

U0CQ254F5 : <@U5QJW0DDE> as for embedding elm within a react app, i agree with <@U3SJEDR96>, its relatively straightforward and a good way for teams to test it out

U5P4FLYLE : Hi, I would like to go elegantly from `values` type Poly

```
= Float Float
| String String
| Date Date.Date
xval=List Poly
yval=List List Poly
values=(xval,yval)
...
```

to extractedValues which has type `(List Date, List List Float)`

How do you pattern match something like that?

U3SJEDR96 : easiest would be `List.unzip values |> List.foldr (\val (dates, strings, floats) -> case val of *pattern match and prepend it to the correct entry*) ([], [], [])`

U3SJEDR96 : or you can do it in one go, too, if you write a function `Poly -> (List Date, List String, List Float) -> (List Date, List String, List Float)`

U3SJEDR96 : Or wait, I totally misunderstood :smile:

U3SJEDR96 : sec

U3SJEDR96 : <<https://ellie-app.com/3BNZD5X5DKKa1/0>> <@U5P4FLYLE> something like that I suppose

U5P4FLYLE : thanks, <@U3SJEDR96>

U60DVU1PS : Is there any article/documentation covering the reasoning for using "->" in anonymous functions?

Apologies if this is an obvious matter.

U2W8J47KQ : I would guess it's because Haskell does the same

U5Y2S6SJF : is it possible to pass in a "none" function to main's update argument? similar to subscriptions?

```
...
, subscriptions = (\_ -> Sub.none)
...
```

U0LPMPL2U : `always Sub.none` ?

U0LPMPL2U : oh wait, for `update` :stuck_out_tongue:

U5Y2S6SJF : yeah i figured that was coupled to subscriptions so didn't try

U0LPMPL2U : You want something like this?```

```
, update = (\msg model -> (model, Cmd.none))
...
```

U5Y2S6SJF : That should work. Get's a lot closer to my goal than writing out a basic update and pattern matching on a NoOp msg

U0LPMPL2U : I suppose you could also do:```

```
, update = always init
...
```

assuming you have an `init` function that returns an initial model and `Cmd.none`

U0CLDU8UB : Nope, that would expect just a single argument.

```
...
, update = \_ _ -> init
...
```

would work though.

U0LPMPL2U : good catch :thumbsup:

U5Y2S6SJF : thanks.. i guess i should of been able to figure that out if i read the types in the documentation

:slightly_smiling_face:

U0LPMPL2U : It can be easier to start by writing a constant `update` function before translating to an anonymous function. e.g.```

```
update : Msg -> Model -> (Model, Cmd Msg)
update _ model =
  (model, Cmd.none)
```

-- OR

```
update : Msg -> Model -> (Model, Cmd Msg)
update _ _ ->
  init
...
```

U5Y1YQD6Y : y'all, does Elm have a concept of `unless`?

U3SJEDR96 : Nope.

U0JFGGZS6 : `if (not x) then .. else ..` ?