

U3SJEDR96 : you can put a case expression just about anywhere, for Elm, in the end, it's just another expression
U3SJEDR96 : for your own sanity, it might be worthwhile `_not_` putting it just about anywhere :stuck_out_tongue:
U2SR9DL7Q : This was my original concern? Part of it was just not knowing the correct syntax. But the other part is while it's simple now, the button will become part of it's own div. So Imagine two divs on the page. The one with the buttons that always exists. And the one with the game that only exists after the button is clicked. I say 'pages' but this is an SPA

U23SA861Y : umm, why would start game always exist?

U2SR9DL7Q : So I was wondering as to what collection of case statements, lets and ifs could produce something... sensible.

U2SR9DL7Q : Not start game specifically, but there'll be a row of buttons that's always there.

U23SA861Y : this is why game state might be a better abstraction

U23SA861Y : because then you can have something like ``header : State -> Html Msg``

U23SA861Y : think about how you want everything to be laid out in each case, what do you want it to look like?

U2SR9DL7Q : ... When the user clicks on the tab to get to page they're shown either a prompt to start a game, `*or*` if there's already a game in progress, that game is brought up.

U23SA861Y : ok so ```

```
case game of
  Nothing -&gt;
    renderStartGame
  Just game -&gt;
    renderGame game
...

```

U23SA861Y : startgame returns a div with a button in it, render Game returns the game board

U2SR9DL7Q : I was about to ask another question, but I `_think_` I just answered it for myself.

U2SR9DL7Q : I was going to ask what if I did just want a small piece of a page to show up or not. But I realized that a helper function just makes that this scenario again.

U2SR9DL7Q : And putting a case in a list is exactly the situation where I'd want to abstract with a helper

U23SA861Y : you always need to choose between one or more things

U23SA861Y : you don't get the option of a case/if returning nothing

U23SA861Y : what they return could be an empty ``div`` or a ``text ""``, but it has to be something

U2SR9DL7Q : spoiled for choice

U2SR9DL7Q : `<@U23SA861Y>` thank you `*very*` much, and `<@U0FP80EKB>_` as well

U64M3DF1B : `<#C12KMAYJX|elm-mdl>` I'm on elm 0.18 and using debois/elm-mdl - and this piece will not agree with me at all :cry:
...

menuItemView : AppState -> String -> Html Msg

menuItemView appstate name =

```
<http://List.li|List.li>
[]
[
  List.content
  [
    Options.css "cursor" "pointer"
    , Options.attribute &lt;| Options.onClick (ShowMenuItemPane name)
  ]
  [
    text name
  ]
]
...

```

I have the imports, the ``type Msg = ... | ShowMenuItemPane`` - but she tells me:

...

The right side of `(<|)` is causing a type mismatch.

```
167|           Options.attribute &lt;| (Options.onClick (ShowMenuItem name))
      ~~~~~
```

(<|) is expecting the right side to be a:

Attribute m

But the right side is:

```
Options.Property c Msg
...
```

U23SA861Y : do you even need the `Options.attribute <|`

U64M3DF1B : ? perhaps no! I'll tell you in 2sec :slightly_smiling_face:

U2SR9DL7Q : depending on the mdl element, I've had to use Options.attribute... since mdl expect a `Style a` type

U64M3DF1B : :partyparrot3: you're amazing! thx!

U64M3DF1B : for reference - this is what works (thx to <@U23SA861Y>)```

```
List.content
[
  Options.css "cursor" "pointer"
, Options.onClick (ShowMenuItem name)
]
[
  text name
]
```

...

U64M3DF1B : that - at least - is NOT obvious if you scout the package documentation :slightly_smiling_face:

(well - perhaps it is obvious and I'm just not totally converted to reading the Elm way of typing and spec'ing the syntax)

U5ABF3BH7 : Hello! Does anyone have a very basic example in Elm of an app with new page, edit page, index page, etc.. (ruby on Rails platform with controllers, etc..). I am getting a bit lost in the complexity of the examples I see.

U5ABF3BH7 : I have all the data from the backend, that I decode into my Model. I want to use the view of my FirstCallNew, I don't know how to enter that in

U5ABF3BH7 : ```main = Html.programWithFlags

```
{ init = init
, view = FirstCallNew.view
, update = FirstCallNew.update
, subscriptions = FirstCallNew.subscriptions
}```
```