U0CLDU8UB : You'll have to decide if you want that to be a success or a failure
U5D4VHEN7 : Thanks for the reply <@U0CLDU8UB>. I apologize, because I have actually just left that part out. I'll edit it
U0CLDU8UB : Okay. That to me seems like it should work. What is your error message?
U5D4VHEN7 : And from this I get:```
I ran into something unexpected when parsing your code!

206| Decode.string `andThen` decodeDayGrade
                      ^
I am looking for one of the following things:

    end of input
    whitespace

Detected errors in 1 module.
```

U0CLDU8UB : Oh right, the backtick syntax of course
U0CLDU8UB : That was removed in 0.18
U5D4VHEN7 : ahhhhhh, got it. Let me try it without. Is there an alternative, or must I just place the args in order?
U0CLDU8UB : So now it's```
dayStatusDecoder : Decoder DayStatus
dayStatusDecoder =
  Decode.string
    |&gt; andThen decodeDayStatus
```

U5D4VHEN7 : !!! Works! Thanks so much <@U0CLDU8UB> :slightly_smiling_face:
U0CLDU8UB : Awesome! Glad I could help!
U5D4VHEN7 : Actually, was still doing something wrong, but fixed it. :slightly_smiling_face: Here is the final solution (with poor naming prior to refactoring):```
dayBreakdownDecoder : Decoder DayBreakdown
dayBreakdownDecoder =
  decode DayBreakdown
  |&gt; required "grade" dayGradeDecoder
  |&gt; required "status" dayStatusDecoder

dayStatusDecoder : Decoder DayStatus
dayStatusDecoder =
  Decode.string
  |&gt; Decode.andThen doDecodeDayStatus

doDecodeDayStatus : String -&gt; Decoder DayStatus
doDecodeDayStatus dayStatus = Decode.succeed (decodeDayStatus dayStatus)

decodeDayStatus : String -&gt; DayStatus
decodeDayStatus dayStatus =
  case dayStatus of
    "past" -&gt; Past
    "present" -&gt; Present
    "future" -&gt; Future
    _ -&gt; NoStatus

```

U6303RTK7 : strange issue
U6303RTK7 : I'm seeing this error: ```duration : Span -&gt; Int
```

duration span = span.duration
```
```
`span` does not have a field named `duration`
```

U6303RTK7 : ```type Span
  = Span { id : Int, duration : Int }

```

U5D4VHEN7 : I could be wrong, but I think you need to use a type alias instead?```
type alias Span =
  { id: Int, duration : Int }
```

U5D4VHEN7 : I very well could be wrong
U6303RTK7 : that seems to have resolved the issue, thanks :slightly_smiling_face:
U5D4VHEN7 : :slightly_smiling_face:
U153UK3FA : <@U6303RTK7> in your above code you defined  a new type called `Span` with a constructor also called `Span` that takes a record as a parameter.
U153UK3FA : You would construct a value of that type by writing `Span {id = 5, duration= 5}`
U635238TG : i'm doing the 1st exercise in the tutorial, adding a reset button. why did I have to say `Reset -&gt; 0` instead of `Reset -&gt; model = 0`
U153UK3FA : <@U635238TG> the return value of the `update` function becomes the new value of the model
```