Introduction to Scheme

- Designed to support the functional paradigm (but has procedural features)
- “Easily” extendable to support other paradigms (e.g., the object-oriented paradigm)
- A relatively small programming language (unlike, say, Java)
- Not a toy: used in industry, AI, and academia
- Watch out for fanatics (both pro and con)

A Brief Scheme Timeline

- 1956: FLPL (FORTRAN List-Processing Library)
- 1959: LISP (LISt Processing)
- 1962: LISP 1.5
- 1975: Scheme
- 1984: Common LISP
- 1986: R3RS (Revised Revised Revised Report on the Algorithmic Language Scheme)

A Brief Scheme Timeline

- 1990: IEEE Scheme
- 1991: R4RS (Revised Revised Revised Revised Report on the Algorithmic Language Scheme)
- 1998: R5RS (Revised^{5} Report on the Algorithmic Language Scheme)
- 2007: R6RS (Revised^{6} Report on the Algorithmic Language Scheme)
- 2013 (partially): R7RS (Revised^{7} Report on the Algorithmic Language Scheme)

A Scheme Implementation Sampler

- Armpit Scheme
- Bigloo
- Chez Scheme
- Guile
- MIT Scheme
- Pixie Scheme
- Racket
- Many others: see `www.schemers.org`

A Scheme Implementation Sampler

- Armpit Scheme
- Bigloo
- Chez Scheme
- Guile
- MIT Scheme
- Pixie Scheme
- Racket *The impl'n we'll be using*
- Many others: see `www.schemers.org`

A Scheme IDE Sampler

- Bee
- DrRacket
- fluxus
- Others: see `www.schemers.org`

Note

- Racket is an extension of Scheme with added features, libraries, etc.
- We'll (almost exclusively) be sticking to standard R6RS Scheme