

U3SJEDR96 : `<http://Html.td|Html.td> [] <| List.intersperse (Html.text " , ") [Html.a [] [], ...]`

U1G7E1P9S : thanks

U6DUP1WMA : Has anyone deployed an Elm app into an android WebView? I'm thinking about using elm to implement a part of my android app that has to do with a WebView...

U0EUHKVGB : <@U6DUP1WMA> yes, I have

U0EUHKVGB : Works fine

U6DUP1WMA : cool, might give it a shot. Thanks

U6FK3K430 : <@U6DUP1WMA> we are doing so now using React Native to wrap it.. works fine on Android

U6DUP1WMA : Wow that sounds dangerous, but I'm glad it's working fine for you. Will definitely try it out.

```
U1G7E1P9S : `` let roomNumber =
    if List.length (String.words room) > 1 then
        List.head (String.words room)
    else
        room
in ``
```

U1G7E1P9S : how do I get rid of `Maybe` type to just get a `String`?

U48AEBJQ3 : <@U1G7E1P9S>

```
``
roomNumber =
    List.head (String.words room)
    |> Maybe.withDefault room
``
?
```

U23SA861Y : yeah so rather than check first that it's possible to get the head, you should try to get the head and then deal with the fact that you couldn't

U48AEBJQ3 : It might even make more sense to not use a default and actually handle the error case in a more intentional way.

U1G7E1P9S : Ah

U1G7E1P9S : In this case I find it awkward, because I'm only going to evaluate that branch if `String.words room` produces a list > 1, so of course there's a head

U1G7E1P9S : but that works much better, thanks

U23SA861Y : The compiler doesn't really have anyway of knowing the list length was just checked

U48AEBJQ3 : If you know the list has values, you might try using a data structure which models a non-empty list as early as possible.

U2LAL86AY : I want to clarify something for my brain :simple_smile: - can you point out if i have some mistakes in the following statements? I'm interested in what happens to the app before the first message arrives. :simple_smile:

1. when the app first starts the first function called is init. Is not a real function more like a value - but if it contains functions inside they get called.

2. update does not run until the first message arrives.

3. view is rendered using the model generated by init function.

4. what about subscriptions? it is safe to assume that the subscription function is called?? or will not be called until i receive the first message?

U23SA861Y : I believe it is called immediately after init generates your model

U23SA861Y : and then subsequently after each update

U23SA861Y : it is possible to write headless programs and without subscriptions being registered early the program would never start

U2LAL86AY : :simple_smile: makes sense

U23SA861Y : I'm not actually sure if it's after "each" update or if the message queue is burned down and then it is called

U2LAL86AY : and i also want to know - do you believe it's possible to take elm code - and add debug.log calls in certain places? something like``

```
myFunction x =
    x + 1
``
```

transformed into:

```
...
```

```
myFunction x =
```

```
  let
```

```
    _ = Debug.log "argument x " x
```

```
  in
```

```
    x + 1
```

```
...
```

It's for my debuggger - i'm trying to come up with ways of inspecting functions on the go.. Maybe not debug.log but Inspector.log or something that works similar with how Debug.log is implemented.

U23SA861Y : you can add debug.log like that yes

U2LAL86AY : no i mean doing that automatically. using the ast or something.

U23SA861Y : you can do it there, or because the x is passed through you can also ```

```
myFunction x = Debug.log "argument x" x |> ((+) 1)
```

```
...
```

U2LAL86AY : hmm.. i need to do some research on that for sure. One more question. I can't wrap my head around this. When a function is partially applied: Say `add x y` is only applied with `add 5` -> this in principle means it has:

```
...
```

```
add y =
```

```
  5 + y
```

```
...
```

IS there a way to see that `5`? I mean to see the values that are already passed in? Because `x` is known, is bound to the `add y` function, i would like to see it somehow. `Debugging.log` `add 5` gives back `<function>` -> not Debug.log - but do it in some other way - can it be actually possible to see that 5 there?

How function application is actually implemented? I'm still in researching phase now - so any hint is a good one - not looking for a full blown explanation - just some point were to start wrapping my head around this :simple_smile:

U5X2ZRFDF : Sure, there's a way to see the 5. Just call `add 0` and you'll get `5`.

U5X2ZRFDF : I haven't seen the implementation code, but I'll bet function application in Elm is just implemented as function application in JavaScript.

U2LAL86AY : :smile: that's a nice trick :smile: No i'm looking for a way to see inside partially applied functions in general. that only works will add 5 not with `Maybe.withDefault (SomeComplexDecoderwarped as maybe)`

U5X2ZRFDF : No, there's no way to do it in general. Closures are, well, closed.

U5X2ZRFDF : That is, function values