

U3SJEDR96 : I'm a little unsure what you're really after, tbh  
 U3CUFAX4H : will test, thanks <@U3SJEDR96>  
 U3CUFAX4H : I'll give it a test and get back to you  
 U3SJEDR96 : `toCode` gives the UTF-16 code unit as a decimal  
 U3CUFAX4H : I need a replicatable key in elm and in rails so that I don't have to pass keys around  
 U3CUFAX4H : perfect  
 U3SJEDR96 : `elm-char-codepoint` gives that as a utf-32 decimal, basically  
 U3SJEDR96 : (oddly, I just implemented `utf-16` & `utf-8` which involves a side-step through utf-32. In elm. It was painful.)  
 U3CUFAX4H : your pointer is perfect for my use, thank you. I wasn't aware the keyCodes were exactly the same operation that I was looking for.  
 U3LUC6SNS : <@U3SJEDR96>, Yes, `delayedCommit` works quite nicely! Thanks again.  
 U5P4FLYLE : Hi, I have two components: innerCard and outerCard. In outerCard I have:``  
 type Msg = InnerCardMsg InnerCard.Msg  
 | SomeMsg``  
 And I call in outerCard :  
 `` somefunctionToHtml outerCard  
 |&gt; Html.map InnerCardMsg``  
 The problem: innerCard has of course its own events, but outerCard has one represented by SomeMsg. If I delegate through `somefunctionToHtml` I have compile error that I am expecting Html InnerCard.Msg but received Html Msg (from outerCard). How to solve this kind of problem?  
 ie. Allow within innerCard not only eject its own InnerCard.Msg messages but also the ones from outerCard?

U3LUC6SNS : re Parsing again. I currently have``  
 type alias ARG =  
 { value : String }

```
arg : Parser ARG
arg =
  succeed ARG
  |. symbol "{"
  |= keep zeroOrMore (\c -&gt; c /= '}')
  |. symbol "}"
...
```

I'd prefer to have  
 ...

```
type alias ARG = String
...
```

However, this gives an error:  
 ...

```
-- NAMING ERROR ----- ./src/LatexParser/Latex.elm
Cannot find variable `ARG`
87|   succeed ARG
...
```

Is there a way to have `String`?

U3SJEDR96 : You lose the constructor function when not using a record  
 U3SJEDR96 : however, you sure can  
 U3SJEDR96 : `succeed identity`  
 U3SJEDR96 : i.e. `Arg` is a `String` & `Arg` function, and you'd rather have a `String` & `String` function so you end up with a `String`  
 U3SJEDR96 : `identity` fits the bill :wink:  
 U663M2MB7 : is there any syntax for { model | x = y } with nested models? I guess the Focus library is one, but is there anything I am missing here?  
 U0JL9RPC4 : I'm afraid not  
 U0FP80EKB : I've generally found that when I need to update nested models, I'm looking at some sort of abstraction that wants to come out with some helper functions around it. Not always, but a good indication. In a way, you can look at updating nested records as a good time to think about how the law of demeter applies in your case.  
 U3LUC6SNS : <@U3SJEDR96>, thanks again! Got it working like I want with `identity`