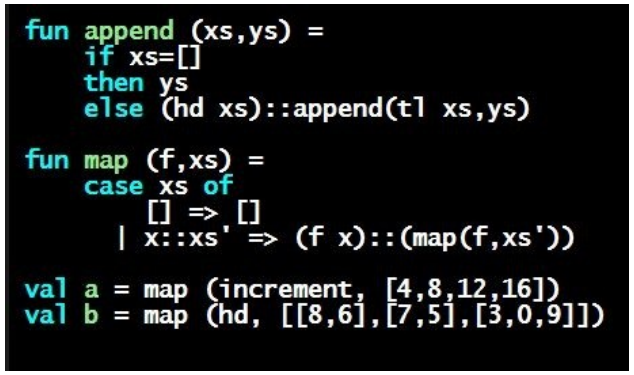

```
fun append (xs,ys) =
    if xs=[]
    then ys
    else (hd xs)::append(tl xs,ys)

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

A Longer Example

Now

- Put together much of what we have learned to define and use a small class for rational numbers
  - Called **`MyRational`** because Ruby 1.9 has great built-in support for fractions using a class **`Rational`**

- Will also use several new and useful expression forms
  - Ruby is too big to show everything; see the documentation

- Way our class works: Keeps fractions in reduced form with a positive denominator
  - Like an ML-module example earlier in course