

U5SJJ85B : I tried onChange
 U5SJJ85B : (coming from React)
 U5SJJ85B : actually
 U5SJJ85B : thats the answer i guess
 U5SJJ85B : React provides the value attribute as a convenience
 U5SJJ85B : but i guess here id have to do it on the option
 U0LPMPL2U : If you were hard coding HTML, you'd writ something like:``
 <select>
 <option value="1">1</option>
 <option value="2", selected="selected">2</option>
 </select>
 ...

U5SJJ85B : Thanks for the direction!
 U0LPMPL2U : adding `selected="selected"` to an HTML option makes it the pre-selected option in a `<select>`
 U0LPMPL2U : You'd do the same thing in Elm
 U5SJJ85B : ``import Html exposing (..)
 import Html.Attributes exposing (..)
 main =
 select [value "4"]
 (List.range 1 100
 |> List.map (\n -> option [value <| toString <| n, selected (n == 4)] [text <| toString <| n]))
 ...

U5SJJ85B : thanks!
 U0JL9RPC4 : Is it possible somehow to define a "set" of union types? For instance:
 ``Haskell
 type Foo = Val1 | Val2 | Val3
 type alias SetOfFoo = ??
 ...

U0JL9RPC4 : `Set` only accepts comparable values
 U0LPMPL2U : yes, unfortunately union types aren't comparable (for now) and can't be put into a set or used as keys in `Dict`s
 U0JL9RPC4 : well, `Dict` are fine, thanks!
 U5J0H2NS2 : same thing happens when you use elm-mdl and Layout...
 U4872964V : there is <http://package.elm-lang.org/packages/truqu/elm-dictset/latest> and
 <http://package.elm-lang.org/packages/eeue56/elm-all-dict/latest> (which you can turn into a set by storing `()` or something
 U4872964V : (Elms `Set` is basically a `Dict` comparable ()) I think)
 U0K7EBT3J : does anybody know an easy way or a lib to strip away html markup from a string in elm?
 `click here` -> "click here"

U23SA861Y : strip away or escape?
 U0K7EBT3J : remove