U57KYFW67 : Let ? = 3. Then ? = 3. I have shadowed the variable ?. Not good practice, but you can't say I'm wrong, because I just gave a definition for what I mean by ?.
U57KYFW67 : The only thing is that some variables have historical importance, and are used nearly universally in certain ways. But that's convention, not logic.
U23SA861Y : we coloquially refer to archimedes constant as pi because its frequent use as the symbol. But the constant itself doesn't change
U57KYFW67 : A good example is something like ?, the Euler-Mascheroni constant. Surely ? is used in lots of ways, but it is also an important naturally-occurring number -- not all that different from ?.
U23SA861Y : the concept of a fixed ratio between a diameter and a circumference that is the constant.
U57KYFW67 : jonf: Just taking this aside, since it's not programming-related. But if you look at situations like multiple integrals, you can have "constants" which "vary" Like ??xydxdy (taken over, say, a rectangle) which is equal to ?y (?xdx) dy. We say the variable y is constant with respect to x. It's treated as constant because when you zoom in on the inner integral "?xydx", you can see the quantifier for x (the "dx" in a sense brings the variable x into existence), but you can't see the quantifier for y until you zoom back out.
U23SA861Y : y isn't a constant in that scenario it is a variable of the inner function f(y,x)=xy you are simply recoginizing that there exists an identity g(y) = ?xydx = y * ?xdx
U57KYFW67 : Right. But that requires a global perspective. Locally, (when you confine your analysis to just the inner scope) it acts in all ways exactly like a constant.
U57KYFW67 : The perspective I'm arguing here is useful if you're, say, writing a compiler, and you want to concern yourself with local data whenever possible.
U57KYFW67 : But ultimately, you can take different perspectives on it.
U5ABF3BH7 : <@U23SA861Y> thanks for helping me earlier. I am not fetching the data though. The decoder isn't doing its job and I get an empty list when I shouldn't . Would you might taking a look to see if you catch some error in my code?Going to the url "cases/frontend/all_rolodex" fetches the data so the error isn't there.
```
entryDecoder = Json.Decode.map2 Types.RolodexEntry (Json.Decode.field "id" <http://Json.Decode.int|Json.Decode.int>) (Json.Decode.field "name" Json.Decode.string)

categoryDecoder = Json.Decode.map2 Types.RolodexList (Json.Decode.field "category" Json.Decode.string) (Json.Decode.field "list" (Json.Decode.list entryDecoder))

getRolodexLists =
  Json.Decode.list categoryDecoder
          |&gt; Http.get "cases/frontend/all_rolodex"
          |&gt; Http.send Types.LoadRolodexLists
```
```
type alias RolodexList =
   { category : String
   , list : List RolodexEntry}

type alias RolodexEntry =
     { id : Int
     , name : String}
```
```


Types.LoadRolodexLists (Ok rolodexLists) -&gt;
   ({ model | rolodexCategoriesAndEntries =  rolodexLists}, Cmd.none)

Types.LoadRolodexLists (Err _) -&gt;
    (model, Cmd.none)
```

U5ABF3BH7 : I also get the data with ```Cmd.batch [..., ..., getRolodexLists]
U153UK3FA : <@U5ABF3BH7> are you sure the http request isn't erroring?
U153UK3FA : what does the json from the response look like?
U3HQVHERX : Could someone provide me with an example of `uniqueBy : (a -&gt; comparable) -&gt; List a -&gt; List a`?
U3HQVHERX : Like `unique [1,1,2] == [1,2]`

U3HQVHERX : `unique ?? [1,1,2] == [1,2]`
U3HQVHERX : from List.Extra
U57KYFW67 : <@U3HQVHERX> I don't know if this is right, but my guess would be something like unique (\p -&gt; p.age) personList
U57KYFW67 : or to be a bit more concrete, unique (\x -&gt; abs x) [-1, 2, -3, 4, 1, -2]
U236M9FH9 : :point_up: