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>
U0JFXEUCT : ah, flip the args.. `neighbor :: cell.links`
U2FP79HN3 : lol
U2FP79HN3 : -_-
U2FP79HN3 : thanks
U0CLDU8UB : Good catch, I missed that too!
U2FP79HN3 : Guess I can get started now, thanks a lot :wink:
U5VTA57UN : is there a nice way to make this work: ```
[y1s, y2s] = List.map toString <|
case textLoc of
Above -> [-3, 10]
Below -> [0, 13]
```

U5VTA57UN : If I use `(y1s, y2s)` then I can't map toString over all the values, and have to type toString more than once.
U48AEBJQ3 : You should use tuples. You might want to define `(a -&amp;gt; b) -&amp;gt; (a, a) -&amp;gt; (b, b)` if you want it to be cleaner.
U5VTA57UN : So make a tupleMap2 utility function?
U48AEBJQ3 : Yep.
U48AEBJQ3 : It is also defined in a bunch of packages already, but not something you are likely already using.
U48AEBJQ3 : <https://klaftertief.github.io/elm-search/?q=(a+-%3E+b)+-%3E+(a%2C+a)+-%3E+(b%2C+b)>
U5VTA57UN : Thanks. Toolkit is a useful pointer.
U611WQPL4 : Can I ask a <#C192T0Q1E|beginners> question? There's something I don't follow. In type definitions, what's the difference between `(String -&amp;gt; Int) -&amp;gt; Int` and `String -&amp;gt; Int -&amp;gt; Int`. What is `(String -&amp;gt; Int)` denoting here? I understand that `String -&amp;gt; Int -&amp;gt; Int` is a function that takes a `String` and `Int` parameter and returns an `Int` expressed in a curried format. But what is `(String -&amp;gt; Int)` expressing?
U0LPMPL2U : `(String -&amp;gt; Int)` is saying that one of the arguments is a function `String -&amp;gt; Int`
U0LPMPL2U : Yes, Elm allows you to pass functions as arguments to other functions :smile:
U611WQPL4 : Ah, that's what I thought it was. Awesome. Thanks for the tip. It's ~turtles~ functions all the way down.
U0LPMPL2U : it's common to see more generic versions like: `map : (a -&amp;gt; b) -&amp;gt; List a -&amp;gt; List b`
U0LPMPL2U : the first argument to `List.map` is a function whose input is whatever type of your input list is and it's output is whatever you want the type of your output list to be
U0CLDU8UB : By the way, maybe you had realized this already but `String -&amp;gt; Int -&amp;gt; Int` could also be written as `String -&amp;gt; (Int -&amp;gt; Int)`, which is sort of a "more literal" representation of the currying.
U17PWHU4D : Hi all, can someone point me towards documentation on how to type unknown keys? example as in normalized data, where the keys are unknown strings. but the data contained is known?
U601ELFEG : My page has a bunch of "static-ish" content: help panels, a carefully laid out control button area.... I need to interact with these static areas from elm (decide which help panels should be shown based on the context of the model, act on those control buttons).... .... but I really don't want to author that static ish content in elm code
U601ELFEG : what's the best way to go about that?
U0LPMPL2U : <@U17PWHU4D> are you asking about `Dict`?
U0LPMPL2U : string keys pointing to values of a known data type?
U3SJEDR96 : <@U17PWHU4D> you could do that with a Dict, where the keys are those strings and the values... If they're all the same type, that would work just like that, if not you'd have to make a union type to represent the different possibilities
U17PWHU4D : thanks <@U0LPMPL2U> and <@U3SJEDR96> , going to look into `Dict`. and yes. example:
```
{
"1234": {
id: 1
},
"1235": {
id: 2
}
}
```
```

```

U17PWHU4D : assuming that's the way to do it.  thanks again!
U0LPMPL2U : Is this JSON you're trying to convert to an Elm structure?
U0CLDU8UB : <@U601ELFEG> What do you mean by "don't want to author [..] in elm"? Do you mean that you don't want to write that in Elm or that you want to keep it in your codebase as plain HTML?
U601ELFEG : meaning I want to keep things like the content of the help panels authored in HTML - probably in a file apart from the elm code -
U17PWHU4D : in the example there <@U0LPMPL2U>, but i also have the same issue for types.  so i think Dict should get me on the right path.
U601ELFEG : ditto the control panel (it has some meticulously laid out and styled buttons)
U0LPMPL2U : The structure of your types doesn't have to mirror the structure of your JSON
U0LPMPL2U : for example, you could have:
```
type alias Thing =
  { number : String
  , id : Int
  }
```

U0LPMPL2U : and convert the JSON into this type
```