U4872964V : <https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/text-anchor> perhaps
U4872964V : oh, you want to change the size of the box, hard
U4872964V : the would depend on the font rendering, so either you can approximate it or you have to wait until it has rendered and adjust the box
U2LAL86AY : can someone can help me understand why this library is using `comparable` here and not `number`?<https://github.com/jschomay/elm-bounded-number/blob/1.0.0/src/Number/Bounded.elm#L35>

U2LAL86AY : what is the advantage? or there is none?
U5J0H2NS2 : Just a higher level of abstraction I guess.
U5J0H2NS2 : You can use it for any comparable type.
U2LAL86AY : yeah but comparable means `strings` and `list of strings` - and it doesn't make sense to "bound" a list of strings - in the context of this library- or i'm missing something..
U2LAL86AY : oh and i found something more interesting. How can i create a function of type : `number -&gt; Float` the fancy search returns nothing.<http://klaftertief.github.io/elm-search/?q=number+-%3E+Float>

U681TBBUP : The only notion needed for bounding something is comparability, maybe it doesn't make sense in some contexts but there isn't a reason to unnecessarily restrict it
U3ZNWN526 : Hmm... There's a strange thing happening (or maybe I'm just missing something obvious): I have an extensible record type `WithUserFields` and an extended record `UserFields`, which doesn't add any additional fields:```
type alias WithUserFields a = { a | ... }
type alias UserFields = WithUserFields {}
```
But when I try to use `UserFields`, the compiler spits out "cannot find variable `UserFields`"

U2LAL86AY : how do you try to use `UserFields` can you show that particular example?
U2LAL86AY : and you can't have `WithUserFields`  and `WithUserFields ` twice in the same module i think - ups i thougth is a `type` - delete that :smile:
U5QJW0DDE : Incidentally this is only true if your data model is structured the same as your UI. Rarely does a UI page equate to an exact segment of your data model.
U3ZNWN526 : Sure! The first case for example, is:```
userFieldsFrom : User -&gt; UserFields
userFieldsFrom u =
  UserFields u.userRole u.username u.name u.email u.password u.passwordConfirmation
```

U3ZNWN526 : Well for `WithUserFields` the first time is defining it, and the second time I'm using it (to define `UserFields`)
U5QJW0DDE : I look forward to watching that.
U2LAL86AY : there is a small article about extensible records here. I can't find the other one i saved once - was much better at explaining how to compose extensible records. <https://dennisreimann.de/articles/elm-data-structures-record-tuple.html>
U2LAL86AY : ok let check it out
U2LAL86AY : i simplified a bit:```
type alias WithUserFields a =
    { a | name : String }

type alias UserFields =
    WithUserFields {}

userFieldsFrom : String -&gt; UserFields
userFieldsFrom string =
    WithUserFields string

```
error:
```
Cannot find variable `WithUserFields`

```

U2LAL86AY : so it seems like you can't have parametric types act as constructors
U3ZNWN526 : Oh, no, I have `UserFields` as the constructor within that last function, not `WithUserFields`
U2LAL86AY : Maybe is the same thing - since  it gives the same error.
if you have `type alias Thing = String `
you get back a function named `Thing : String -&gt; Thing`.

But if you have `type alias Thing a  = String `
Seems you don't have this `Thing: String -&gt; Thing` anymore. It's how elm works.
 Let me try a few more things first

U3ZNWN526 : Ooh, so you think even with `type alias Thing = MetaThing {}` maybe wouldn't give you the constructor?
I suppose that's possible... very unexpected though, and doesn't seem to be documented anywhere
```