

U3KSN5MAL : lol ok  
 U48AEBJQ3 : The next exercise will have you use it for something more useful.  
 U3KSN5MAL : I'm still not convinced of the format of this sort of exercise when there is no place to really go but to ask people if you get stuck.  
 U3KSN5MAL : I tried exercise for a while and kept running into the same thing  
 U3KSN5MAL : am i supposed to be using the decodeString on it?  
 U3KSN5MAL : nvm got it  
 U3KSN5MAL : supposed to use succeed  
 U3HGVHERX : redroyal: how often do you use the `elm-repl`? i found myself slowly understanding things after doing small things and guessing at the output.  
 U3KSN5MAL : It's not a case of that, it's more lack of not knowing the tools they want me to use  
 U3KSN5MAL : the hell, my code is compiling with elm-make but it's failing when compiling with elm-test  
 U3KSN5MAL : ```type Color = Green  
                   | Blue  
                   | Red

```
decoder : String -> Decoder (Result String Color)
decoder colorString =
  succeed &lt;| fromStringColor colorString
```

```
fromStringColor : String -> Result String Color
fromStringColor string =
  case string of
    "blue" ->
      Result.Ok Blue

    "green" ->
      Result.Ok Green

    "red" ->
      Result.Ok Red

    _ ->
      Result.Err ("I don't know a color named " ++ (toString string))
  ...
```

U3KSN5MAL : the code in question  
 U48AEBJQ3 : What is elm-test saying?  
 U3KSN5MAL : `Compilation failed while attempting to build  
 C:\Users\muzzo\Documents\elm-demystify-decoders\tests\Exercise08\Tests.elm`  
 U48AEBJQ3 : Oh, you changed the type of `decoder`  
 U48AEBJQ3 : It should be `String -> Decoder Color`. I think the talk about `Result` caused some confusion.  
 U3KSN5MAL : yes  
 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`

U3HQQVHERX : `Maybe` is `Nothing` or `Just something` and you can pattern match it to get your value

U3HQQVHERX : maybe date's default value would be `Nothing`

U3HQQVHERX : and if it gets updated, it would be `Just date`