

# Typing (con't)

- Common Lisp allows deferred typing
  - types can be added later to generate better code and to tighten up the design
- ML-like languages infer typing
- Haskell-like languages attempt to do all type checking statically,
  - and, make flexibility/language trade-offs to achieve static analysis

# Nesting (Scoping)

- I don't want to discard any data, since I expect that I will change my plan
- I use a mix of .CSV files (downloaded from the bank) and manual entry (data that I forgot to download and can't access) to create my data and check it against paper and .PDF statements
- Some data is too repetitive to be entered manually, reliably
- I will *nest* certain slowly changing information, e.g. *statement dates* (there are many transactions in each statement) and *year* and *from-account (debit/credit)*