U26LR8F4H : oh thanks, that looks pretty useful as well.
U153UK3FA : You can also use a port to get JS to query the DOM and pass it back to Elm, but that's quite a lot of effort
U64MK7215 : how do i use a for loop in elm?
U0JFXEUCT : there is no for loop, you'll need to check out `List.map`
U0JFXEUCT : there are other functions too, but it depends what you want to do :slightly_smiling_face:
U64MK7215 : i am trying to build a caesar cipher, for that i need to iterate through the strings
U153UK3FA : <@U64MK7215> sounds like a perfect use of `List.map`
U153UK3FA : a caesar cipher is mapping one character to a different character
U153UK3FA : <http://package.elm-lang.org/packages/elm-lang/core/5.1.1/String#map>
U64MK7215 : how to a define a custom function in elm and call it inside update?
U14Q8S4EM : What like this?
```
update : Message -&gt; Model -&gt; Model
update message model =
    Shift i -&gt;
        let
            by : Model -&gt; Model
            by model =
                model + i
        in
            by model
```

U5W5F6QGP : <@U64MK7215> you can declare it as a function on the top level and you can use it in your update function, this is an example where you declare an "addThree" function, which gets used in your update function
```
addThree: Int -&gt; Int
addThree num =
    num + 3
type Msg
    = AddThree

update: Msg -&gt; Model -&gt; Model
update msg model =
    case msg of
        AddThree -&gt;
            { model | count = addThree model.count }
```

U5W5F6QGP : or alternatively, you can use a `let...in` block like chadtech showed above
U14Q8S4EM : Yeah, I would delcare it top level way before I would declare it in the `let` statement like I did in the example.
U2U94G0QG : Had a break at work today and ran through <https://github.com/zwilias/elm-demystify-decoders>Sad that I finished it! Who knew writing JSON decoders in Elm would be such a nice bit o respite

U153UK3FA : <@U2U94G0QG> Don't worry, now you can use your new found parser combinator powers to parser things other than json.
U3LUC6SNS : <@U153UK3FA>, <@U2U94G0QG>, where can I learn about parser combinators?
U0CLDU8UB : <@U5QJW0DDE> Types help with the problems you're describing. Watch this talk: <https://www.youtube.com/watch?v=DoA4Txr4GUs>
U3LUC6SNS : Currently I am sending a long string to JS-land where it is rendered first by Asciidoctor.js, then by MathJax. This happens inside a div "rendered_text". I have a truly awkward set up where this div is carefully placed so as to look like an ordinary part of the Elm app. Is there a way of capturing the rendered HTML in this div by a subscription and displaying it in an ordinary HTML element in the Elm app?
((In fact I am using <@U0JFXEUCT> 's `style-elements`, so I want to display it in one of those))

U3LUC6SNS : <@U48AEBJQ3> I noticed you mentioned ways of building apps other than passing `model` around. That is what I have been doing, and it has worked for me so far. However, I'm very interested in learning about alternatives.
U153UK3FA : <@U3LUC6SNS> for more general parsing you can use <http://package.elm-lang.org/packages/Bogdanp/elm-combine/3.1.1/Combine> you'll notice that the concepts there are

very similar to those used in Json.Decode