

# SLIME Quick Reference Card

## Getting help in Emacs

**C-h** **<key>** describe function bound to **<key>**  
**C-h b** list the current key-bindings for the focus buffer  
**C-h m** describe mode  
**C-h l** shows the keys you have pressed  
**<key> l** what starts with **<key>**

## Programming

### Completion

**M-tab**, **C-c C-i**, **C-M-i** complete symbol  
**C-c C-s** complete form  
**C-c M-i** fuzzy complete symbol

### Closure

**C-c C-q** close parens at point  
**C-]** close all sexp

### Indentation

**C-c M-q** reindent defun  
**C-M-q** indent sexp

### Documentation

**spc** insert a space, display argument list  
**C-c C-d d** describe symbol  
**C-c C-f** describe function  
**C-c C-d a** apropos search for regexp  
**C-c C-d z** apropos with internal symbols  
**C-c C-d p** apropos in package  
**C-c C-d h** hyperspec lookup  
**C-c C-d ~** format character hyperspec lookup

### Cross reference

**C-c C-w c** show function callers  
**C-c C-w r** show references to global variable  
**C-c C-w b** show bindings of a global variable  
**C-c C-w s** show assignments to a global variable  
**C-c C-w m** show expansions of a macro  
**C-c <** list callers of a function  
**C-c >** list callees of a function

### Finding definitions

**M-.** edit definition  
**M-**, or **M-\*** pop definition stack  
**C-x 4 .** edit definition in other window  
**C-x 5 .** edit definition in other frame

## Macro expansion commands

**C-c C-m** or **C-c RET** macroexpand-1  
**C-c M-m** macroexpand-all  
**C-c C-t** toggle tracing of the function at point

## Disassembly

**C-c M-d** disassemble function definition

## Compilation

**C-c C-c** compile defun  
**C-c C-y** call defun  
**C-c C-k** compile and load file  
**C-c M-k** compile file  
**C-c C-l** load file  
**C-c C-z** switch to output buffer  
**M-n** next note  
**M-p** previous note  
**C-c M-c** remove notes

## Evaluation

**C-M-x** eval defun  
**C-x C-e** eval last expression  
**C-c C-p** eval & pretty print last expression  
**C-c C-r** eval region  
**C-x M-e** eval last expression, display output  
**C-c :** interactive eval  
**C-c E** edit value  
**C-c C-u** undefine function

## Abort/Recovery

**C-c C-b** interrupt (send SIGINT)  
**C-c ~** sync the current package and working directory  
**C-c M-p** set package in REPL

## Inspector

**C-c I** inspect (from minibuffer)  
**ret** operate on point  
**d** describe  
**l** pop  
**n** next  
**q** quit  
**M-ret** copy down