U5E3DU81K : how do I get html character entities to show rather than literally display their code?
U3SJEDR96 : <@U5E3DU81K> can you give an example?
U4872964V : <@U5E3DU81K> you can just use the unicode directly
U4872964V : or do you get them from somewhere else?
U6GB56346 : Thanks. Maybe I've understood it. `[]` is a special kind of `List a` and `text "hello"` is a special kind of `Html msg`. They are nothing so they will become everything.
U3SJEDR96 : Yeah; that works :stuck_out_tongue:
U6CB44HMF : Hello all, I am giving a talk on how to understand the Elm Architecture when coming from Redux, and I had a question. `combineReducers` in redux makes the shape of the store, and sets up the global reducer all in one step. Is there any way to do this (set up both the update function and init function when combining multiple reducers) all in one step in Elm?
U4872964V : <@U6CB44HMF> no, not really. In the Elm architecture, those things are separate. It's a move away from thinking in terms of components
U6CB44HMF : <@U4872964V> Yeah I thought so, thanks!
U4872964V : There are a few packages that can help you though, I don't know if any lets you combine the init and update functions though
U4RR7KX45 : I'm trying to delay a command for 2 seconds and can someone help me out with type annotation please?```
type Msg
  = UpdateText String

updateText : Cmd Msg
updateText =
  sleep 2 (Task.perform UpdateText "hello world")
```
It's saying that the second argument of Task.perform is causing a mismatch

U31FGNWCT : It seems like you should pass a task as a second parameter, not a stringperform : (a -&gt; msg) -&gt; Task Never a -&gt; Cmd msg

U4RR7KX45 : `Task.perform UpdateText Task "hello world"` ?
U31FGNWCT : Without "hello world", I guess
U4RR7KX45 : says it can't find variable Task :disappointed:
U3SJEDR96 : `Process.sleep (2 * Time.second) |&gt; Task.perform (\_ -&gt; UpdateText "hello world")`
U3SJEDR96 : the `\_` is to ignore the value `sleep`'s task succeeds with (`()`)
U4RR7KX45 : yes, that worked :slightly_smiling_face: I use Task / Future monad in JS a lot, but I can't figure out how it works in Elm
U4RR7KX45 : so Process.sleep generates some Task right?
U3SJEDR96 : Yep. A Task which succeeds with `()`. To perform it, you have to provide a `Msg` that uses this value, which is what will end up being handed to your `update` on completion
U3SJEDR96 : an alternative way to write the above would be `Process.sleep (2 * Time.second) |&gt; Task.andThen (\_ -&gt; Task.succeed "hello world") |&gt; Task.perform UpdateText`
U4RR7KX45 : is `andThen` the same as `chain` ?
U3SJEDR96 : which waits 2 seconds, then succeeds with `"hello world"`, which you perform with a tag for your `msg` that takes that parameter
U4RR7KX45 : ah so `Task.perform UpdateText (Task.success "hello world")` would've worked I guess
U4RR7KX45 : *succeed
U3SJEDR96 : Yeah, but it would execute immedeatly, since it's not chained with your timeout
U4RR7KX45 : yeah
U4RR7KX45 : ok I get it now, thank you :slightly_smiling_face:
U4RR7KX45 : so for example's sake, if I do```
Task.succeed (2 * Time.second)
  |&gt; Task.andThen Process.sleep
  |&gt; Task.perform (\_ -&gt; UpdateText "hello world")
```
that should work too

U3SJEDR96 : Yap
U4RR7KX45 : nice!

U4RR7KX45 : thank you so much :slightly_smiling_face:
U4RR7KX45 : one more question please :slightly_smiling_face:
```
updateText : List Project -&gt; Cmd Msg
updateText projects =
    projects
        |&gt; List.map
            (\p -&gt;
                if p.published then
                    Task.succeed p.title
                 else
                    Task.fail p.title
            )
        |&gt; Task.sequence
        |&gt; Task.perform SetProjects
```
Again I get a type mismatch:

expected:
`Task String (List String) -&gt; a`

actual:
`Task Never (List String) -&gt; Cmd Msg`

how should I search where I'm making a mistake?

U3SJEDR96 : `Task.fail` means you no longer have a task that can never fail
U3SJEDR96 : so you need to use `attempt` to attempt executing it
U3SJEDR96 : which means `SetProjects (Result String (List String)`
U3SJEDR96 : to capture either the `Err "I failed"` or the `Ok [ projects ] `
U4RR7KX45 : I wanted to see how `sequence` works, but I guess I cannot do it without actually using HTTP