Checklist:

- js_repl.cljs freshly run
  - display JS-REPL output
- counter demo template
  - freshly started, including server
  - counter tested
- get started repl project
  - repl disconnected
  - output window cleaned and closed
-

Zoom in three times in a new window

Workshop: Welcome to Clojure!

- Hej! Jag heter Peter Strömberg, jag jobbar på Agical, och jag älskar Clojure.
- Tack Ville, för att du ordnat det här! Tack för att jag får chansen att prata om Clojure och om interaktiv programmering.
- Check-in, 1min. Jot down
  - hope to take away / expectations
  - role,
  - Clojure experience so far
- Someone wants to share?
- Consider themself a Clojure programmer?
- Who has tried Clojure?
- Who knows very little about Clojure?

Give me Interactive Programming, or give me death!

Agical deal

❤ □ □ □

Scramble

From Nothing, Something

HP 9872C 8 Pen

PostScript

❤ □ □ □

Timeline:

Clojure at a glance

- ask questions
- expressions
- lists, function position
- defn macro
- 90% of clojure syntax

An Epigram in Programming

- Immutable data structures, powerful abstraction
  - so is lists
- One API
- Each class provides new API
- Clojure answers:
  - sequence abstraction
  - immutable data structures,
  - maps, vectors, sets

“Just use maps”

- 100+ functions on one data structure?
- Alan Perlis

Clojure data

What is Clojure?

- 2,5 years sabbatical.
  - Wanted to do something that matters.
  - Without caring what anybody else thinks about it.
- Fed up with Java and C++.
- Concurrency.
- Fond of LISP.
- Third take on a LISP for the JVM.
- dotLISP.
- Here to stay
  - Ecosystem funded by NuBank + many companies + the community

How is Clojure special?

- Pragmatic, you are not a jail
- Has changed my TypeScript
- Brackets, comma is whitespace
- 20% compared to Java not uncommon
- Just use maps! EDN
- Little boilerplate, small code bases
- Culture shift macros -> data, homoiconicity
- Abstractions, immutable, sequence, first class functions
- JVM and NPM ecosystems, interop strength
- Not Rich Hickey’s experimental platform
- Focused on business problems
  - Minimize accidental complexity
- Easy to create DSLs (but often you don’t need to)
- Attracts experienced developers, tops SO charts for highest payed
- Pragmatic, sometime interop instead of core
  - CLJS changed this a bit

Is Clojure perfect?

- Leiningen, or deps.edn?
- Error messages improved
- Frameworks are showing up
- If you know you know
- Tradeoffs

What’s in stability?

Rich Hickey makes Clojure special

- Clojure is not a cult, but
- Open Source, but not collaboratively built

Interactive Programming

- The REPL.
- You have already seen me use the REPL (Joyride)
- Literal programming, notebooks
- LISP and SmallTalk
- Clojure has a large gap to close, and is closing it

Calva

- Getting started repl, Java and Calva needed
  - `hello_repl.clj`
  - I skip paredit.clj (you shouldn't)
  - `welcome_to_clojure.clj`
  - the basics of Clojure
  - since we know how to use the repl ...
  - I hope you try this yourself and give me feedback.
- We'll skip learning Clojure here and now.
  - I was Interactive Programming.
    * app running
    * querying and modifying,
    * not a very obvious.
    * another app.
    * Two apps.

Demo: Interactive Programming

- Very simple app, a counter and also a wizard
- Clojure server + client app, both have REPL servers
- Change button color, render, shadow, notice state
- Inspect state on the client, app-db, wizard, subscription, dispatch on the counter
- Inspect state on the server, modify state, inspect request, modify behaviour
- the request is an open map
- Modify behaviour on the client, fizzbuzz
- Create branch fizzbuzz
- (mod n m) … defn fizz, installera fizz
- multiple-of?, map partial, 3 5 15, tables (range 1 19)

- defn fizz med multiple-of?
- defn fizz-buzz med if, sedan cond, installera fizz-buzz
- The file is not saved.
- Playing it safe. Gains much bigger with real programming.

Demo: Interactive Programming is a mind-set

- It can be easy <-> fully interactive
  - Small Talk, Clojure can get there
    - Joyride is almost there
- We can work more interactive though
- An example for JS
- `hello Hello ${s}!~;`
- `hello =`, we lack s-expressions
- `const s=`
- redeclare const
- promises work
- importing node modules

- not top level await, work-around
- updating imported files
- namespacing is lacking

I intend to try create a Clojure level experience.

Please give me feedback!

- I love Clojure and interactive programming.
- I want to get really good at presenting both.

e-mail to pez@agical.se:

- How likely would you recommend this presentation to a friend or colleague?
- 1, not at all likely
- 10, very likely
- What did you enjoy the most?
- Mention something you think was missing or could be improved
- Expectations?

Calva Clojure IDE setup

- `clj` is `clojure` with `rlwrap`
  - I very seldom use `clj`
- What about Leiningen?
- shadow-cljs
- Install Joyride for good measure 😀

A Minimal Clojure Project

- Defaults to `clojure`'s version of Clojure
- Defaults to `{:paths ["src"]}`
- directories -> dots in namespace
- Needs a dotted namespace
- underscore in files -> dashes in the namespace
- Will bite you
- The REPL doesn't care
- We'll get back to this. Open a new window

A Minimal ClojureScript Project

- Copy the project to your project directory
- Open in vscode
- Jack in **deps.edn ClojureScript built-in for Node**
- Check Calva’s Jack-in command
- Check index.html

Clojure data revisit

- Use the getting started repl project
  - Jack-in deps.edn
  - Create a new file `src/get_started/my_basics.clj`
  - `(ns get-started.my-basics)`
  - Load/Evaluate current file
  - Add a rich comment form, RCF
- No complex numbers

namespaces, vars, symbols

Special forms, macros, functions

- try (apply list :f 1 2 3)
- str
- fn, defn, def

Special forms, macros, functions

- All this can be combined
- REPL variables *1, etc

Immutable data

- Records and Tuples, coming to JavaScript
- evaluate to cursor

Sequences

- Vector in, sequence out