

U0JFGGZS6 : `if (not x) then .. else ..` ?
U0LPMPL2U : Elm forces you to handle all cases. Because of this, you always need an `else` clause
U0LPMPL2U : `unless` `then` `else` would be the same as `if` `then` `else` but just flipping the clauses
U0LPMPL2U : `unless` / `else` is harder to reason about because it's expressed negatively
U0LPMPL2U : <@U5Y1YQD6Y> I'm guessing you may be coming from Ruby which has `unless`? It's convenient in Ruby when you aren't handling the `else` case e.g.``
return "foo" unless condition?

OR

```
unless condition?  
  do_thing  
end  
...
```

When you add `else` it becomes less useful. `unless` / `else` is generally frowned upon and better expressed positively as `if` / `else`

U5Y1YQD6Y : Right you are, <@U0LPMPL2U>
U5Y1YQD6Y : (I'm coming from Ruby)
U0LPMPL2U : Same for me :slightly_smiling_face:
U0FP80EKB : me three!
U23SA861Y : I kinda wish if didn't follow the if else then paradigm like it was implemented more like a regular function
`if : Bool -> a -> a -> a`
U5P4FLYLE : Hi, how would you extract values like below:``input = [[1,2,3] , [4,5,6] , [7,8,9],...]
output=[[1,4,7,...], [2,5,8,...], [3,6,9,...]]``
and in such a way that it is prepared to support extracting input array that can have arrays of different lengths?
``input = [[1,2], [3,4], [5,6],...]
output = [[1,3,5,...], [2,4,6,...]]``

U48AEBJQ3 : <@U5P4FLYLE> Does this meet what you are hoping for?
<<http://package.elm-lang.org/packages/elm-community/list-extra/6.1.0/List-Extra#transpose>>
U5P4FLYLE : yes, this is what I am looking for - thanks
U5J08KX0D : Hi. I'm looking for a simple example of responding to an onMouseOver message over a named SVG element. I'm confused as to what message I will get in updates and whether or not I need to listen to the DOM. Say I have an element
`circle [name "head", cx canvas_center, cy "100", r "65", fill "#FFCD94"] []`

U5J08KX0D : how would I get a mouse over message for that element?
As ever, grateful for the help!!

U153UK3FA : <@U5J08KX0D> it works just like the Html package,
<<http://package.elm-lang.org/packages/elm-lang/svg/2.0.0/Svg-Events#onMouseOver>>
U5J08KX0D : OK, thanks <@U153UK3FA> - exactly the advice I needed and got me underway!
U5J08KX0D : I now get a "Click" msg onClick... how can I get the name of the object clicked on. onClick only expects one argument, and I'd like to say [onClick Click "head",...]
U5J08KX0D : got it: [onClick (Click "some string"), ...] andMessage = Click String

U236M9FH9 : `onClick (Click "head")` or `onClick <| Click "head"`
U5J08KX0D : is that correct?
U5J08KX0D : Great <@U236M9FH9> - thanks!
U236M9FH9 : But if you have a fixed amount of strings you're going to use, you could make it its own type: `type BodyPart = Head | LeftArm | Torso | ...`, `type Message = Clicked BodyPart`, `onClick <| Click Head`
U236M9FH9 : And then the compiler will tell you if you accidentally misspell one :slightly_smiling_face:
U5VTA57UN : Architecture question. I have an API module with internal state that's needed to generate a Http.Request. Other child modules don't have access to that state, so they can't call the API module directly. Instead, I use an OutMsg from children back to Main. That OutMsg causes Main to tell API to send a specific request. I'm having trouble finding a clean way to get responses back to the children that initiated requests. The way I'm doing it involves a lot of tagging functions and making all my child models and msgs variable types (where the type variable is the main

Msg type). Any examples of clean solutions to this type of problem?

U236M9FH9 : <@U5VTA57UN> I have a model w/ an array of items nested under another array of items. I use indexedMaps to generate the HTML. When one of the nested items creates a Http.Request, I identify which one by adding the indexes to the msg: `type Msg = SaveSectionExercise Int Int (Result Http.Error SectionExercise)` , that lets me update the proper nested child.

U5VTA57UN : <@U236M9FH9> I see. I suppose I could do the routing using identifier strings (in place of the 2 Ints you use) and have main pass the response to the child that wants the response. But that means that for every call to the API in every child I'll need a unique routing string, and I don't get any compile time checks to make sure I didn't typo a routing string.

U236M9FH9 : The initiating children don't have some unique identifier in their model?

U48AEBJQ3 : <@U5VTA57UN> Talk of children and routing of messages sounds like you might benefit from letting go of how you are accustomed to architecting things. Instead of children, it is often useful to think in terms of data structures and functions. Evan's talk at Elm Europe might be worth watching for some guidance in this area:

<<https://www.youtube.com/watch?v=2ihTgEYiKpI>>

U5J08KX0D : <@U236M9FH9> Read my mind - Just worked that out - and it's fantastic!

Also

Eyes = LeftEye | RightEye

BodyPart = Head | Eyes | ...

I could get used to this help from the compiler :slightly_smiling_face:

U0SES5Y3T : so, when I do floating type division 0 / 0 I get NaN, but integer division 0 // 0 gives zero, is there another way to safely handle this case?

U0SES5Y3T : you can test this out in <<http://elm-lang.org/try|elm-lang.org/try>> if you want:``

import Html exposing (text)

main = text (toString (0 / 0))

``

U3SJEDR96 : <@U0SES5Y3T> the "issue" is that `NaN` is a Float thing, so `0 // 0` can't really return that. The other option is returning a Result, but that's pretty annoying, too. What would you define as safe?

U3SJEDR96 : (also, <<https://github.com/elm-lang/core/issues/721>> tracks that type of thing, there's a few interesting tidbits in there :)

U5QJW0DDE : i have a js project that i'd like to port to Elm as a test and am wondering if some of what it does would be difficult to do in Elm. In short, I take an SVG and render it to PNG in JS by using an HTML canvas as the conversion renderer. it looks like this in JS: <<https://paste.ofcode.org/XACLTwrZwaTgDvizZUY77T>>

U5QJW0DDE : i know Elm has good SVG support, but I haven't read much about dealing with canvas or the techniques described here, would it be fairly complicated?

U0CLDU8UB : You probably can't do that in Elm

U5QJW0DDE : good to know

U3KSN5MAL : Has anyone written a recursive rename function? My brain is being silly and i can't work out a clean way to do it atm,

need to check a list if a name duplicate exists, and then add a number, then recheck the new name ect

U3SJEDR96 : <<https://ellie-app.com/3C3LPJ6tmT9a1/0>> something like that?

U3KSN5MAL : Perfect!