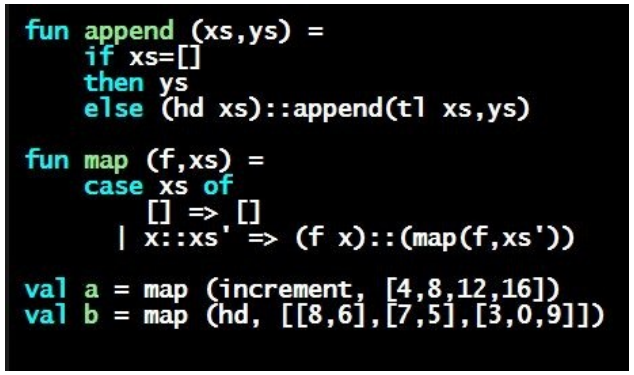

```
fun append (xs,ys) =
    if xs=[]
    then ys
    else (hd xs)::append(tl xs,ys)

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
2013

*Type Inference Examples*

# *Key Idea*

- Collect all the facts needed for type-checking
- These facts constrain the type of the function
- This segment:
  - Two examples without type variables
  - And one example that does not type-check
- See the code file and/or the reading notes