

## Top 10 keybindings in Emacs/Edwin for MIT Scheme

These are keybindings specific to using MIT Scheme. I *STRONGLY* recommend that you spend some time learning how to use the emacs editor itself, i.e. learn the basic keybindings for moving the cursor around, switching buffers, killing and yanking text, etc. The time you spend will be more than repaid in your productivity.

1. `TAB` properly indents your scheme code  
Type a line of code. Hit return. Hit `TAB`. Type the next line of code. Repeat.  
OR, put the cursor at any point on a line of improperly indented code. Hit `TAB`, and the line will be indented the proper amount (relative to the lines above it, so start at the top if you need to do multiple lines, or use `C-M-q`).
2. `C-x C-e` sends the expression to the left of the cursor. This works both in the Scheme interaction buffer as well as in `scheme-mode` buffers where you are editing your code.  
This command looks left from the cursor until it has a complete expression and then sends that to the Scheme interpreter.
3. `M-z` sends the current expression to the Scheme interpreter. This works both in the Scheme interaction buffer as well as in `scheme-mode` buffers where you are editing your code.  
This command looks for an open parenthesis in the leftmost column and sends all the text from there up to the cursor position, and then to the right of the cursor until all parentheses are balanced.
4. `C-c C-s` when done in a `scheme-mode` buffer, will switch to the Scheme interaction buffer.
5. `M-p` and `M-n` in the Scheme interaction buffer will scroll through the history of expressions that you have sent to the Scheme interpreter. (This doesn't work if you are using version 7.3. You can use `C-c C-y` to yank the previous command.)
6. `C-c C-c` sends an interrupt to the Scheme interpreter.
7. `M-o` from a `scheme-mode` buffer, will send the entire buffer to the Scheme interpreter.
8. `C-M-f` and `C-M-b` move the cursor over the previous and next (respectively) parenthesis balanced list. (`C-M-p` and `C-M-n` perform the same functions.)
9. `C-M-a` and `C-M-e` move the cursor to the beginning and end of a definition, respectively.  
Your code should be indented properly for these to have the desired effect.
10. `C-M-q` properly indents an entire definition.  
The cursor should be on the opening parenthesis of the define for this to work properly.