U41NK9BM4 : Well, it does exactly that. Also the name implies that you should find a better way to deal with it sooner or later :slightly_smiling_face:
U5WS7CJLV : question about Json.Encode (with context). In javascript it is so easy to go JSON.stringify({ isThisEasy: true }) - but in Elm, one cannot simply Json.Encode.encode { isThisEasy = true }
U5WS7CJLV : why is that?
U3SJEDR96 : <http://package.elm-lang.org/packages/Gizra/elm-keyboard-event/1.0.0/Keyboard-Event>
U5WS7CJLV : <@U5GSY0G9J> - I am new to this myself, but I think if you sent some kind of "Key13Pressed" Msg then you could kind of handle it that way
U5WS7CJLV : but maybe you need a solution more general than that
U3SJEDR96 : (you can't actually simulate the "click", but you can handle what _would_ happen on a click)
U0CLDU8UB : You can likely set the same Msg for both clicks and enter-presses
U0K92QFST : <@U5WS7CJLV> I think the most direct answer to your question is that the `Encode` package prefers type safety and reliability to convenience
U5WS7CJLV : I see. For me, I just want my JSON stringified ASAP so I can make a web request. Seems like there's a lot of resistance for something so simple (in javascript).
U0K92QFST : Theoretically you should be able to reliably encode an Elm record, but the type signature wouldn't really make sense. It would be:```
encode : a -&gt; Value
```

U0CLDU8UB : This should help you get going: <http://eeue56.github.io/json-to-elm/>
U0K92QFST : which suggests that you can encode ANYTHING, which isn't true (union types, etc)
U0K92QFST : you would need something similar to `comparable`, i.e. `encodable` which signifies the types that can be encoded to valid JSON representations
U5WS7CJLV : my brain just wants to stringify records because the syntax and concept seem real close to a javascript object
U5WS7CJLV : I know it is not, but it is hard to be rational sometimes
U0K92QFST : Right, but what if there's a union type in your record?
U0K92QFST : what does the `encode` function do?
U3SJEDR96 : You'd also need runtime type introspection so look at "what are the fields of this record?", which means compiler support, etc etc. While realistically, Elm's encoders/decoders allow you to decouple your internal representation from the API format, and this is a very powerful feature
U0K92QFST : Right. Long story short, it would be a lot more complicated. I could see something like that being implemented down the line, but for now you need to deconstruct your record.
U0K92QFST : the nice part is that once you implement your `toValue` function it will always work, guaranteed :smile:
U0K92QFST : and if you ever need to change the shape of your record, the compiler will guide you through the refactor
U5WS7CJLV : good answers.  The thing that I will probably remember is that "stringifying a record is not as simple as it seems"
U0K92QFST : yeah! one of the things Elm is great for is showing you how much complexity you've been overlooking all these years. IO, handling errors, etc.
U5WS7CJLV : I'm only just getting started here, so my records are all of a shape that they would be easy to stringify
U0CLDU8UB : I've also found many times we wouldn't actually need to transfer that much data over the wire, but since it's more convenient to just put the entire JS object in, we do that. :slightly_smiling_face:
U5WS7CJLV : :slightly_smiling_face: yep, that's true
U5WS7CJLV : and that's what I'm doing in this case since I'm still mostly experimenting
U0CLDU8UB : When you need the encoder, you have a step where you go "Oh, I only need the id and a boolean here"
U0CLDU8UB : Yeah, that's fair. It'll be nice in real projects though, I promise!
U5WS7CJLV : <@U0CLDU8UB> - that link is helpful - thanks!
U635238TG : if i'm new to programming and web dev in general would it be too hard to learn elm as my first language? it just speaks to me, haha, and the whole js landscape chaos...doesn't. plus i feel learning from a strong typed language will make me a better developer in the long run.
U3SJEDR96 : It would be very interesting, but there _may_ also be some frustration along the way - a lot of tutorials will (blindly) assume some prior knowledge
U635238TG : that was my fear too. i was thinking maybe there were intro to functional programmers for complete beginners that i could use alongside elm specific tutorials
U635238TG : *programming
U0K92QFST : I think definitely, definitely start with Elm
U0CLDU8UB : I think Elm is one of the most _designed to be approachable_ languages out there, but it is true that most tutorials assume some programming knowledge
U0K92QFST : it will teach you good habits from the start
```

