

```
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

Arrays

Ruby Arrays

- Lots of special syntax and many provided methods for the **Array** class
- Can hold any number of other objects, *indexed* by number
 - Get via **a[i]**
 - Set via **a[i] = e**
- Compared to arrays in many other languages
 - More flexible and dynamic
 - Fewer operations are errors
 - Less efficient
- “The standard collection” (like lists were in ML and Racket)

Using Arrays

- See many examples, some demonstrated here
- Consult the documentation/tutorials
 - If seems sensible and general, probably a method for it
- Arrays make good tuples, lists, stacks, queues, sets, ...
- Iterating over arrays typically done with methods taking blocks
 - Next topic...