U3LT1UTPF : Great!
U3LT1UTPF : Then, one must not call a Msg from another Msg, right, <@U3SJEDR96>?
U3SJEDR96 : Well, you can, but if you find yourself doing that, it _usually_ makes more sense to take the functionality from that branch that you actually want, make it into a separate function (like `withCurrentProduct : Product -&gt; Model -&gt; Model`) and call that from both places
U3SJEDR96 : so then you'd have `(model |&gt; withCurrentProduct product) ! []` in one place, and `( model |&gt; withCurrentCycle newCycle |&gt; withCurrentProduct (productInNewCycle |&gt; withDefault initialProduct) ) ! []` in the other place
U3SJEDR96 : the alternatives are:- calling `update` from `update` directly, i.e. making it recursive, which can lead to nasty bugs and doesn't seem necessary at all
- forcing `Msg` into a `Cmd Msg` and letting the runtime call `update` instead, which should make you wonder "why do I need to asynchronously call a function I defined myself?" and "what happens in between?", since your model will essentially be in an invalid state between those calls

U3LT1UTPF : Wowww... I get it :smile::bananadance: Thank you so much, <@U3SJEDR96>
U0CLDU8UB : My favorite alternative is to rethink what I am trying to do. Many times I can use the same Msg in a couple of places, instead of having two separate `update` cases.
U5P4FLYLE : Hi all, I am working with elm-mdl with card. And I am adding action block like below:```   , Card.actions
        [Card.border, css "vertical-align" "center", css "text-align" "right", backgroundColor]
        [Button.render Mdl [8,1] model.mdl
            [Button.icon, Button.ripple]
            [Icon.i "favorite_border"]
        , Button.render Mdl [8,2] model.mdl
            [Button.icon, Button.ripple]
            [Icon.i "event_available"]
        ]   ```
And it is added *below* the other blocks. What would you change to add it to the *right side* of already existing blocks? I bet it is more css question than elm-mdl one...

U0CLDU8UB : Of course sometimes I do need the two cases, but even then the separation might change with the rethinking.
U3LT1UTPF : Good advice, <@U0CLDU8UB> :smile:
U5H8JJP24 : Hi, I have this weird problem. If I write my functions like this:
```
newLocation : Result Http.Error Location -&gt; Model -&gt; ( Model, Cmd Msg )
newLocation result model =
    case result of
        Err error -&gt;
            handleHttpError error model

        Ok location -&gt;
            model |&gt; updateLocation location |&gt; fetchRoute



handleHttpError : Http.Error -&gt; Model -&gt; ( Model, Cmd msg )
handleHttpError error model =
    ({ model | error = Just (toString error) } |&gt; Debug.log "Error") ! []
```
I get the error:

```
Function `handleHttpError` is expecting the 1st argument to be:

    Http.Error

But it is:

    String
```

If I change the function annotation to:
```
handleHttpError : String -&gt; Model -&gt; ( Model, Cmd msg )
```
I get the error:

```
Function `handleHttpError` is expecting the 1st argument to be:

    String

But it is:

    Http.Error
```

