U300LJUAK : Gotta admit that's a feature I would love too.
U5XC2FJ1Y : what's the best alternative, just moving the conditional inside the matched pattern?
U300LJUAK : Yup. Although it can lead to duplicated code in your `else` case, that's basically the only way to go right now.
U2FP79HN3 : How do recursive types work? Say I have a cell which can be linked to other cells..
```
type alias Cell =
    { row : Int
    , column : Int
    , links : List Cell
    }
```

doesn't work, so I tried

```
type alias Cell =
    { row : Int
    , column : Int
    , links : Links
    }


type Links
    = List Cell
```

which didn't really work with:

```
link : Cell -&gt; Cell -&gt; Cell
link cell neighbour =
    { cell | links = cell.links :: neighbour }
```

and then I tried

```
type Links
    = Links (List Cell)
```

But now I'm in type un/wrapping hell

U0JFXEUCT : I believe you want something like `type Cell = Cell {}`
U0JFXEUCT : instead of a type alias
U0JFXEUCT : There is still some unwrapping, but remember you can unwrap in the function arguments
U0JFXEUCT : something like `link (Cell cell) = --do stuff`
U0CLDU8UB : The compiler suggests something like that to you when you make a recursive type alias! :slightly_smiling_face:
U2FP79HN3 : Yeah, I've read <https://github.com/elm-lang/elm-compiler/blob/0.18.0/hints/recursive-alias.md> but still confused
U0CLDU8UB : Okay, so reiterating what Matt said, you can do this:```
type Cell =
    Cell
    { row : Int
    , column : Int
    , links : List Cell
    }
```
```

U0CLDU8UB : and then something like```
link : Cell -> Cell -> Cell
link (Cell cell) neighbour =
  Cell { cell | links = cell.links :: neighbour }
```

U0LPMPL2U : If you find yourself unwrapping, doing something with the data, and re-wrapping a lot, I find it helpful to define a `map` function
U2FP79HN3 : I have tried something like that <@U0CLDU8UB>, but then I get
<http://cloud.jorisooms.be/2I1v291t0y2y>
U2FP79HN3 : which confuses me even more tbh, because I don't know why it's a List (List Cell)
U2FP79HN3 : or should be*
U2FP79HN3 : this is the neighbour it's complaining about
U0CLDU8UB : Can you put your code on <http://ellie-app.com|ellie-app.com> ?
U2FP79HN3 : sure
U0CLDU8UB : Great
U2FP79HN3 : <https://ellie-app.com/3CdkPHcHsLYa1/0>
```