

U5TBHUM8B : Awesome! Thank you!

U5YHX0TQV : And be careful with slurp'ing in data, you'll have all in memory

U38J3881W : Hey! I was wondering if repeated subvecs would ever allow the excluded data to be garbage collected if there was no way to access it any more? Or would I need to use `(vec (rest x))` if I wanted to _actually_ drop the first item in a vector?

U38J3881W : I would use a persistent queue but I want in place updates too, performance of `(vec (rest x))` being O(n) is a non-issue for me too, looking for the "idiomatic" solution more than anything :slightly_smiling_face:

U06HTKDMF : damn didn't realize `for` re-evals the inner range expression

U06HTKDMF : ``boot.user=> (for [x (do (println "x") [1 2]) y (do (println "y") [1 2])] [x y])x

y

y

([1 1] [1 2] [2 1] [2 2])``

U060FKQPN : it's by design

U060FKQPN : how would `(for [x [[1 2] [3 4]] y x] y)` work otherwise

U06HTKDMF : i was p confused for a bit because i was using `(q/random ...)` (from quil) inside the inner one xD

U06HTKDMF : yea makes sense :open_mouth:

U5XMV6DQT : <@U28947274> or you can make something like that``

(let [coll [1 2 3]

res (concat (map #(vector :test %) coll)

(map #(vector :test2 %) coll))]

(doseq [x res]

(apply println x)))

...

U5XMV6DQT : I prefer not to mix side-effects with pure stuff

U5N8R3NF4 : Someone Please suggest real-time-messaging clojure library. I might also need Screen Sharing and video chat.

U051SS2EU : jgeraert: `doseq` works great for this, but there's also `run!` which like this usage of map takes a function and a collection as args, but is run for side effects eagerly.

U051SS2EU : <@U28947274> meant to tag you above

U28947274 : <@U051SS2EU> Ah I see, thanks for the tip :slightly_smiling_face:

U06HHF230 : Is this expected on clojure 1.9-alpha17?

...

(keyword :foo) ;; :foo

(keyword "anamespace" :foo) ;; ClassCastException

...

U060FKQPN : yes, the 2 arity of keyword only accepts strings

U06HHF230 : I understand that supporting coercions can be problematic but it is a little counter intuitive given the arity 1 behaviour

U61KCTX8S : can anybody tell me how to use pprint? I mean in a file, not the REPL. What is the magic :require :use line i need to use?

U61KCTX8S : i google this shit with no examples

U050SC7SV : `(with-out-str (clojure.pprint/pprint x))`

U050SC7SV : then you can `spit` it in the file

U61KCTX8S : I build an uberjar

U61KCTX8S : does compilation fails because it does not recognize pprint

U61KCTX8S : it is not properly imported