U0J8D9M2P : there is <#C0HJVT881|elm-css> channel
U4PT7NG4V : Is there a hamburger-menu package people could recommend? :slightly_smiling_face:
U60SXAF96 : Is there an easy way to defer the execution of a Cmd for a specified period of time?   I've got an interstitial that I'd like to automatically close after X seconds.
U60SXAF96 : (Where "X" is determined by the business and therefore subject to frequent and arbitrary changes.)
U2LAL86AY : <@U60SXAF96> from what i know, no - once you get a CMD from a function you can only give it to elm - can't do anything with it.
It's only possible to have a task - and you can chain different tasks such that it matches all your conditions. Example - waiting for multiple servers to respond - before you get a response msg back in elm
But in your case - can't you use ports - let js tell you when to close X?
or if is a time interval you need to wait -&gt; you can use the time library.

U0JFGGZS6 : there's also Process.sleep <http://package.elm-lang.org/packages/elm-lang/core/5.1.1/Process#sleep> which you could chain to the front of your Task before passing it into a Cmd
U2LAL86AY : ericgj: couple of days ago i asked how to do that multi dimensional chat layout - there was that problem of calculating the height - of boxes containing text - and i wasn't able to do it in elm.
I followed your advice on `ports`- and not `ready/ready states` - and worked out pretty awesome. Just wanted to let you know that your advice was very helpful :smile: I will come back with a gif later today.

U0JFGGZS6 : glad it helped, but I think it was at least partially someone else's idea :wink: anyway talking it out often helps regardless!
U625M23DE : i'm working on a single-page app which produces SVG as an output. i'm contemplating how to get compile-time guarantees about reusable chunks of my views, specifically that they have appropriate aspect ratios.
U625M23DE : so with that in mind, i defined a type like this:
U625M23DE : ```type alias GroupBox ratio msg = { scale: Int, contents: List (Svg msg) }```
U625M23DE : and use it with a bunch of auxiliary types, like this:
U625M23DE : ```type Ratio11 = Ratio11type Ratio41 = Ratio41
button41 : String -&gt; GroupBox Ratio41 msg
button41 label = ...```

U625M23DE : but this induces a large amount of repeating myself, in that `Ratio41` isn't actually tied to the ratio 4:1 anywhere. i considered just defining the `Ratio` type with parameters so that i could extract those parameters, but this would lose the compile-time guarantee that whenever i render a thing the parent and child must agree about the aspect ratio that should be used.
U625M23DE : so, given that elm doesn't have typeclasses, is there a tidier way to keep the compile-time safety in place?
U1CE9DL9H : <@U625M23DE> it looks like phantom types might be helpful here
U1CE9DL9H : ```type Ratio41 = Ratio41

type alias Element ratio msg = Svg msg

compose : Element ratio msg -&gt; Element ratio msg -&gt; Element ratio msg
```

U1CE9DL9H : depending on exactly what you want you'll probably have to extend/change that a bit.
U1CE9DL9H : the trick here is that `Element` takes an extra type variable `ratio` that isn't actually used
U1CE9DL9H : ah i see now that is what you use
U625M23DE : that's similar to how i'm doing it now: i have a function which takes a ratio-tagged element and a ratio-tagged parent request for rendering, and renders it. but the problem is that the actual ratio values aren't retrievable:
U625M23DE : ```render : GroupBox ratio msg -&gt; ParentContext ratio -&gt; Int -&gt; Int -&gt; Svg msgrender box pc actualW actualH = ...
render41 : GroupBox Ratio41 msg -&gt; ParentContext Ratio41 -&gt; Svg msg
render41 = render 4 1```
```