

U3TSNPRT9 : Yes - perhaps I should stop being so purist and get better at using it!

U0567Q30W : It's essential to my work, definitely. Fortunately we don't have to choose, we can have both!

U5NMX95UJ : hey, I have a function `f` that calls `g`, and `g` calls `f`, how to order `defn's` so the compiler knows about all definitions?

U0T4T70KB : Declare

U0T4T70KB : `declare` I mean

U5NMX95UJ : <@U0T4T70KB> that's it, thanks

U6BQSDQRE : Hello. I'm trying to set up a project for both client and server development, using compojure and http-kit. Since I already had a project going, I decided to modify my own `project.clj` to match the the one I would get using `lein new compojure ..`, but I'm getting some errors from ring regarding jetty. Here is my `project.clj`:
<<https://gist.github.com/dea054c015658d4ed3295b40b12c8fd5>>

Here are the errors: <<https://gist.github.com/f3f62cb6633979ebbf849a8b6b20861>>

U6BQSDQRE : This is upon running `lein ring server`

U6BQSDQRE : I also just noticed that `lein deps :tree` seems to output a good deal of warnings regarding my dependencies. I'm honestly not sure what the best way to fix that would be, so I'll post that as well:
<<https://gist.github.com/87f04f1805c2cc20e1dee2a9defe0462>>

U6BQSDQRE : Hmm. This seems to have something to do with datomic. I added exclusions like this:
`[com.datomic/clj-client "0.8.606" :exclusions [org.eclipse.jetty/jetty-http]]` and eventually got it to run. Now the `lein ring` server starts but attempting to connect to it just results in a connection reset

U5ZAJ15P0 : Why does `(if-let [x y] ...)` expands to`
(let*
 [temp__4655__auto__ y]
 (if temp__4655__auto__ (closure.core/let [x temp__4655__auto__] ...) nil))
...`

and not to
...`

(let [x y]
 (if x ...))
...`

?

U06F82LES : so that `x` doesn't have to be re-evaluated multiple times

U06F82LES : Different question: Is this an ok way to access both the first arg and all args? `(fn [& [head :as all]])`

U5ZAJ15P0 : how would x be re-evaluated multiple times in that scenario?

U5ZAJ15P0 : <@U06F82LES> ^

U06F82LES : ``user=> (macroexpand-1 '(if-let [x (do (prn :foo) true)] [x x] (prn :no)))(closure.core/let
[temp__6751__auto__ (do (prn :foo) true)] (if temp__6751__auto__ (closure.core/let [x temp__6751__auto__] [x x]) (prn :no)))``

U06F82LES : i.e. in cases where `x` is used in the body of the if-let

U5ZAJ15P0 : <@U06F82LES> it still wouldn't get evaluated multiple times with``
(let [x (do (prn :foo) true)] (if x [x x]))
...`

U5ZAJ15P0 : ?

U06F82LES : isn't that pretty much what the actual implementation does?

U060FKQPN : <@U5ZAJ15P0> that binds `x` in the else branch

U060FKQPN : `if-let` doesn't

U060FKQPN : that's what the extra local binding is for

U5ZAJ15P0 : <@U060FKQPN> oh, I see. that makes sense; thanks!

U5ZAJ15P0 : I am getting a StackOverflow error when trying to use log4j with Leiningen, could someone please take a look at the stacktrace? <<https://gist.github.com/hmaurer/40c1bc6d08550222f65e06a6366f0eb9>>