U0LPMPL2U : `5` is both a `Dollar` _and_ an `Int` because they are aliases for each other (with the type alias)
U3HQVHERX : How would I use the `Dollar` type in the `Dollar` module? And why are opaque types useful?
U3HQVHERX : er - how are they usefule
U0LPMPL2U : you can change the implementation without breaking code
U0LPMPL2U : for example, say you changed `type Dollar = Dollar { dollar : Int }`
U3HQVHERX : Wouldn't that change how it's used?
U0LPMPL2U : Since you couldn't ever reach inside of the `Dollar` type in your code, this won't break for you because you aren't depending on the internal implementation
U3HQVHERX : i think i need to see an example of it being used in a different module
U0LPMPL2U : Sure
U3HQVHERX : `import Dollar exposing (..)` then what
U0LPMPL2U : so you have something like:```
module User exposing (..)

import Dollar exposing (..)

type alias User = { balance : Dollar }

payday : User -> Dollar -> User
payday user paycheck =
  { user | balance = Dollar.sum user.balance paycheck }
```


U0LPMPL2U : This code doesn't care _what_ the implementation of a `Dollar` is
U0LPMPL2U : as long as the `Dollar.sum` function works as expected
U0LPMPL2U : As a library maintainer, this is really powerful because I can make changes to the structure of `Dollar` without breaking everyone else's code
U0LPMPL2U : for example, if I add the concept of cents
U0LPMPL2U : now the `Dollar` type looks like `type Dollar = Dollar { dollars : Int, cents : Int }`
U0LPMPL2U : I can release a new version and not break your code because you only depend on the `Dollar.sum` function, not on a particular structure for `Dollar`
U0LPMPL2U : make sense?
U3HQVHERX : `Dollar.sum` takes 2 `Dollar`s and returns a dollar, but the whole time i don't really care what the `Dollar`s are,
U3HQVHERX : coooool
U0LPMPL2U : yup :thumbsup:
U3HQVHERX : Should i be doing this in my own domain specific modules?
U0LPMPL2U : It depends?
U0LPMPL2U : That's not really helpful :stuck_out_tongue:
U0LPMPL2U : For "utility" structures that are used all over the place probably yes
U0LPMPL2U : For example, if you're doing a lot of things that involve coordinates, it would probably be worth having an opaque `Point` or `Coordinate` type
U0LPMPL2U : That way, if you end up making a change to it later such as adding a third dimension, you don't have to make changes all over the code ( continuation of conversation with <@U3HQVHERX> )
U0LPMPL2U : You probably want two things:1) A way to keep track in your model of which note(s) are animiated 2) A conditional in the view that only adds the class to the HTML based on that data in your model

U0LPMPL2U : <@U5KEJPXHP> ^^
U0LPMPL2U : <@U3HQVHERX> there's a section in the design guidelines that discusses opaque types: <http://package.elm-lang.org/help/design-guidelines#keep-tags-and-record-constructors-secret>
U5L8E6N2V : Hi all, I am trying to make an Elm wrapper for the Chosen js library.  I am having trouble capturing onChange events (events aren't firing), perhaps someone would be willing to take a look at what I've done: <https://ellie-app.com/3BkyBmJwkRha1/0>
U0LPMPL2U : Does chosen suppress change events on the hidden dropdown?
U5L8E6N2V : I'll check, that's a good lead
U4JT89FGB : Just guessing, but you might need to get chosen's events via port
U5L8E6N2V : Makes sense as Chosen hides the original select and creates a new dropdown out of divs.  Thanks to both of you!
U5Y1YQD6Y : Hey y'all, total beginner here.  Struggling with the repl.  Why does this `fruit = { type = "apple" }` throw a syntax error (`The = operator is reserved for defining variables`)?

U23SA861Y : type is a reserved keyword
U23SA861Y : use `kind` or `type_`
U5Y1YQD6Y : ha.  thank you
U23SA861Y : np :slightly_smiling_face:
U0LPMPL2U : <@U5Y1YQD6Y> you may want to use a different key name that's more descriptive or domain specific than `kind` or `type_`. In this case, perhaps something like `name` ?
U0LPMPL2U : Also, welcome to the community! :slightly_smiling_face:
U5Y1YQD6Y : no worries <@U0LPMPL2U>, I'm just messing about in repl.  thanks for the welcome!
U601ELFEG : I'm working on my first elm app - and it will have on the order of 2k + DIVs generated from the model - think of like a kanban or thing with a lot of small things the user can re-organize.
U601ELFEG : I want to have lots of feedback - like hovering over one of those things will hilight related things
U601ELFEG : should I be worried about re-generating this 2k+ node DOM with every mouse move event?
U601ELFEG : do I need to use the HTML node with id thing?
U48AEBJQ3 : <@U601ELFEG> It's not that difficult to add performance improvements like that after you have things working the way you want. You don't save much if any effort doing it from the start. Write it first and then worry about performance.
U601ELFEG : indeed - I plan on writing the "straight forward way" first - but was just wondering if people felt 2k HTML nodes wandering into "yer gonna have to optimize that" territory
U601ELFEG : on the plus - as I picked up elm last night - and have a skeleton already running (okay okay - so I've written a ton of Haskell before - and a tone of javascript / html /css --- so  picking up elm was pretty easy) --- and I'm really liking it (if missing some Haskell common functions...)
U23SA861Y : depends on if all 2k of those divs are changing
U23SA861Y : simultaneously
U48AEBJQ3 : Nothing inherently overwhelming about a couple of thousand HTML nodes on the Elm side of things, but it always depends on implementation details.
U0JFGGZS6 : I wouldn't think it a big deal, from what you describe it's just changing attributes not forcing nodes to be removed and added
U601ELFEG : my guess is that the classes on about a dozen of them will change as the user moves the hover - and if during a drag, a several dozen will be changing position as the user drags
U601ELFEG : yup - just attribute changes (really, just classes and x/y positioning) - so  the differ will get this efficiently? if so, awesome!
U23SA861Y : You may need to key things, or salt lazy here or there but it doesn't sound like there is any show stoppers.
U601ELFEG : excellent
U601ELFEG : synth patch manager here I come!
U601ELFEG : (or rather, here elm comes!)
U5Y1YQD6Y : Does anyone have experience with using the time-traveling debugger with a third-party build tool (e.g., brunch, webpack, browserify)
U5Y1YQD6Y : ?
U0EUHKVGB : <@U5Y1YQD6Y>, for webpack, just add `debug=true` to the end of the compiler options
U2XRG0UKA : ```{
  test: /\.elm$/,
  exclude: [/elm-stuff/, /node_modules/],
  use: {
    loader: 'elm-webpack-loader',
    options: {debug: true, warn: true, verbose: true}
  }
}
```

U2XRG0UKA : Is one of the loaders in my webpack.config.js file
U3KSN5MAL : having trouble googling this. How do i unstringify some json?
U3LUC6SNS : Hi all, I am going to need a debouncer in my app. Currently, each character of user input sets off a whole train of actions, including sending text through a port where it it is processed as asciidoc, and also to an elixir backend to save the data. OK for small docs, but not large ones. I've seen several debounce packages -- any recommendations?
U3LUC6SNS : Something like `JSON.parse`
U3KSN5MAL : ok i'll have a look thanks!
U3LUC6SNS : <@U3KSN5MAL> Here is a referennce: <https://www.w3schools.com/js/js_json_parse.asp>
U3KSN5MAL : thanks!

U5YDC1YUW : what's the general style for variable naming in Elm, `someLongVariable` or `some_long_variable`?
U3KSN5MAL : Most code i've seen just uses the first