly.com/catalog/0636920013754

@cemerick, @BrianCarper, @cgrand

"Clojure Ireland”

@clojureireland

http://groups.google.com/group/clojure-ireland

Where to go from here?

Mark Volkmann's tutorial:
http://java.ociweb.com/mark/clojure/article.html

Satish Talim's tutorial:
http://clojure-notes.rubylearning.org/

ClojureDocs:
http://clojuredocs.org/

Daily digest of news:
http://disclojure.org

Best starting page on the Web for Clojure:
http://disclojure.org/where-to-start/

Conclusion

Clojure is a powerful, functional, dynamic language on the JVM.

Clojure is built on a raft of solid ideas from computer science.

Use your existing Java codebase.

Use the libraries and tools of the Java world.

Avoid the tar-pit of hand-written multi-threaded code.

Clojure is not your grandfather's LISP.

Questions?

Kevin Noonan

@kevin_noonan

knoonan Attt... gmail Dottt... com