

U236M9FH9 : `Array.indexedMap`, but switch `val i ->` to `i val ->`
 U2SR9DL7Q : Newbie Question of the day : If I get a package from elm-package, and it's not compatible with my version of elm, or I decide I no longer need it, do I just just remove it from the package json?
 U236M9FH9 : yup
 U2SR9DL7Q : <@U236M9FH9> cool. Also, if I'm having trouble with a package, I've seen some answer suggesting deleting the elm stuff folder. How would I rebuild said folder after?
 U236M9FH9 : It gets generated when you run elm-make
 U5XC2FJ1Y : actually i'm not sure if i can even write any function `List number -> Float` except by implementing a constant function `const : a -> Float` and using letbindings to cheat the type inferer
 U2SR9DL7Q : <@U236M9FH9> double cool. Thanks thanks
 U6D3ERLA1 : There's no ternary operator in elm?
 U0LPMPL2U : `if condition then value else otherValue`
 U5XC2FJ1Y : `>` doesn't take a polymorphic `number`
 U0LPMPL2U : same as the multiline form but without whitespace``
 if condition then
 value
 else
 otherValue
 ``

U6D3ERLA1 : I'm using the elm vim formatter which is pretty ruthless in its enforcement of white space... a L O T of whitespace
 U6D3ERLA1 : O n c e y o u s t a r t r e a d i n g i t t h o y o u s t a r t t o w o n d e r w h y e v e r y t h i n g e l s e i s s o c r o w d e d
 U6D3ERLA1 : :white_large_square:
 U0LPMPL2U : haha yes. I've found it also pushes me to write many small functions instead of nesting things deeply
 U6D3ERLA1 : true, it doesn't really work with long functions
 U6D3ERLA1 : definitely makes sense with the terse / dense idiom
 U6D3ERLA1 : I feel like I'm getting closer and closer to a form of pure notation as my career goes on
 U6D3ERLA1 : `` Array.indexedMap
 (y row ->
 Array.indexedMap
 (x val ->
 if hasCoord (getCoords { x = x, y = y } l d k grid) then
 key
 else
 val
)
 row
)
 grid
 ``

U6D3ERLA1 : this is a little ridic tho
 U2SR9DL7Q : Has anyone experience a situation where the elm package installer is successfully downloading elm packages, but the compiler/linter can't find the imported packages in elm-stuff? (But you can see them there)
 U0LPMPL2U : I feel like there's some way to use a `map2` function to make that nicer
 U14Q8S4EM : Yeah but, I would kinda doubt my own sanity first. Have you tried deleting `elm-stuff`, and re-installing?
 U14Q8S4EM : What package are you trying to use <@U2SR9DL7Q>, and what is the error?
 U2SR9DL7Q : <@U14Q8S4EM> <<http://package.elm-lang.org/packages/wernerdegroot/listzipper/1.0.2/List-Zipper>>
 And the error just says that it can't import import it. But the dependency is defined in elm-package.json
 U2SR9DL7Q : The same thing happened when I tried another package recently. And when I tried just copy pasting the relevant elm files in my main project folder, it was recognizing most of the exposed functions, but not a key exposed type. It's all very strange. Contemplating rebuilding elm-stuff.
 U14Q8S4EM : Do you have your code online? I could try building it.
 U2SR9DL7Q : <<https://github.com/tkshill/tkshill.github.io>>