

```

— Austin Clyde
— Lecture 4
import Prelude hiding (foldr, product, map, lookup)

— 4.2*
product :: Num a => [a] -> a
product = foldr (*) 1
— *Main> product [0]
— 0
— *Main> product [1, 2, 3]
— 6

map :: (a -> b) -> [a] -> [b]
map f = foldr ((:) . f) []
— *Main> map (++[3]) [[1, 2],[3, 4],[5, 6]]
— [[1,2,3],[3,4,3],[5,6,3]]
— *Main> map (+1) [1, 2, 3, 4]
— [2,3,4,5]

--4.4*
type Bindings = [(String, Int)]

lookup :: String -> Bindings -> Int
lookup - [] = error "value: key is not bound."
lookup key ((k, v) : bindings)
  | k == key = v
  | otherwise = lookup key bindings

bind :: String -> Int -> Bindings -> Bindings
bind key value [] = (key, value) : []
bind key value ((k, v) : bindings)
  | k == key = (key, value) : bindings
  | otherwise = (k, v) : (bind key value bindings)

— *Main> let ages = [("Susan", 12), ("Austin", 19)]
— *Main> lookup "Susan" ages
— 12
— *Main> lookup "Austin" ages
— 19
— *Main> bind "Yaju" 21 ages
— [("Susan",12),("Austin",19),("Yaju",21)]
— *Main> bind "Joe" 19 []
— [("Joe",19)]

```