U1EEBCQM6 : Yes.. correct
U0LPMPL2U : OK, that should be possible without a `Dict`
U0LPMPL2U : let me see if I can write up something
U1EEBCQM6 : Not that I would have multiple lists. Just one list of all the options. And the `attribute` would allow me to filter based on that.
U1EEBCQM6 : I thought that might be most efficient
U3SJEDR96 : <https://ellie-app.com/3xfPcjxfy4ja1/0> something like that?
U1EEBCQM6 : check that out! Let me try to understand it.
U0LPMPL2U : I wrote up <https://ellie-app.com/3xfTdwMjGCra1/0>
U0LPMPL2U : looking at <@U3SJEDR96>'s solution :eyes:
U0LPMPL2U : ooh! `Json.Decode.keyValuePairs` super nice! TIL :thumbsup:
U23SA861Y : are we playing code golf```
itemdec : Dec.Decoder (List {attribute:String, id:String, name:String})
itemdec =
  Dec.map2 (,) (Dec.field "id" Dec.string) (Dec.field "name" Dec.string)
    |> Dec.list
    |> Dec.field "options"
    |> Dec.keyValuePairs
    |> Dec.map (List.concatMap (\(n,l) -> List.map (\x -> (n,x)) l))
    |> Dec.map (List.map (\(a,(i,n)) -> {attribute=a,id=i,name=n}))
    |> Dec.field "attributes"
```

U2FP79HN3 : So uhm, I just expanded a case of statement into 33 possible matches and now it takes about 3 minutes to compile
U3SJEDR96 : Yeah, the exhaustiveness checker goes a little crazy with large case of stmts. Should be much better in 0.19, tho
U2FP79HN3 : make it ten
U37PQL9PB : I just discovered that elm treats this as a valid type signature: "foo : List Int Bool -&gt; Int". I don't understand how that's valid. I can understand "foo: List Int -&gt; Int", but I don't understand why "Bool" is allowed in this context when it's not a separate argument. Is this a bug or can someone educate me as to what this means?
U3SJEDR96 : It's a bug, and it's been logged :)
U3SJEDR96 : (same with `foo : List -&amp;gt; Int` btw)
U37PQL9PB : Great thanks. I tried searching for it, but google failed me. It messed me up as I was trying to do "List (List a)" but typed "List List a" and got confusing error messages :wink:
U5ABF3BH7 : Hello! Does anyone know if there is a good package implementing 'search' in a select2
U5ABF3BH7 : ?
U2FGMAQ5R : <@U5ABF3BH7> , I was looking for something similar and started looking at implementing select2 in Elm , is there a project like this already ?
U5ABF3BH7 : <@U2FGMAQ5R> I don't know. I need to implement something using select2 search. I am just starting.
U236M9FH9 : <@U5ABF3BH7>  <@U2FGMAQ5R> Check out <https://github.com/thebritican/elm-autocomplete>
U5ABF3BH7 : <@U236M9FH9> Thanks!
U5HM74BD0 : I have a json string I need to decode. It's complex enough (in my mind, anyway) that I don't understand how to use Json.Decode with it to break it down into an Elm model. Its structure is something like this:
```{playerNames: ["Fred", "Barney"], games: [{date: "May 20", stats: [{ab:2, hits: 1}, {ab: 3, hits: 0}]}, {date: "May 27", stats: [{ab: 1, hits: 1}, {ab: 3, hits: 1}]} ]}```

I'm not sure how to begin to decode this. Can I use Json.Decode on its own or should I use the elm-decode-pipeline package, or something else?

U153UK3FA : <@U5HM74BD0> you can use Json.Decode on it's own. When writing a Json decoder start by writing a decoders for the child nodes and compose those together to create decoders for the parent nodes
U153UK3FA : eg. start by making a decoder for `{ab:2, hits: 1}`
U5HM74BD0 : <@U153UK3FA> So begin with my records?  Yes, okay.  Then how do I use the result of that with its parent?
U153UK3FA : once you have a decoder for `{ab:2, hits: 1}` you can pass that decoder to <http://package.elm-lang.org/packages/elm-lang/core/5.1.1/Json-Decode#list> and get a decoder that decoders lists of `{ab:2, hits: 1}`
U5HM74BD0 : Okay, sounds good. Let me give it a whirl.  Thank you.
U153UK3FA : <@U5HM74BD0> note that what you pasted above isn't valid json since the field names need to be

quoted
U5HM74BD0 : Yes, you're right. I was short-cutting it with my representation.  I do have valid json with quotes in the right places.
U153UK3FA : alrighty, I just didn't want you to get frustrated trying to write a parser for invalid json
U5HM74BD0 : Thanks for that.  Here's the actual json I'm working with:```{"playerNames":["Fred","Barney"],"games":[{"date":"May 20","stats":[{"ab":2,"hits":2},{"ab":2,"hits":1}]},{"date":"June 3","stats":[{"ab":2,"hits":2},{"ab":3,"hits":3}]},{"date":"June 4","stats":[{"ab":2,"hits":2},{"ab":2,"hits":2}]},{"date":"June 10","stats":[{"ab":2,"hits":2},{"ab":2,"hits":0}]},{"date":"June 17","stats":[{"ab":2,"hits":2},{"ab":3,"hits":2}]},{"date":"June 19","stats":[{"ab":0,"hits":0},{"ab":4,"hits":4}]},{"date":"June 20","stats":[{"ab":2,"hits":2},{"ab":0,"hits":0}]}]}```