U3SJEDR96 : are you calling that `handleHttpError` function from anywhere else? You first attempt looks correct to me...
U4872964V : yes, check the location of the error
U5H8JJP24 : oumph, thx <@U3SJEDR96> <@U4872964V>. I was searching at the wrong place. There was another call which caused the error... This took me 30 min to realise xD
U62R599PU : so new 'beginner' may be overstating it ... been trying to wrap my head around the concepts in Elm vs JavaScript. I have some basic framework (object ... record, some similarities, many differences ...that kind of thing). At a high level I understand subscriptions in the time clock sense or even keyboard input. The one area I can't seem to figure out is the equivalent approach/style to deal with observables.
U62R599PU : Can someone point me in the right direction. In the past I would have used Flyd observable streams.
U4872964V : in Elm, the concept of observable corresponds to a Msg
U4872964V : there is only one stream of Msg, they go into your update function
U4872964V : maybe that confuses you more, and now I have to go, sorry
U62R599PU : All the examples I have seen have been ... not sure how to say it... external. Does anyone know of an example where one variable impact another. For example, if game variable of "health" changes, a number of dependent variables also updates.
U0CLDU8UB : I would say there is no concept of observable in Elm. The entire program is a reaction to a "stream" of messages, like norpan said, but you can't really see it from the perspective of the developer
U0CLDU8UB : You're on the right track. You cannot do dependent streams in Elm. The concept just doesn't exist.
U62R599PU : Yeah, I kind of get that. So now I am looking for an example of the pattern I would use to do it in an Elm way. Have not found it yet.
U1CE9DL9H : well, when the health changes, a message must have been triggered
U1CE9DL9H : then in update, you can do whatever you need to move the model to a new, valid state, say `{ model | health = model.health - 5, durability = model.durability - 10}`
U0CLDU8UB : This is very abstract without a concrete example, I'm afraid.
U62R599PU : Yeah, your right. I guess I just need to dive in. <@U1CE9DL9H> that makes sense. I guess this model in my head where you assign relationships and move on needs rework.
U0CLDU8UB : I don't know if this helps or just confuses more, but here's an update with plenty of interdependent things from a game I'm building for fun: <https://github.com/ohanhi/bike-wars/blob/master/src/Bike.elm#L53>
U62R599PU : oh thanks, yeah, looking at examples for me is a great way to learn. Appreciate it.
U0CLDU8UB : Essentially there are just multiple calls to functions and some case expressions in the let block. They are interdependent too: figuring out the collisionPoint needs the nextPosition to be calculated etc.
U0CLDU8UB : But the thing is, in Elm you need to store _everything_ in the model. There is no other source of state in a program. If you think about it, every single Observable can be a source of state. This is why the two mindsets are a little hard to combine.
U62R599PU : In some ways that is how I ended up here. In JS I had just moved my world state to Baobab (immutable structure) and was struggling with 1. Changing my mindset 2. All the options and tools and what was right. Hence, Elm offers (I think) someone with entry to mid level skills (kind of new to this in general) ... structure.
U62R599PU : II will follow your code example through. Thanks for the help. Off to work.
U4872964V : <@U62R599PU> if you look at something like debouncing, which you'd use an observable for, this can be handled by explicit state, like done in the various debounce packages for Elm
U41NK9BM4 : pzolla: I'm doing an RPG in Elm and your observations resonate a lot. Coming from imperative language we need to reframe a lot fo things :slightly_smiling_face:
U4SM7ECAG : Hi there! I need some help with some 0.16 elm...

I've got some signal that 's not being handled properly...
in the main view I've a "escape catcher" that allows me to dismiss modals

```
 onWithOptions "keyup"
        { stopPropagation = False, preventDefault = False }
        keyCode
        (\keyCode -&gt;
          Signal.message address &lt;|
            if keyCode == 27 then
              App.Action.EscPressed
            else
              Debug.log "asdasdasd" App.Action.NoOp
        )
```

the issue is: I'm listening on the `change` event on some inputs which are being "ignored".
(i.e. the `App.Action.NoOp` gets triggered on any non-`esc` key press and so my view is re-rendered and my inputs reset.

Any idea of how I could listen to onkey on a single touch? or have something better than `App.Action.NoOp` for the case that doesn't interest me and that would not interfere with the user inputs?

U0LPMPL2U : Could you filter the signal in the part of the code that handles signals rather than having a conditional in the event handler?
U0LPMPL2U : Alternatively, write a custom decoder instead of using `keyCode` and make sure it returns `Json.Decode.fail` for non-ESC characters
U0LPMPL2U : For reference, see how the `onEnter` handler is implemented here: <https://github.com/elm-community/html-extra/blob/2.2.0/src/Html/Events/Extra.elm#L267>
U4SM7ECAG : oh that's a neat idea!
U3SJEDR96 : `escDecoder tagger = keyCode |&gt; andThen (\k -&gt; if k == 27 then succeed tagger else  fail "not esc")`
U3SJEDR96 : or, indeed, go all the way and make it a custom `onEsc` :slightly_smiling_face:
U4SM7ECAG : I'll try that :slightly_smiling_face:
U4SM7ECAG : Awesome that works !
U4SM7ECAG : Thanks a lot <@U3SJEDR96> and <@U0LPMPL2U>