

U0NCTKEV8 : continue-trampolining?

U051SS2EU : yeah, I guess the answer of `(defmulti foo first)` `(fn? foo)` can be a guide there

U050ECB92 : -----Soooooooooooo who here has played with `tools.deps.alpha` and what do they think?

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

U11BV7MTK : i looked at the version compare and it seems like it will return `1.1` > `1.9`

U11BV7MTK : but i don't know how to crank it up since it's based on the pom file rather than a project file

U0NCTKEV8 : 10541c0254973ce045305499d4b5622073ede622 (is the trampoline commit fyi)

U11BV7MTK : i'm not a java guy

U11BV7MTK : it also specifies its own deps in the new edn format. does that make it "self-hosted"?

U050ECB92 : <@U11BV7MTK> if you have `mvn` installed you can crank it up using the commands in the README.

Use of mvn/pom is a one-time thing at setup. (It will probably be handled by the package manager when this stuff filters over there)

U11BV7MTK : thanks. i'll get that going tonight

U11BV7MTK : yeah i wanted to play with it and do a PR for ```

;; TODO - choose better

(defn- choose-coord

[coord1 coord2]

(if coord1

(if coord2

(let [v1 (:version coord1)

v2 (:version coord2)]

(if (pos? (compare (str v1) (str v2)))

coord1

coord2))

coord1)

coord2))

```

U050ECB92 : once you have it installed, you can use it in a project folder (using `\$PWD/deps.edn`) or globally (using `\$HOME/.clojure/deps.edn`)

U050ECB92 : The version comparison seems intentionally naive, no need to poke fun at that yet :smiley:

U11BV7MTK : oh i know. i wasn't doing it in the notion of sniping

U050ECB92 : If you specify versions explicitly you won't invoke that code

U11BV7MTK : but let alex and others figure out the hard stuff and I could help out with the easy solved stuff

U5ZAJ15P0 : I would like to write a function (if possible with the stdlib) to check if a map "matches" a pattern. More specifically, I would like to describe a map in terms of another map with the same structure, whose values can either be ints/bools/strings or regexs. If the value are ints/bools/strings they they should be compared for equality with the target map, and if they are regexs they should match

U11BV7MTK : i meant "take this off their plate" not like "i can't believe they coded this"

U5ZAJ15P0 : e.g. ```

{:first-name "john"

:last-name #"\*."}

```

should match

```

{:first-name "john"

:last-name "doe"}

```

U5ZAJ15P0 : it should work for deeply nested maps

U050ECB92 : i know :smiley: I wonder what lein and boot do

U11BV7MTK : spec?

U5ZAJ15P0 : (I would use it to write tests checking the expected output of an API)

U0NCTKEV8 : there are a few options

U0NCTKEV8 : (for the map thing)

U5ZAJ15P0 : mmh yes specs would work but it seems a bit overkill for what I need

U0NCTKEV8 : you could just take two maps, the first being the shape and the second being the data and recurse

through them

U5ZAJ15P0 : <@U0NCTKEV8> that's precisely what I'm wondering how to do. Is there an easy way to express this recursion procss with stdlib functions?