

Chapter 1 & 2 outline:

- ▶ Introduction, sets and elements (Wednesday, Aug 24)
- ▶ Set operations; visual verification of set propositions (Friday, Aug 26)
- ▶ Introduction to SML; cardinality and Cartesian products (last week Monday)
- ▶ Making types in SML (last week Wednesday)
- ▶ Functions in SML (last week Friday)
- ▶ Lists and functions on lists (**Today**)
- ▶ Powersets; a language processor (next week Friday)

Today:

- ▶ Review of functions
- ▶ Principles of lists
- ▶ Type analysis of lists
- ▶ Functions on lists

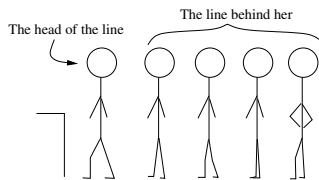
1. Lists must have at least one item.

2. All elements in a list must have the same type.

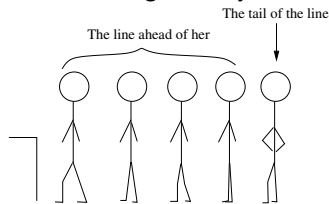
3. Lists can have tuples in them

4. Tuples can have lists in them.

5. This is a good way to think of lists:



6. This is a good way to think of lists:



`[t1([5, 12, 6])@[8, 9]]`

`hd([12, 5, 6]) :: [2, 7]`

$[(2.3, 5), (8.1, 6)], []]$

`([1, 12, 81], ["a", "bc"])`

For next time:

If you had trouble on the programming problems from last time, ask for help and try again.

Pg 70: 2.1.(2-4, 9, 10) [on “paper”, electronically through Schoology]

Pg 74: 2.2.(2, 3, 8, 9) [through turn-in page]

See notes on Ex 2.2.8 and 2.2.9 on the Schoology description of the assignment for clarifications and hints. See also the code from class for “starter code.”

*You do **not** need to include your SML code with your on-paper problems that you turn in.*

Read 2.(4 & 5)