U0K92QFST : whereas most of us had to unlearn bad habits :slightly_smiling_face:
U635238TG : exactly what is drawing me in!
U635238TG : i think i'm just going to go for it. i did the same with learning vim, it just felt like the right approach for me, and it's been tough but so fun. js didn't keep me up until 2am last night researching it, elm did. i think i'm going to trust my gut. could bode badly for you folks though. haha
U0K92QFST : keep the questions coming!
U0J1M0F32 : If you get stuck, let us know :smile:
U0CLDU8UB : Somehow your attitude sounds a lot like mine when I started! I wrote this article back then, (it's very outdated now but) the foreword has a certain similar tone: <http://ohanhi.com/learning-fp.html> (a more recent version of the same intro is here: <http://ohanhi.com/base-for-game-elm-017.html>)
U2SR9DL7Q : learning FP... your brain gets that "oh. I like this. Let's do more of this."
U3SJEDR96 : Heh, I definitely recognize those feelings, <@U0CLDU8UB>. Nothing has ever "clicked" for me the way Elm has :slightly_smiling_face:
U2SR9DL7Q : i always wonder what the state of contemporary programming would be like if most people _started_ FP, then did imperative afterwards? Would we find it similarly fascinating upon discovery?
U635238TG : that is what I'm hoping for. I was probably doing too much research on js fatigue and the state of things and how it was all a necessary evil and I was just like, "is it?" That rabbit hole led me to Elm and I was just thinking, why not start with a better approach right off the bat even if the learning curve is steep.
U2SR9DL7Q : I'm currently in the process of skipping js entirely (besides learning the basic syntax ad usage) and just learning web dev straight through elm. I'd like to write about it eventually, and hopefully get some feedback from js developers. As <@U0K92QFST> said, unlearning bad habits was a big concern for me. javascript looks so... rickety. it seems like a language best used after a thorough grounding in good design principle.
U0K92QFST : <@U2SR9DL7Q> I would be super interested to read about it!
U0CLDU8UB : Me too! Do you have a blog?
U2SR9DL7Q : <@U0K92QFST> i wont know for awhile if it was the smart decision, but elm was the tipping point in me deciding to do web development in the first place. I said, "if I already know that good web design incorporates FP principles, and I know that js devs are enjoying transitioning to elm, then I should just skip the middle step?"
U2SR9DL7Q : no blog yet, currently building the site (with elm!)
U0CLDU8UB : Nice!
U0CLDU8UB : Sorry for the off-topic, but this tweet just fits very well into the discussion here:<https://twitter.com/aprilwensel/status/880822230229303298>

