### Motion

<table>
<thead>
<tr>
<th>Cursor motion:</th>
<th>backward</th>
<th>forward</th>
</tr>
</thead>
<tbody>
<tr>
<td>entity to move over</td>
<td>C-b</td>
<td>C-f</td>
</tr>
<tr>
<td>character</td>
<td>M-b</td>
<td>M-f</td>
</tr>
<tr>
<td>word</td>
<td>C=p</td>
<td>C=e</td>
</tr>
<tr>
<td>line</td>
<td>M-a</td>
<td>M-e</td>
</tr>
<tr>
<td>go to line beginning (or end)</td>
<td>M-[]</td>
<td>M-[]</td>
</tr>
<tr>
<td>sentence</td>
<td>C-x [</td>
<td>C-x ]</td>
</tr>
<tr>
<td>paragraph</td>
<td>C-x [</td>
<td>C-x ]</td>
</tr>
<tr>
<td>page</td>
<td>C-x [</td>
<td>C-x ]</td>
</tr>
<tr>
<td>sexp</td>
<td>C-x DEL M-</td>
<td>M-k</td>
</tr>
<tr>
<td>function</td>
<td>M-- C-M-h M-</td>
<td>M-k</td>
</tr>
<tr>
<td>go to buffer beginning (or end)</td>
<td>M&lt;&lt;</td>
<td>M&gt;&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screen motion:</th>
<th>v</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>scroll to next screen</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>scroll to previous screen</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>scroll left</td>
<td>v</td>
<td>v</td>
</tr>
<tr>
<td>scroll right</td>
<td>v</td>
<td>v</td>
</tr>
</tbody>
</table>

### Killing and Deleting

<table>
<thead>
<tr>
<th>entity to kill</th>
<th>backward</th>
<th>forward</th>
</tr>
</thead>
<tbody>
<tr>
<td>character (delete, not kill)</td>
<td>DEL</td>
<td>C-d</td>
</tr>
<tr>
<td>word</td>
<td>M-DEL M-d</td>
<td></td>
</tr>
<tr>
<td>line (to end of)</td>
<td>C-x DEL M-k</td>
<td></td>
</tr>
<tr>
<td>sentence</td>
<td>M-- C-M-h M-</td>
<td>M-k</td>
</tr>
<tr>
<td>sexp</td>
<td>M-x DEL M-</td>
<td>M-k</td>
</tr>
<tr>
<td>kill region</td>
<td>C-w</td>
<td></td>
</tr>
<tr>
<td>kill to next occurrence of char</td>
<td>M-z char</td>
<td></td>
</tr>
<tr>
<td>yank back last thing killed</td>
<td>C-v</td>
<td></td>
</tr>
<tr>
<td>replace last yank with previous kill</td>
<td>M-y</td>
<td></td>
</tr>
</tbody>
</table>

### Marking

<table>
<thead>
<tr>
<th>set mark here</th>
<th>C-s or C-SPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>exchange point and mark</td>
<td>C-x C-s</td>
</tr>
<tr>
<td>set mark sly words away</td>
<td>M-x</td>
</tr>
<tr>
<td>mark paragraph</td>
<td>M-h</td>
</tr>
<tr>
<td>mark page</td>
<td>C-x C-p</td>
</tr>
<tr>
<td>mark sexp</td>
<td>M-x C-e</td>
</tr>
<tr>
<td>mark function</td>
<td>C-M-e</td>
</tr>
<tr>
<td>mark entire buffer</td>
<td>C-M-h</td>
</tr>
</tbody>
</table>

### Query Replace

| interactively replace a text string | M-% |
| using regular expressions | M-x query-replace-regexp |

Valid responses in query-replace mode are:
- replace this one, go on to next | SPC |
- replace this one, don't move | DEL |
- skip to next without replacing | ! |
- replace all remaining matches | M-% |
- back up to the previous match | ! |
- exit query-replace | ESC |
- enter recursive edit (C-M-c to exit) | C-r |

