U3KSN5MAL : ouch
U3KSN5MAL : thanks
U0EUHKVGB : I've edited it for you. You can click the "edit" button to see what I changed
U3KSN5MAL : ok that's better
U3KSN5MAL : whoops i overwrote it i think with the same thing
U0EUHKVGB : perfect :slightly_smiling_face:
U3KSN5MAL : I'll make issues if i find anything else
U3KSN5MAL : Made one for nested records too. because the rest of your issues people wrote in there seem super unclear...
U3KSN5MAL : Oh just saw the comment
U3KSN5MAL : <@U0EUHKVGB> what do you mean by anonymous record?
my actual record has extra data in it, i just made a minimum test case.

U0EUHKVGB : ```type alias Model = { model : { anon: String } }
```

U0EUHKVGB : `{ anon: String }` is anonymous, i.e it has no name
U0EUHKVGB : It's still a useful bug report
U0EUHKVGB : But generally you want to avoid anon types like that :slightly_smiling_face:
U3KSN5MAL : Why would i want to avoid that? i don't understand
U0EUHKVGB : Because it makes it harder to explain what that piece of your program is.
U3KSN5MAL : `model.sliderOptions.rgbo`
U0EUHKVGB : Imagine you want to make a function that only works on that piece of the data
U3KSN5MAL : what about this is hard to access?
U0EUHKVGB : It's not hard to access - it's hard to understand.
U3KSN5MAL : oh do you mean the name is bad?
U0EUHKVGB : Imagine you want to have a function that takes `rgbo` and returns a new `rgbo`
U0EUHKVGB : Well, the name might be bad, but that's not my point
U3KSN5MAL : hmm, these are just flags
U3KSN5MAL : not data that will be transformed
U0EUHKVGB : ```type alias ConvertedModel =
    { sliderOptions :
        { rgbo : Bool
        , rgbl : Bool
        , hsvo : Bool
        , hsvl : Bool
        }
    }
```

U0EUHKVGB : If you want to do _anything_ with `sliderOptions`, then you have to write the full type out again.
U0EUHKVGB : Error messages are worse as a result.
U0EUHKVGB : So, for example:
U0EUHKVGB : ```changeSlideOption : Bool -&gt; { rgbo : Bool, rgbl : Bool, hsvo : Bool , hsvl : Bool } -&gt; { rgbo : Bool, rgbl : Bool, hsvo : Bool , hsvl : Bool }
```

U0EUHKVGB : this is unreadable.
U0EUHKVGB : ```type alias SliderOptions =
  { rgbo : Bool
        , rgbl : Bool
        , hsvo : Bool
        , hsvl : Bool
        }

changeSlideOption : Bool -&gt; SliderOptions -&gt; SliderOptions
```
this is easy to understand.

U3KSN5MAL : Ok, i getcha. For my specific usecase with these, it worked out to be fine

U3KSN5MAL : I only did it like this because they are only flags that are only ever directly accessed in update
U0EUHKVGB : Decoding in Elm for records is generally reliant on having a named type aliases. If you want to write a decoder easily for `SliderOptions`, you would need to create a type alias so that it generates the `SliderOptions` constructor.