

U5QJW0DDE : Elm might wrap numbers, for example, into its own numbers type (perhaps?) but does it do something more complicated than this

U3SJEDR96 : For the most part, they're just type-checks. Like, a string in JS or a String in Elm, it's the same thing. A record is essentially a JS object. A few things need little more work, like lists. I wouldn't worry about it, if it does become a problem, I'm sure we can figure out a way to get it working :slightly_smiling_face:

U5QJW0DDE : i'm trying an experiment to port part of an existing Clojurescript app to Elm. The Clojurescript app uses React via Om and is extremely fast, handling streaming financial data dozens of times per second, streaming over a socket.

U5QJW0DDE : I'm hoping there won't be any surprises to Elm's handling of this kind of data at this speed

U3SJEDR96 : Dozens should be fine, generally. Handling a subscription to requestAnimationFrame (which fires at 60fps) is perfectly fine in Elm. One thing to do is to try and keep calculated data in your update low; prefer to just take the data in, letting the render manage the calculated data instead. If you find the overhead of the DOM rendering to be too slow, adding some `Html.lazy` nodes high up in the tree should take care of that, too

U5QJW0DDE : i don't follow you with regards to calculated data; can you express with a simple example?

U3SJEDR96 : say you receive a string that represents an integer, prefer to save that string in your model rather than doing the `String.toInt` and matching on the result in your `update`.

U5QJW0DDE : oh, i see

U3SJEDR96 : Though even that is something to worry about only after you notice you need it. Start with the most simple, most obvious setup. If you do get into trouble, benchmark and figure out how to speed things up :slightly_smiling_face:

U5QJW0DDE : is relying on Chrome's dev tools typical in the Elm world, or are there more Elm-specific ways to profile and debug?

U641LDZFU : Given this function ``delay : Float -> msg -> Cmd msgdelay seconds msg =
Process.sleep (seconds * Time.second)
|> Task.perform (_ -> msg) ``

U641LDZFU : used as a Cmd like `` (nextModel, delay 999 (NewStoryNode newNode)) ``

U641LDZFU : To pass a random number instead of 999

U641LDZFU : Do I need somehow store it in the model, and go through setting up new Msg types

U0P088J1L : <@U5QJW0DDE> I assume you have tried Elm debugger?

U641LDZFU : or am I missing something?

U5QJW0DDE : <@U0P088J1L> no, but i will now! i'm quite green on Elm at the moment

U5QJW0DDE : doing more high-level research of the lang and its ecosystem

U5QJW0DDE : just started writing some code today

U641LDZFU : aka is there a way to generate a random number without Msg (s)

U641LDZFU : ... "You cannot just get random values willy-nilly. You create a command, and Elm will go do some work behind the scenes to provide it for you. In fact, any time our program needs to get unreliable values (randomness, HTTP, file I/O, database reads, etc.) you have to go through Elm."