U0JL9RPC4 : It means you shouldn't manipulate an `Attribute` _directly_ but through some exposed functions
U57KYFW67 : Opaque types are essentially what C++ and Java would call abstract types
U57KYFW67 : You never call constructors for them directly and you never pattern-match on them with `case` directly. You have to go through the public API to work with them.
U57KYFW67 : It's like "all rights reserved" for the implementation, since the library owner can make changes without fear of breaking any existing code
U1L4GLFJ6 : OK. `Svg` exposes itself. so does `Html` but `Attributes` doesn't not. Fine
U57KYFW67 : I can't vouch for the design choice on `Html`, but it's a bit odd to see that it does.
U57KYFW67 : Although if you look, it shows it's just an alias for `Node`, which itself is an opaque type
U1L4GLFJ6 : OK new question
U1L4GLFJ6 : i want to decode a complicated JSON file
U1L4GLFJ6 : how am I suppsed to know the structure all at once?
U1L4GLFJ6 : ```{municipios: [`geometry`: [[[....]]]] }```
U1L4GLFJ6 : I'd just like to dump the entire json file as as string
U57KYFW67 : sadly, I haven't worked through the JSON tutorial yet, or I could probably help &gt;_&gt;
U57KYFW67 : How is the `Array` type implemented?
U4F64AKQV : <@U1L4GLFJ6> You just got the import slightly wrong. It should be `import Html exposing (Attribute)`
U4F64AKQV : Attribute being an opaque type is somewhat peripheral to the matter
U4F64AKQV : <@U1L4GLFJ6> You need to create a decoder. <http://json2elm.com> might be helpful.
U4F64AKQV : I'm not sure what you mean by "knowing the structure all at once." Don't you need to know that anyway in order to extract the parts of the data that you care about in JS.
U3LUC6SNS : <@U1AN4JRFV> Thanks so much -- yes, it turns out that with <@U0JFXEUCT> 's 3.2 update to style-elements, the problem is solved for those using that package - code at <https://github.com/jxxcarlson/nanoedit> in case anyone else runs into this. :slightly_smiling_face::+1:
U0EUHKVGB : Folks, explaining what an opaque type in this context was not the right path to go down, and can lead to more confusion than aid. It's best to identify what the questioner is actually stuck on before answering in such situations
U0EUHKVGB : Especially in this case since it was an import error :slightly_smiling_face:
U57KYFW67 : Yeah. I just kind of went into Wiki mode :X
U5XHTBFS6 : When I do `port addSpotMarker : Geolocation -&gt; String -&gt; Cmd msg` elm-make complains that it has the wrong type, but `port addSpotMarker : Geolocation -&gt; Cmd msg` type checks (note: Geolocation is a `type alias` for a record). Does it mean ports must always take a single argument?
If that's the case, what's the best way of passing multiple values through ports? I thought maybe doing `port addSpotMarker : (Geolocation -&gt; String) -&gt; Cmd msg` and having the JS-side subscriber take an array, but that seems error-prone. I could also make `addSpotMarker` take a single record that "embeds" both parameters, but that seems too verbose. Is there a better way?

U153UK3FA : <@U5XHTBFS6> yep, only one argument
U153UK3FA : you can send a tuple or a list
U153UK3FA : `port addSpotMarker : (Geolocation, String) -&gt; Cmd msg`
U153UK3FA : the JS side will get it as an array
U5XHTBFS6 : hmmm that's what I thought, then. thanks!
U153UK3FA : Because it's talking to JS it's error prone either way, if your JS callback to the subscription took less arguments than it was given those values would still be lost
U5FU80S06 : I want to have a global `Int` count which increments by one to give latest count. But everything is immutable, so I do not know what to do.
U4JT89FGB : <@U5FFVDC4W> See something like this: <https://ellie-app.com/D9Q8Xy9sdza1/2>
U57KYFW67 : Do you need to specify anything more than the "main" .elm file to elm-make?
U57KYFW67 : Oh nice. It looks like it.
U3HQVHERX : jdm06: `Json.Decode.field "municipios" Json.Decode.field "geometry" Json.Decode.list Json.Decode.list Json.Decode.list` .... :slightly_smiling_face: I would reccomend playing around in the repl for a good while and decoded small parts, then put them all together
U0J8D9M2P : Hello. On this example <http://elm-lang.org/examples/binary-tree>, on 5th exercise is saying that `flatten: 21` - means that it's possible to write `flatten` function with `fold` so the length of function definition will be 21 character including function name and whitespaces.
U0J8D9M2P : My solution is```
flatten : Tree a -&gt; List a
flatten = fold (::) []
```
Which is 22 character. How I can do it with 21? Or is it typo?
```

U5Q1H5XC2 : Hey, i would like to see how my elm code looks like after its compiled to JS. Whats the best way? When I compile with elm-make I cant find my own code.
U3SJEDR96 : <@U5Q1H5XC2> look for `_user$project`. It will probably not be all in one place, but that's to be expected. Function names are preserved, so you can look for those, too. Be aware that the compiled JS is an implementation detail, though :slightly_smiling_face:
U5Q1H5XC2 : Sweet, thanks :slightly_smiling_face:
U3SJEDR96 : <@U0J8D9M2P> probably a typo, that looks correct
U37HUSJ4R : Hi all, how can I pass multiple arugments to a `List.map` function? I currently am passing my type alias `Foo` but I also want to pass a float called `time`
U37HUSJ4R : currently the type signature looks like `renderFoo : Foo -&gt; Html Msg`
U3SJEDR96 : `List.map (myFun withArgs andMoreArgs) someList`
U3SJEDR96 : assuming you have a list of `Foo`, you just have to make sure that's the last argument, so you can partially apply the others, first
U37HUSJ4R : perfect