U2D7NUGS1 : :boom: :smile:
U3SJEDR96 : Ah, too bad I can't make it there, I'd have loved to see that :slightly_smiling_face:
U2D7NUGS1 : I hope it will be recorded.
U2D7NUGS1 : You may take a look at <https://github.com/lzrski/elm-polymer/tree/pre-publish>
U2D7NUGS1 : It's work in progress. Unpublished and very poorly documented.
U2D7NUGS1 : What I'm trying to do now is reimplement the scaffolding script in Elm. It's getting too hairy for shell scripting.
U2D7NUGS1 : Basically I'm trying to generate Elm code with Elm program :slightly_smiling_face:
U2G4P6QJ2 : Is there some convenient syntax for "updating" these kind of types? -&gt;```
type MyType = MyType Int String
```
I know for records I can do this:
```
{ record | someField = someValue }
```
But for the first kind I keep desctructuring the type and "rebuilding" it from scratch every time..

U3LT1UTPF : If it's recorded, please share (I also speak Dutch)
U3LT1UTPF : <@U2D7NUGS1> ^^
U2G4P6QJ2 : Right now I do this:```
changeNr : MyType -&gt; Int-&gt; MyType
changeNr (MyType _ str) newNr = MyType newNr str
```

U48AEBJQ3 : <@U2G4P6QJ2> You are already doing the right thing.
U48AEBJQ3 : Although, `MyType` should be the last parameter. It makes pipelining easier.
U48AEBJQ3 : And you might do `mapNr : (String -&gt; String) -&gt; MyType -&gt; MyType`
U2G4P6QJ2 : Thanks a lot <@U48AEBJQ3>. I will keep that in mind, it makes perfect sense.
U62Q5C999 : Hi, just started with learning elm. Coming from react redux. I wonder is there something like redux store in elm? Some global state container and message bus to change it?
U2G4P6QJ2 : Yes there is. Check out the docs on "The Elm Architecture". <https://guide.elm-lang.org/architecture/>.
U2G4P6QJ2 : On that page, `Model` can be seen as the state that your redux store contains, and `Update` is pretty much your reducer.
U2G4P6QJ2 : Or "rootReducer" if you use that concept.
U3SJEDR96 : So two things: try not to think of the elm architecture in terms of "how would I do this in react". It's going to give you a lot more trouble, pretty quickly. And, interestingly, redux based its model off of elm's architecture :)
U62Q5C999 : <@U3SJEDR96> that's how I got here :) but have hard time going through documentation, maybe it's just because I have wrong mindset
U62Q5C999 : I think I've already got how to structure update and model
U62Q5C999 : But it's hard to imagine how some nested components can listen to updates
U62Q5C999 : Without passing model all the way down from parent
U3SJEDR96 : There are parallels and somewhat subtle but important differences that are sometimes made by choice, sometimes inherent to the two languages. Trying to learn elm while doing some sort of "internal translation" is definitely going to give you some headaches, tho. Letting go of "components" etc is step one.
U3SJEDR96 : I hope someone else can give you some more guidance here, I have to go for now :)
U63LX5HFY : Hey guys i'm new here. could you help me with a Json decoding problem with my code. I'm mapping over a list of emojis .the code works but i think it's maybe an error in logic. here is the repo : <https://github.com/nynd/Emoji-Search>
U451CRP62 : Hi everyone, I'm still a beginner here. Can anyone tell me how to convert a String into a Float? I used String.toFloat, but I end up with a "Result String Float", and have no idea how to extract my float number from it!
U62V8HFJR : `String.toFloat` returns a `Result String Float` because it's possible the initial string doesn't decode well
U62V8HFJR : e.g. `"hello"` doesn't really turn into a float
U62V8HFJR : so you have to handle two cases: `Ok value`, in which case `value` will be the float you want
U62V8HFJR : and `Err error`, in which case `error` will be a `String` describing the error you received while trying to parse out the float
U1L4GLFJ6 : any way I can get the current time in Elm
U1L4GLFJ6 : ?
U451CRP62 : How do I do this, because the value returned from String.toFloat doesn't have a field named Ok, nor a field named value?
U62V8HFJR : `Result` is a type with two constructors: `Ok value` and `Err error`

U62V8HFJR : your code will likely look something like this:
U62V8HFJR : ```case Result.toFloat str of
    Ok x -&gt; x
    Err error -&gt; &lt;whatever value is appropriate for an error&gt;
```

U62V8HFJR : <@U1L4GLFJ6> <http://package.elm-lang.org/packages/elm-lang/core/5.1.1/Time#now>
U62V8HFJR : But keep in mind that `now` gives you a `Task`, meaning getting the time is asynchronous.
U1L4GLFJ6 : <@U62V8HFJR> how does that Task turn into a string?
U1L4GLFJ6 : is there `succeed` ?
U451CRP62 : I'm sorry, but I don't get it. I try this on the repl but I get a type mismatch error:
U62V8HFJR : <@U451CRP62> That's because `x` and `error` have different types
U62V8HFJR : Elm is strongly typed, so you can't have variables that switch between `Float` or `String` willy-nilly like in Javascript or Python
U62V8HFJR : so the result of `Ok x -&gt; &lt;VALUE&gt;` and `Err error -&gt; &lt;OTHER VALUE&gt;` will need to match
U62V8HFJR : A simple option would be to use `Maybe`:```
b = case String.toFloat a of
    Ok x -&gt; Just x
    Err error -&gt; Nothing
```