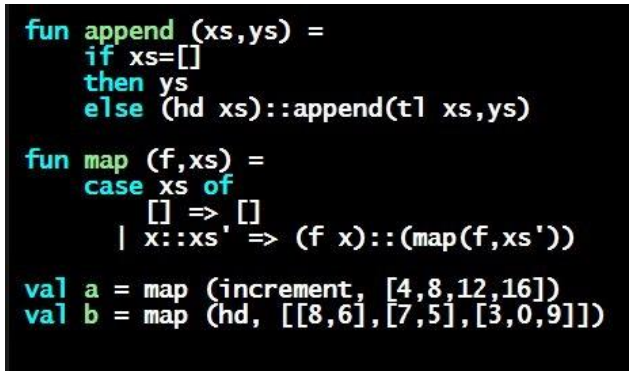

```
fun append (xs,ys) =
    if xs=[]
    then ys
    else (hd xs)::append(tl xs,ys)

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
University of Washington

Why Part A, Part B, Part C

One Course in 3 Parts

- 100-200 hours of amazing stuff

- "My novel" that I can't bear to think of broken up
  - Cross-cutting themes
  - Neat things that tie together later
  - Key contrasts that come only at the end
  - A lot of thought in what to include and what not to

- I *really* want you to do all 3 parts
  - (Certificates priced accordingly)
  - When I say "the course", I usually mean everything

- So why break it up…

Reasons for parts

- Overall, much more material and more challenging than “most MOOCs”
- Give you tough-but-intermediate goals that deserve recognition
- Give you a chance to “rest” before proceeding
  - Closer approximation to “different paces”

How it breaks down

- Unfortunately “by language” when that’s not the point

- Part A is roughly as much as Part B and Part C combined
  - No natural place to sub-divide Part A in my opinion

- Part A: 3 homeworks, 1 exam
- Part B: 2 homeworks (1 more challenging)
- Part C: 2 homeworks (1 more challenging), 1 exam