U663M2MB7 : I guess that would not be entirely accurate, but my understanding of Elm and where it differs from Haskell is limited.
U3LUC6SNS : I want to start writing tests for the parser I'm working on. What are the recommended packages and practices for this?
U23SA861Y : there is an elm-test package I believe, but I couldn't tell you anything more than that as I haven't used it
U2GPAEU1L : Hey everyone, if you're curious how you would go about writing encoders/decoders for recursive data types, I just made a tutorial here where I encode/decode a binary tree:
<http://codetidbit.com/#view/snipbit/5966ae6a7890d775c05191ef/introduction>
Additionally you can check out the entire series on elm encoders/decoders here:
<http://codetidbit.com/#view/story/58f7ac012bdce7111285c2ea>

I cover concepts all the way from the basics of working with the decode pipeline all the way to more complex things such as working with tagged unions and recursive data structures.

Cheers :smile:

U300LJUAK : Neat little tutorial <@U2GPAEU1L>. Never had to decode such structures, but it's sure nice to see how it's done.
U663M2MB7 : Has anyone used <https://github.com/rtfeldman/elm-css> without having it create an external css file? I would like to use it in my code, and have it style my elements when running `elm-reactor` but I am confused as to whether this is an possible.
U46JV6X3K : What's the best way to encode a record to a JS object?
U68RXKS5D : Check out Json.Encode
U68RXKS5D : <http://package.elm-lang.org/packages/elm-lang/core/5.1.1/Json-Encode>
U0CL0AS3V : <@U663M2MB7> definitely doable - ask in <#C0HJVT881|elm-css> and folks can help!
U48AEBJQ3 : <@U46JV6X3K> Presuming you want to make a JS object:```
type alias MyRecord =
   { foo : Int
   , bar : String
   }

encodeMyRecord : MyRecord -> Encode.Value
encodeMyRecord myRecord =
   Encode.object
      [("foo", <http://Encode.int|Encode.int> myRecord.foo)
      , ("bar", Encode.string myRecord.bar)
      ]
```

U663M2MB7 : <@U0CL0AS3V> thanks man, will do! :slightly_smiling_face:
U65B9414J : Folks posted my first blog on Elm. This was an issue for me and my team members while understanding Elm. It's very basic stuff. <https://blog.bigbinary.com/2017/07/12/difference-between-type-and-type-alias-in-elm.html> . If anyone has any feedback then let me know. Thanks everyone and keep on Elming.
U3LUC6SNS : I need to frequently to the below after starting `elm-repl`.  Is there a way of putting this in a script or something to avoid repetitious typing?
```
> import LatexParser.Latex exposing(..)
> import LatexParser.Parser exposing(..)
> import Parser exposing(run)
```

U494Y62N7 : huh, i guess it would be nice to do something like `elm-repl YourCode.elm`
U3LUC6SNS : <@U494Y62N7> I tried this:```
jxxmbp:koko_client carlson$ elm-repl start.elm
Unhandled argument, none expected: start.elm
```

U494Y62N7 : Yea, I was thinking that would be nice. F# has the ability where you can just highlight code & send it to the repl.
U494Y62N7 : Clojure does as well.

U494Y62N7 : I guess it's time to start learning Haskell :slightly_smiling_face:
U3LUC6SNS : Another question. In the `elm-repl` I do the following:```
 r = run latexList "a b c\nx y z\n"
Ok { value = [Words { value = ["a","b","c"] },Words { value = ["x","y","z"] }] }
    : Result.Result Parser.Error LatexParser.Parser.LatexList
```
Is there an easy way to extract the part after the `Ok`?  Working with multi-line code in the repl is a pain.

U494Y62N7 : The only thing I can think of is to write a function with `Result.withDefault` and then have the `value` just be an empty array if it fails
U494Y62N7 : Sorry :disappointed:
U153UK3FA : <@U3LUC6SNS> `Result.withDefault` is what you want. But you might just want `Result.map` if you're planning on using that value for something
```