

U5AEH3L05 : But generically, for any fn that takes 2 values?

U5AEH3L05 : I wrote my own little function for it but I've run into it enough times that I'm wondering if there's already a functional way to do this that I'm missing

U0J1M0F32 : List-Extra has a lot of list helper functions that are sort of a community agreed upon idiomatic way of building out these functions. You may also find something to simplify the function you wrote.

<<http://package.elm-lang.org/packages/elm-community/list-extra/6.1.0/List-Extra>>

U5AEH3L05 : aha, looks like I wanted scanr!

U0J1M0F32 : :thumbsup:

U5AEH3L05 : Wait, nope. I'll keep reading. Thanks justin!

U48AEBJQ3 : <@U5AEH3L05> I think this is the right search, but I'm not seeing anything:

<[https://klaftertief.github.io/elm-search/?q=\(a+-%3E+a+-%3E+b\)+-%3E+List+a+-%3E+List+b](https://klaftertief.github.io/elm-search/?q=(a+-%3E+a+-%3E+b)+-%3E+List+a+-%3E+List+b>)>

U5AEH3L05 : <@U48AEBJQ3> Yeah, I just tried that as well and didn't see anything

U0LPMPL2U : You could use `foldl`. Any fancier function would probably be implemented in terms of `foldl`

U5AEH3L05 : Got it. Okay, sounds like the version i have is as idiomatic as it gets!

U5AEH3L05 : For reference, that looks like: ```

pairwiseMap : (a -> b) -> List a -> List b

pairwiseMap fn list =

case list of

a :: b :: xs ->

(fn a b) :: (pairwiseMap fn (b :: xs))

_ ->

[]

...

U0J1M0F32 : I like it.

U5AEH3L05 : I guess what I'm wondering is -- am I thinking about this wrong if I keep running into this?!

U5AEH3L05 : <<https://media.giphy.com/media/NPyHgTkMStCXC/giphy.gif>>

U48AEBJQ3 : theotherben:```

mapAdjacent : (a -> b) -> List a -> List b

mapAdjacent f xs =

xs

|> List.foldr

(\x (next, acc) ->

case next of

Nothing ->

(Just x, acc)

Just y ->

(Just x, f y x :: acc)

)

(Nothing, [])

|> Tuple.second

...

U48AEBJQ3 : Just as an alternative. I don't think it is really better.

U0J1M0F32 : Depends on what it is you're trying to do.

U0J1M0F32 : In my experience, from being here in <#C192T0Q1E|beginners> for a long time, a lot of people run into issues when they aren't leveraging the type system fully.

U0J1M0F32 : Or their datastructures could be remapped.

U48AEBJQ3 : <@U5AEH3L05> By 'this' do you mean wanting to use functions which aren't already part of some standard package? I don't think that should necessarily discourage you. Elm's ethos is to be cautious about deciding which features to add in an effort to encourage exploration and experimentation to see if we can find good solutions without extra complexity.

U23SA861Y : a pairwise map is a perfectly reasonable method of traversing a list, however it is preferable to implement it via the base fold method as opposed to recursion

U5AEH3L05 : Ahh, thanks for showing me how to do it with foldr! I was thinking about that.

U5AEH3L05 : <@U23SA861Y> why's that?

U5AEH3L05 : <@U48AEBJQ3> Good to know. I guess it felt like I saw every other kind of map but this :slightly_smiling_face: