U5X2VC483 : <@U3SJEDR96> thanks!

U5X84PY9G : <@U4872964V> do you mean: Encode.list &lt;| List.map Encode.string &lt;| Set.toList formData.threat_objectives_at_stake

U4872964V : looks right

U5X84PY9G : Ok thanks a lot

U0JFEBK6F : <@U5X2VC483> <https://github.com/fluxxu/elm-hot-loader> maintains state too :smile:

U23SA861Y : perhaps, but unlike in python the compiler enforces it. You also don't get ancestor imports either. I suspect it's less of an issue.

U2M4VPZ9D : <@U5X2VC483> I also used this <https://www.npmjs.com/package/create-elm-app> which packages up the elm-hot-loader mentioned by <@U0JFEBK6F>

U42H40UBW : Hi <@U0CL0AS3V>, I have a quick question about the SPA Conduit App that you shared a few weeks back. Is this the appropriate forum in which to ask it?

U42H40UBW : In your comments for the Main.elm file, you say that the majority of the code in that file will go away in 0.19. I've found that as I've built my app (based on your work), that Main.elm is getting bigger. I'm anxious that based on your comment that I'm moving in the wrong direction. so, when you say that Main.elm will largely go away what does that actually mean?

U42H40UBW : (I guess I'll just ask the question here :slightly_smiling_face:)

U42H40UBW : The comment in question was this: ```-- WARNING: this whole file will become unnecessary and go away in Elm 0.19,
-- so avoid putting things in here unless there is no alternative!
```

U0CL0AS3V : Evan is working on a design for code splitting that would mean each page has its own entry point

U5ABF3BH7 : Hello! Does anyone have an example of a search/autocomplete for an input field of a  select2. I am looking for something somewhat similar to <https://github.com/evancz/elm-sortable-table/>  , but in a select2 instead of a table.

U0CL0AS3V : that may get pushed back to the release after 0.19 in order to let 0.19 ship sooner, so I may end up regretting saying 0.19 specifically :sweat_smile:

U0CL0AS3V : point being, avoid putting logic in there

U0CL0AS3V : because it'll inevitably end up getting moved somewhere else when that file goes away

U0CL0AS3V : if all you're doing is things like expanding on the existing case-expressions in there, that's :thumbsup:

U0LPMPL2U : <@U5ABF3BH7> this may be close to what you're looking for: <http://package.elm-lang.org/packages/sporto/elm-select/latest>

U5ABF3BH7 : <@U0LPMPL2U> Thanks, I will take a look

U0LPMPL2U : I haven't used it myself but the examples on the demo site look kind of like select2 <https://sporto.github.io/elm-select/>

U42H40UBW : Thanks <@U0CL0AS3V>. My file is getting bigger to handle more case-expressions, so that's reassuring. Also, any indication as to when 0.19 might land :grin:

U5ABF3BH7 : <@U0LPMPL2U> This is not quite what I need. I need something where you can also see the choices you'd like to select from, not just an autocomplete field. You want the ability to see the content and also the ability to auto-complete. something like this <https://screencast.com/t/qsn74lvauf>

U0LPMPL2U : It looks like this is closer: <http://package.elm-lang.org/packages/sporto/elm-dropdown/latest>

U0LPMPL2U : (BTW, I'm getting these by searching for "dropdown" and "select" on <http://package.elm-lang.org/>. There are several libraries that there that might also fulfill your need)

U5ABF3BH7 : <@U0LPMPL2U> Yes, that is what I am looking for. Thanks again.

U5ABF3BH7 : <@U0LPMPL2U> after all, it is not what I am looking for, this one is just a select, the other one is just an auto-complete field, I am looking for something that does both. I will do as you suggested (searching more in <http://package.elm-lang.org/>.

U5ABF3BH7 : Haven't been successful at finding something that could be used both as select and auto-complete :disappointed:

U0CLDU8UB : Do you mean you want the list to open with all the options without typing but that you can "filter" it by typing something?

U3SJEDR96 : yeah, I think a basic select2 type thing (<https://select2.github.io/examples.html>)

U3SJEDR96 : I think both `elm-searchable-menu` and `elm-autocomplete` support such behaviour

U0CLDU8UB : Hmm, I suppose that is useful with some sort of data. We used this in our case (choosing the user's town of residence), but with a cursory look it doesn't look like you can make the dropdown open just by clicking the input. <http://package.elm-lang.org/packages/thebritican/elm-autocomplete/latest>

U333XNTNY : Hi! What's the best way to control the selected option of a `select`? I want to set it from the model :thinking_face:

U1CE9DL9H : <@U333XNTNY> that's done with the `selected` attribute
```

<http://package.elm-lang.org/packages/elm-lang/html/2.0.0/Html-Attributes#selected>

U1CE9DL9H : so typically you might do something like `List.map (\item -&gt; option [ selected (item == model.activeItem)] [ {- some html -} ]) allItems` in your view

U333XNTNY : I see, thanks

U29JSAR9S : hi guys, I'm sure I'm missing something really obvious but I can't figure out how to invert the boolean condition in this filter statement: `List.filter (String.startsWith "0")`

U0LPMPL2U : `List.filter not &lt;&lt; (String.startsWith "0")`

U29JSAR9S : ah, I was using `&lt;|` instead

U29JSAR9S : doh :slightly_smiling_face:

U29JSAR9S : cheers

U0LPMPL2U : Quick rule of thumb, if the result needs to be a function, you probably want `&lt;&lt;`

U0LPMPL2U : You can always combine manually using a lambda too: `List.filter (\n -&gt; not (String.startsWith "0" n))`

U0LPMPL2U : I'll often start with the lambda to get a feel for what I'm doing before refactoring to a `&lt;&lt;` implementation

U29JSAR9S : makes sense, and good advice - cheers

U29JSAR9S : since I've got someone who's obviously a few ElmFu belts ahead of me - is there a better way of writing something like this:
```
hexToDec : String -&gt; Maybe Int
hexToDec hex =
    String.split "" hex
        |&gt; List.map hexCharToDec
        |&gt; \decimals -&gt;
            if List.any isNothing decimals then
                Nothing
            else
                List.map (withDefault 0) decimals
                    |&gt; List.reverse
                    |&gt; List.indexedMap toActualValue
                    |&gt; List.sum
                    |&gt; Just
```

??

U48AEBJQ3 : @dans <http://package.elm-lang.org/packages/rtfeldman/hex/1.0.0/Hex#fromString> ?

U5ABF3BH7 : <@U0CLDU8UB> Sorry for the delay, yes, I need something that opens a list with all options without typing and something that I can also filter it by typing.

U48AEBJQ3 : er, <@U29JSAR9S>

U29JSAR9S : <@U48AEBJQ3> I was doing it as a learning exercise, figured it was a solved problem but seemed like an interesting one to solve again

U48AEBJQ3 : Well, then: <https://github.com/rtfeldman/hex/blob/1.0.0/src/Hex.elm#L14>

U0LPMPL2U : In your implementation, I would probably try a combination of `List.foldl` and `Maybe.andThen` rather than the conditional + `List.map` + `List.sum`

U29JSAR9S : cheers, I'll have a look into the `andThen` function

U0LPMPL2U : Actually, summing Maybes should be done with `Maybe.map2 (+)`, not `Maybe.andThen`

U0LPMPL2U : ```sumMaybes : List (Maybe Int) -&gt; Maybe Int
sumMaybes digits =
  List.foldl (Maybe.map2 (+)) (Just 0) digits
```
```