U23SA861Y : because it returns a `Task x Time` the x indicates that it is a type variable and this implies that it can never fail
U62JFJWKT : ah
U62JFJWKT : ok when my brain will be more rested I'll take a closer look at that
U62JFJWKT : but you probable saved me a lot of time
U62JFJWKT : probably
U23SA861Y : that's what this channel is for :stuck_out_tongue:
U20K0C2G4 : is there a good way to update one value of a tuple?
U23SA861Y : how big is the tuple?
U20K0C2G4 : 2
U20K0C2G4 : let me clarify
U20K0C2G4 : ```   newModel =   { model
                | landingProblemSolutionModels = (transferRecord.problemSolutionModel, second model.landingProblemSolutionModels)
            }```

U20K0C2G4 : I have a value that I would like the first part of the tuple to be updated to
U23SA861Y : I think some of the issues here is in the verbosity of the text, what you have written there is not particularly complex
U20K0C2G4 : <@U23SA861Y> thanks for the response, I'm gonna play around with this
U23SA861Y : maybe you could do:```
Tuple.mapFirst (_->transferRecord.problemSolutionModel) model.landingProblemSolutionModels
```

U23SA861Y : but that isn't much more compact
U23SA861Y : the other thing you can do is pull the tuple creation up into a let block
U20K0C2G4 : yea I guess I was hoping there was a similar syntax for updating tuples as there is for records
U20K0C2G4 : a syntax where I don't have to supplying values for the fields I'm not updating
U23SA861Y : <@U20K0C2G4> to be fair, depending on how you are using this you might want a record instead of a tuple. or perhaps even two top level entries?
U64MK7215 : hey
U64MK7215 : how to use multiple divs in elm?
U153UK3FA : <@U64MK7215> wrap them in a div
U153UK3FA : eg. `div [] [div [] [], div [] []]`