### Multiple Windows

- delete all other windows | C-x 1 |
- delete this window | C-x 0 |
- split window in 2 vertically | C-x 2 |
- split window in 2 horizontally | C-x 5 |
- scroll other window | C-0 |
- switch cursor to another window | M-x shrink-window |
- shrink window shorter | C-x 1 |
- grow window taller | C-x 0 |
- shrink window narrower | C-x ( |
- grow window wider | C-x ) |
- select a buffer in other window | C-x 4 b |
- find file in other window | C-x 4 f |
- compose mail in other window | C-x 4 m |
- run Direct in other window | C-x 4 d |
- find tag in other window | C-x 4 |

### Formatting

- indent current line (mode-dependent) | TAB |
- indent region (mode-dependent) | C-M-
- indent sexp (mode-dependent) | C-M-q |
- indent region rigidly arg columns | C-M-TAB |
- insert newline after point | C-x 8 |
- move rest of line vertically down | C-x 6 |
- delete blank lines around point | C-x 0 |
- delete all whitespace around point | M-x 9 |
- put exactly one space at point | M-x SP |
- fill paragraph | M-q |
- fill region | M-g |
- set fill column | C-x f |
- set prefix each line starts with | C-x 0 |

### Case Change

- uppercase word | M-u |
- lowercase word | M-l |
- capitalize word | M-c |
- uppercase region | C-x 0 U |
- lowercase region | C-x 0 L |
- capitalize region | M-x capitalize=region |

### The Minibuffer

The following keys are defined in the minibuffer,
- complete as much as possible | TAB |
- complete up to one word | SPC |
- complete and execute | RET |
- show possible completions | ? |
- abort command | C-g |

Type C-x ESC to edit and repeat the last command that used
the minibuffer. The following keys are then defined,
- previous minibuffer command | N-p |
- next minibuffer command | N-n |
Regular Expressions

The following have special meaning inside a regular expression:

- any single character
- zero or more repetitions
- one or more repetitions
- zero or one repetition
- any character in set
- beginning of line
- end of line
- quote a special character
- alternative ("or") grouping
- with group
- beginning of buffer
- end of buffer
- word break
- not beginning or end of word
- beginning of word
- end of word
- any word-syntax character
- any non-word-syntax character
- character with syntax c
- character with syntax not c

Registers

- copy region to register
- insert register contents
- save point in register
- move point to saved location

Info

- enter the Info documentation reader

Moving within a node:

- scroll forward
- scroll reverse
- beginning of node
- previous node
- move up
- select menu item by name
- select with menu item by number (1-5)
- follow cross reference (return with f)
- return to last node you saw
- return to directory node
- go to any node by name

Moving between nodes:

- next node
- previous node
- move up
- select menu item by name
- select with menu item by number (1-5)
- follow cross reference (return with f)
- return to last node you saw
- return to directory node
- go to any node by name

Other:

- run Info tutorial
- list Info commands
- quit Info
- search nodes for regexp

Keyboard Macros

- start defining a keyboard macro
- end keyboard macro definition
- execute last-defined keyboard macro
- append to last keyboard macro
- name last keyboard macro
- insert user-defined macro
- insert last-keyboard macro

Commands Dealing with Emacs Lisp

- eval sexp before point
- eval current defun
- eval region
- eval entire buffer
- eval all buffer
- execute last minibuffer command
- read and eval minibuffer command
- read and eval Emacs Lisp file
- load from standard directory
- load library

Simple Customization

Here are some examples of binding global keys in Emacs Lisp. Note that you cannot say "M-S-"; you must say "M-".

- (global-set-key "\c-c\c-g" 'goto-line)
- (global-set-key "\c-o\c-r" 'isearch-backward-regexp)
- (global-set-key "\c-x \c-r" 'query-replace-regexp)

An example of setting a variable in Emacs Lisp:

- (setq backup-by-copying "when-linked")

Writing Commands

- (defun (command-name) (args)
- "(documentation"
- "interactive ((template))"
- "(body))"

An example:

- (defun this-line-is-top-of-screen (line)
- "Repainted line point is on to the top of the screen. With ARG, put point on line ARG.
- Negative counts from bottom."
- (interactive "p")
- (recenter (if (null line)
- " (prefix-numeric-value line))"

The argument to interactive is a string specifying how to get the arguments when the function is called interactively. Type C-h f interactive for more information.

Copyright © 1987 Free Software Foundation, Inc.

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all copies.

For copies of the GNU Emacs manual, write to the Free Software Foundation, Inc., 675 Massachusetts Ave, Cambridge MA 02139.