U3KSN5MAL : it did
U48AEBJQ3 : You shouldn't need a helper function here and you should use `succeed` and `fail`.
U48AEBJQ3 : Don't use a `Result` for anything.
U3KSN5MAL : oh ok
U3KSN5MAL : aaah
U3KSN5MAL : ok got it
U48AEBJQ3 : woot!
U5GAFT2AW : how do I initialize `Date` with a default value? if I have something like this below:```
-- MODEL
type alias Model =
  { date : Date
  , time : String
  , entryId : String
  }

init : (Model, Cmd Msg)
init =
  (Model &lt;what goes here?&gt; "" "", Cmd.none)
```

U153UK3FA : <@U5GAFT2AW> what do you consider a default date?
U153UK3FA : perhaps you want a `Maybe Date`
U153UK3FA : ```-- MODEL
type alias Model =
  { date : Maybe Date
  , time : String
  , entryId : String
  }

init : (Model, Cmd Msg)
init =
  (Model Nothing "" "", Cmd.none)
```

U153UK3FA : is `""` a valid usable value for an `entryId`?
U153UK3FA : you might want to wrap the whole thing in a `Maybe`
U5GAFT2AW : Ohhh, I definitely don't understand `Maybe`
U3HQVHERX : `Maybe` is `Nothing` or `Just something` and you can pattern match it to get your value
U3HQVHERX : maybe date's default value would be `Nothing`
U3HQVHERX : and if it gets updated, it would be `Just date`
U3HQVHERX : then if you see `date` you can do `case date of Nothing -&gt; ... Just d -&gt; ...`
U5GAFT2AW : ahhh, because it is indeterminate at run time?
U3HQVHERX : um, because we don't like `null`s
U3HQVHERX : <https://guide.elm-lang.org/error_handling/maybe.html>
U3HQVHERX : &gt; A Maybe can help you with optional arguments, error handling, and records with optional fields.
U3HQVHERX : <http://package.elm-lang.org/packages/elm-lang/core/latest/Maybe>
U153UK3FA : It's a lot like `null` in other languages but the compiler forces you to check it
U5GAFT2AW : got it, thank you <@U153UK3FA> and <@U3HQVHERX> , definitely going to read up on `Maybe`
U3HQVHERX : it's not too scary
U3HQVHERX : just takes some getting used to
U5YDC1YUW : is there a "join" type method in Elm? Say taking a list `["a", "b", "c"]` and joining with `"-"` giving you the string "a-b-c"?
U5YDC1YUW : or would you have to just build that yourself using `intersperse` and `foldl`?
U48AEBJQ3 : <@U5YDC1YUW> <https://klaftertief.github.io/elm-search/?q=String+-%3E+List+String+-%3E+String>
U5YDC1YUW : magic!
U3KSN5MAL : thanks for helping with the exercizes, got to 11, but that covers what i need atm so i'm all good for now
U3SJEDR96 : Yeah, that's my main worry with this format. I tried to only introduce a single concept with every exercise, so at least you don't need to pick up multiple things at the same time. I'm definitely going to try and use your feedback to make it better tho
U61JQ4F4J : Hi! can I match against several options in a `case foo of`? I mean being able to have a pattern such as

`Foo || Bla -&gt;`...
U3SJEDR96 : Oh, and once you feel like you have a decent solution to things, there's a solutions branch which can compare to :grin:
U48AEBJQ3 : <@U61JQ4F4J> No, the pattern matching is fairly simple. You *can* just shunt the code into a function and call it from both branches.
U61JQ4F4J : yeah that's the other option I was thinking... thanks <@U48AEBJQ3>
U3SJEDR96 : I'm a little torn, tho. Do you feel it would've been helpful if I'd first asked, for example, to make a decoder which would always fails with a certain message? (Which would mean "adapt the `fail` that's already there, but might help in identifying how you'd need the opposite of that for a decoder which uses a hardcoded value)
U3SJEDR96 : Similarly, for grokking `map`, it might be useful to point introduce `List.map` or `Maybe.map` first - but do you feel like that would have helped, if it weren't for someone walking you through those in here?
U3SJEDR96 : Hm, right. Good point.
U3KSN5MAL : ```arrayAsTuple3 : Decoder a -&gt; Decoder b -&gt; Decoder c -&gt; Decoder (a, b, c)arrayAsTuple3 a b c =
    Json.Decode.map3 (,,) a b c

decodeConvertedModel : Json.Decode.Decoder ConvertedModel
decodeConvertedModel =
    Json.Decode.Pipeline.decode ConvertedModel
        |&gt; Json.Decode.Pipeline.required "pickerMode" (<http://Json.Decode.int|Json.Decode.int>)
        |&gt; Json.Decode.Pipeline.required "activeColor" (Json.Decode.array arrayAsTuple3)```

U3KSN5MAL : What am i doing wrong here :confused:?
U3KSN5MAL : can't work out how to feed the information correctly into arrayAsTuple
U3SJEDR96 : You can just use `arrayAsTuple3 (index 0 int) (...) (..)` there.
U3KSN5MAL : ah
U3KSN5MAL : perfect
U3SJEDR96 : Of you can change the tupledecoder so that it will do the indexing by itself (by "wrapping" the decoders with the relevant `index x`), so you could just call `arrastup3 int int string` or whatever :)
U3KSN5MAL : ah ok
U3KSN5MAL : thank you!
U3SJEDR96 : (on mobile so being a little terse, sorry ;))