

U11BV7MTK : probably pretty easy to inadvertently hit
 U5ZAJ15P0 : yep, I most definitely inadvertently hit that
 U5ZAJ15P0 : If Clojure's reader threw an error saying "unknown character at position X" it would have been easy/easier to debug, but it considered it as part of the symbol
 U0NCTKEV8 : that actually may be just be a bug in the reader
 U5ZAJ15P0 : Now I know how to confuse the hell out of people though
 U5ZAJ15P0 : ``wef-backend.core=> (def 42)
 #wef-backend.core/
 wef-backend.core=>
 42
 wef-backend.core=>
 ...

U5ZAJ15P0 : this works
 U5ZAJ15P0 : this too:
 U5ZAJ15P0 : ``wef-backend.core=> (def hello world this is a long variable 99)
 #wef-backend.core/hello world this is a long variable
 wef-backend.core=>
 ...

U5ZAJ15P0 : eh
 U5JUDH2UE : What is a fast way to do this without the repeated nested pointer?``
 (-> state
 (assoc-in [:a :b :b/field0] ""))
 (assoc-in [:a :b :b/field1] ""))
 (assoc-in [:a :b :b/field2] ""))
 ...

U0CM1QURZ : ``;; maybe?
 (assoc-in [:a :b] {:b/field1 "" :b/field1 "" :b/field2 ""}))
 ...

U050MP39D : fast as in performance?
 U5JUDH2UE : Fast as in, using standard library. I don't want to add any external libs or anything.
 U5JUDH2UE : Performance isn't a concern.
 U5JUDH2UE : Sorry, I shouldn't be using that work. :stuck_out_tongue:
 U050MP39D : (merge state {:b/field1 "" :b/field1 "" :b/field2 ""}))
 U050MP39D : err
 U050MP39D : sorry, (update-in state [:a b] merge {:b/field1 "" :b/field1 "" :b/field2 ""}))
 U050MP39D : forgot a few words
 U11BV7MTK : `` (let [nav [:a :b]]
 (-> { :a { :b {} } }
 (assoc-in (conj nav :b/field0) ""))
 (assoc-in (conj nav :b/field1) "")))
 { :a { :b { :b/field0 "", :b/field1 "" } } }
 ...