U4RR7KX45 : these types are such a pain sometimes :disappointed: can anyone shed some light, why am I getting an error here?```
module RNATranscription exposing (..)

import Dict exposing (..)

mapping : Dict String String
mapping =
    [("G", "C")
    , ("C", "G")
    , ("T", "A")
    , ("A", "U")
    ]
        |> fromList

getMapping : Char -> Result String String -> Result Char String
getMapping x acc =
    case acc of
        Ok values ->
            case get x mapping of
                Just something ->
                    Ok (String.append values (String.fromChar something))

                Nothing ->
                    Err x

        _ ->
            acc

toRNA : String -> Result String String
toRNA str =
    String.foldl getMapping Ok ""
```

and the error:
```
-- TYPE MISMATCH ------------------------------------ ./../RNATranscription.elm

The 2nd argument to function `get` is causing a mismatch.

20| get x mapping
                        ^^^^^^^
Function `get` is expecting the 2nd argument to be:

    Dict Char v

But it is:

    Dict String String

Hint: I always figure out the type of arguments from left to right. If an
argument is acceptable when I check it, I assume it is "correct" in subsequent
checks. So the problem may actually be in how previous arguments interact with
the 2nd.
```

U4RR7KX45 : it doesn't work even when I change the type definition to ```
```

Dict Char v
```

U48AEBJQ3 : <@U4RR7KX45> `get (String.fromChar x) mapping` ?
U3SJEDR96 : (Yay exercism)
U4RR7KX45 : :smile: yeah. well that worked for that particular one
U4RR7KX45 : but now I have 2 more mismatches
U4RR7KX45 : is there a tool that automatically fixes these? :smile: haha
U4RR7KX45 : takes me half day sometimes
U3SJEDR96 : You'll get (much) better at it as you progress. Eventually, you'll start thinking about the types first, and then finding a function that matches those types :slightly_smiling_face:
U4RR7KX45 : hope so :smile:
U4RR7KX45 : how can I convert a string to character in dictionary key?like here
```
mapping : Dict String String
mapping =
    [("G", "C")
    , ("C", "G")
    , ("T", "A")
    , ("A", "U")
    ]
        |> fromList
```

instead to have
```
mapping : Dict Char String
mapping =
    [("G", "C")
    , ("C", "G")
    , ("T", "A")
    , ("A", "U")
    ]
        |> fromList
```

U153UK3FA : <@U4RR7KX45> read the compiler output carefully and think about why it thinks the types are that way. Like, why does the compiler think it should be a `Dict Char v`?
U153UK3FA : <@U4RR7KX45> Char literals have single quotes eg `'G'`
U4RR7KX45 : hmm
U4RR7KX45 : didn't know, thank you :wink:
U4RR7KX45 : that makes sense now haha
U4RR7KX45 : that's it, works yaayy :smile:
U4RR7KX45 : thank you all
U6CHZ5PN1 : Hello, i'm new to elm and tried to follow <https://www.elm-tutorial.org/en/04-starting/07-multiple-modules.html>, but instead simply make different files to every modules, i also put them in separate folder
U6CHZ5PN1 : but then the files inside the folder somehow doesnt compiled
U6CHZ5PN1 : <https://github.com/iamn00b/elm-tutorial>
U6CHZ5PN1 : any ideas why?
U3LUC6SNS : <@U6CHZ5PN1>, If you have module `Bar` in file `Bar.elm` of folder`Foo`, the module name should actually be `Foo.Bar`
U3LUC6SNS : <@U6CHZ5PN1>  sorry about the typing errors
U6CHZ5PN1 : i've tried to change the name to `Type` or `Type.Type` but same error
U6CHZ5PN1 : it said that Main can't find the module
U6CHZ5PN1 : but if i change the source-directories to `['src', 'src/type']` it compiled
U4RR7KX45 : I've got a simple yeoman elm generator btw, if anyone wants to use it<https://github.com/Bravilogy/generator-elm>

U4RR7KX45 : webpack + elm + sass
```

U3SJEDR96 : <@U6CHZ5PN1> try naming your folder `Type` rather than `type`
U6CHZ5PN1 : <@U3SJEDR96> ah it works if i also name it to `Type.Type`
U3SJEDR96 : There you go :slightly_smiling_face:
U6CHZ5PN1 : is in elm i should use capital for folder, or is this related to webpack/elm-webpack-loader?
U6CHZ5PN1 : also thanks! :slightly_smiling_face:
U3SJEDR96 : Elm wants a 1-1 mapping of path <-> module
U4RR7KX45 : do I have to use Random.generate for Random module? Can I not just get a random value? Or is Msg needed because Random is impure?
U6CHZ5PN1 : is there a way to make module `Type` but it live on `Type` folder? like how index.js works
U4RR7KX45 : for example```
(toString (<http://Random.int|Random.int> 1 10))
```

U3SJEDR96 : <@U6CHZ5PN1> no. You can name you file `Type.elm`, though - having a structure where you have a file named `Foo.elm` and a folder named `Foo` which contains "namedspaced" modules is a fairly common pattern
U3SJEDR96 : <@U4RR7KX45> correct, it's an impure thing. Though you can also use `Random.step`, provided you have a random seed and store the new seed after use :slightly_smiling_face:
U6CHZ5PN1 : <@U3SJEDR96> i see. thanks!
U6BTZ2NSV : Hello. I am trying to understand Tasks. I am confused by the example for `perform` at <http://package.elm-lang.org/packages/elm-lang/core/latest/Task> `Task.perform NewTime Time.now` NewTime is a message, how does it represent `(a -&gt; msg)`? Why do you not need to use `(\_ -&gt; NewTime)`?
U3SJEDR96 : `type Msg = NewTime Time`
U3SJEDR96 : `NewTime` is a tag of `Msg`, and takes a single parameter, `Time`. The type of `NewTime` is `Time -&gt; Msg`
U3SJEDR96 : It's the same as `type Msg = Input String` <-> `input [ onInput Input ] []`
U6BTZ2NSV : <@U3SJEDR96> I see now. Thank you. If NewTime didn't take an extra parameter, you would need to use the lambda in order to type check, right?
U3SJEDR96 : Yep, correct
```