```
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

Standard-Library Documentation

One last thing

This topic is not particularly related to the rest of the section, but we made it a small part of Homework 3

ML, like many languages, has a standard library

- For things you could not implement on your own
  - Examples: Opening a file, setting a timer
- For things so common, a standard definition is appropriate
  - Examples: **`List.map`**, string concatentation

You should get comfortable seeking out documentation and gaining intuition on where to look

- Rather than always being told exactly what functions do

Where to look

`http://www.standardml.org/Basis/manpages.html`

Organized into structures, which have signatures

- Define our own structures and signatures next section

Homework 3: Find-and-use or read-about-and-use a few functions under **`STRING`**, **`Char`**, **`List`**, and **`ListPair`**

To use a binding: **`StructureName.functionName`**

- Examples: **`List.map`**, **`String.isSubstring`**

REPL trick

- I often forget the order of function arguments

- While no substitute for full documentation, you can use the REPL for a quick reminder
  - Just type in the function name for its type
  - Can also guess function names or print whole structure
  - No special support: this is just what the REPL does

- Some REPLs (for other languages) have special support for printing documentation