U0CLDU8UB : I am so glad the Elm community in general lets people enjoy their journeys!
U2SR9DL7Q : js from the outside looks insurmountable. frameworks and tools and platforms and layer upon layer or opinions and styles. i came from python where 'there is one obvious way to do it' is a big thing, along with not needing to venture beyong Just Using Python until you're ready (mainly because people love writing good libraries in the language. elm said, 'use me and you can get a site up and running'. no jquery, no react, no angular, no blog posts telling me how to use jquery, no blog post telling me no one uses jquery anymore...
U2SR9DL7Q : mind you, a lot of this is because elm is new, and uncluttered. it will be interesting to see how it matures, but i am so far extremely impressed with evan, and richard and all the other folks who have brought the language to where it is right now.
U2SR9DL7Q : useful. approachable. rational. i can probably only use one of those adjectives for Haskell :neutral_face: _end rant_
U635238TG : how would you folks bring this same approach/philosophy to the back end? what should I start researching for when I eventually want to learn about that side of things?
U2SR9DL7Q : um... most of my back end work i've done with python's flask framework. very minimalistic, very clean. personally, i like to avoid too many moving parts when i'm learning a new thing. a red flag for me is when i see a tagline that goes. "Just take [x] and run it on [y] with [z] for your [q] in the background, and the rest is a breeze!" I'm sure whatever they're talking about is helpful, but I want to learn one thing well, not four things badly.
U635238TG : i'm also seeing elixir mentioned a lot with elm. any insight there?
U2SR9DL7Q : i'm currently looking for a FP language with a simple back-end framework like flask. Haskell has Yesod, which i looked into before, but at the time i didnt know enough about Haskell, or back end development to get it going. I hear good things about elixir, and pheonix
U635238TG : i'll keep those in the back burner for now. thanks everybody, i'm excited for this new journey and community. I just happen to be going into a 10 day silent meditation retreat in a couple days though so I have to find a way to temper my desire to jump head first into elm until my return. thanks again for the discussion and encouragement!
U2SR9DL7Q : my goal is something with just the right amount of 'magic'. Elm's handling of model, update, view is an example. I know _what_ elm is doing (manipulating the virtual dom and managing state), even though i'm not sure exactly *how* elm does it. If i can find a similar server side experience, I'm all for it.

U2SR9DL7Q : <@U635238TG> good luck and enjoy!
U1L4GLFJ6 : what is the Elm idiom for finding the 1st or nth item of a list
U1L4GLFJ6 : it's suprisingly hard in `List`
U23SA861Y : as a list is a linked list, there is potentially a better way to do random access
U0FP80EKB : first item is with `List.head` or by destructuring in a case
U1L4GLFJ6 : how about the 17th item?
U1L4GLFJ6 : <@U23SA861Y> yes
U0FP80EKB : `List.drop 16 &gt;&gt; List.head`
U0FP80EKB : `nthItem index = List.drop (index -1) &gt;&gt; List.head`
U0FP80EKB : `nthItem : Int -&gt; List a -&gt; Maybe a`
U0FP80EKB : `List.Extra` has a lot of these in them
<http://package.elm-lang.org/packages/elm-community/list-extra/latest/List-Extra>
U1L4GLFJ6 : and the something to handle the `Maybe` types... I'm very nervous about `List.Extra` I'd rather build them myself
U0FP80EKB : I wouldn't be nervous. However, for simple things like this, building your own is great.
U0FP80EKB : you handle the maybe with either a case or perhaps `Maybe.withDefault`
U0FP80EKB : ```nthItemWithDefault : Int -&gt; a -&gt; List a -&gt; anthItemWithDefault index default = nthItem index &gt;&gt; Maybe.withDefault default```

U23SA861Y : I would still suggest that if you need random (nth element) access, then list is the wrong datastructure to use.
U23SA861Y : there is a reason those methods are not in the stdlib and it's intentionally annoying to use
U0FP80EKB : Personally, I prefer to use lists for all my base data structures until I need something else. It is nice to have a good standard structure that all my code uses. Until I'm really frequently doing random access, I stick with lists. However, if the whole point is random access, then, yeah, probably the wrong data structure.I find the cognitive overhead of trying to remember which "listy" data structure I'm using isn't worth it. YMWV :slightly_smiling_face:

U0FP80EKB : In our system, we don't have any other listy data structure except `List`
U0FP80EKB : We had a case where we thought we needed random access, but it turned out we really needed a zipper, so we built one on top of `List` :slightly_smiling_face:
U23SA861Y : precisely my point
U0FP80EKB : This, I think, is the blanket statement I'm not 100% onboard with, at least for my stuff. I think it could be a bit more nuanced as "if the primary reason for this structure is random access"But, of course, context rules

U0FP80EKB : I'd probably say "if you need a random access data structure, you might look at whether you really do" :slightly_smiling_face:
U0FP80EKB : especially for beginners, I'd say "stick with the list!"