```
fun append (xs,ys) =
    if xs=[]
    then ys
    else (hd xs)::append(tl xs,ys)

fun map (f,xs) =
    case xs of
        [] => []
      | x::xs' => (f x)::(map(f,xs'))

val a = map (increment, [4,8,12,16])
val b = map (hd, [[8,6],[7,5],[3,0,9]])
```

Programming Languages

Dan Grossman

Polymorphic Examples

Key Idea

- Collect all the facts needed for type-checking

- These facts constrain the type of the function

- This segment:
  - Examples with type variables
  - Happens when constraints do not require particular types (but some types may still need to be the same as each other)

- See the code file and/or the reading notes