U0JFXEUCT : ultimately, yeah
U0JFXEUCT : `Html.map` just calls that constructor on all the event msgs of the view
U0JFXEUCT : You've checked out the `elm-spa-example`?
U0JFXEUCT : there's also a great talk about growing elm code, by Richard Feldman at Elm Europe
U6DQCTZT2 : I've seen he's elm-spa-example
U6DQCTZT2 : But haven't seen the talk
U6DQCTZT2 : I'll have a look thanks :slightly_smiling_face:
U6DQCTZT2 : <@U0JFXEUCT> wait, I've seen it lol :stuck_out_tongue: Do you mind sharing more about your Common type?
U0JFXEUCT : lol, awesome
U6D3ERLA1 : for instance, can I import the JS Math lib and use Math.random
U0JFXEUCT : umm, yeah, honestly it's pretty simple. So, Im working on an editor of sorts, where multiple views need to be able to change the selection.
I have a top level file called `Action.elm` which defines a type called:

```
type CommonAction
    = DeleteSelection
    | SetSelection Selection

```

Then in my main, I just have

```
type Msg
    = Common Action.Common
    | OtherStuff

```

and in the pages that need to do stuff I jsut say

```
div [onClick (Common DeleteSelection)] []
```

U0JFXEUCT : you would also have to tag your other events manually instead of using `Html.map`
U6FFMA51S : Very useful link.
U6DQCTZT2 : Tagging as in the way I use lift in Page.view?
U6FFMA51S : <https://learnyouanelm.github.io/pages/03-types.html> says that Int is 32 bits. Is that correct?
U0JFXEUCT : yeah, manually applying stuff
U6DQCTZT2 : <@U0JFXEUCT> I was actually about to do a global action file as well so thanks a lot for sharing :heart:
U0JFXEUCT : <@U6DQCTZT2> No problem! one thing to think about too is "does page actually need it's own update"
U0JFXEUCT : anyway, good luck!
U6FFMA51S : I just tried to answer my own question by using `elm-repl` and got: ```&gt; a=2147483647
2147483647 : number
&gt; a+1
2147483648 : number
```
So there it doesn't look like 32 bits.

U6DQCTZT2 : <@U0JFXEUCT> Yeah, a lot of my pages doesn't need a model or update. Hmm btw, you'll have to tag the common msg as well, right? So basically the views will have something like `Module.view CommonMsg ModuleMsg someModel`?
U0JFXEUCT : You would, yeah.  Honestly, I just import `CommonMsg` into the view that's using it instead of passing it in.
U6DQCTZT2 : <@U0JFXEUCT> so you're not defining it in `Main.elm` then, since that would create a circular import? Maybe Elm doesn't struggle with that?
```

U0JFXEUCT : yeah, CommonMsg is in a separate file
U0JFXEUCT : and imported by Main and by the Page
U6DQCTZT2 : <@U0JFXEUCT> that's awesome :slightly_smiling_face:
U3SJEDR96 : No, it's not strictly 32bits. It's 50-something. Under the hood, they're just JS numbers, which are all 64bit floats.
U3SJEDR96 : <@U6D3ERLA1> yeah, random is a little harder in fp because functions have to be pure - every time you call a function with the same inputs, it should return the same output without any side-effects. Things that look simple in other languages, like getting a random number or the current time, require a bit more work in Elm and other fp languages because of that. So, random gives you two options, either running a generator as a cmd, or doing manual seed management and using `step`, taking care to store the new seed in your model.
U6D3ERLA1 : this has led me down the ports rabbit-hole