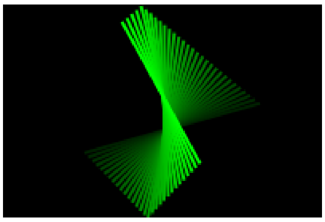
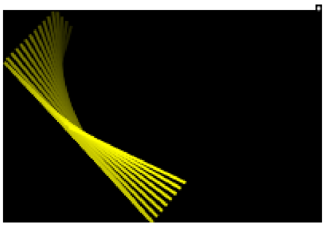
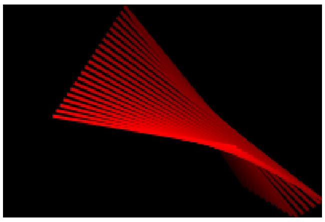
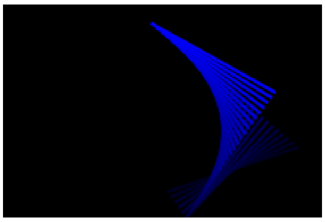


JS
Perl Home Knots
Family Emacs







Emacs Quick Reference

This reference will help in understanding some of the basic and a few of the more useful commands in Emacs. It should at the very least help you to exit.

Notation

It is important to understand the notation commonly used in Emacs documentation. Most of the commands used in Emacs consist of a modifier key, in conjuction with one or more other keys. The following are the common modifier keys:

CTRL	- (C) -	The CONTROL key.
META	- (M) -	Depending upon the terminal this is the ALT key. You may also use the ESC key to send META.
ESC	- (ESC) -	The ESCAPE key.
SHIFT	- (S) -	The SHIFT key.

In Emacs documentation it is common to use an abbreviated syntax when describing key sequences. Instead of typing

```
CTRL-x CTRL-c
```

You would type

```
C-x C-c
```

This would represent holding down the CONTROL key and pressing the letter x, and then holding down the CONTROL key and pressing the letter c. (M-C-\ would represent holding down the the ALT key and CONTROL key and pressing the "\" key. You could also press and release the ESCAPE key hold down the CONTROL key and type the "\" key.)

The following is a table of notation for other keyboard characters.

BACKSPC	The BACKSPACE key.
SPC	The SPACE bar.
ENTER	The Enter key.
RET	The Enter key.
TAB	The TAB key.

Insert Mode?

This is a large place where Emacs differs from the Vi editor. In Vi you are either in Insert Mode or you are not. In order to execute commands other than key insertion you need to be out of Insertion Mode. By using CTRL to escape the key sequences, Emacs allows you to use the commands at any point in your session. For example, if you are in Vi and are inserting text you would have to type the following key sequence to save your file and return to Insert Mode:

```
ESC : w i
```

In Emacs (which is essentially always in "Insert Mode") you would type the following:

```
C-x C-s
```

When viewed from the insertion mode view point, Vi doesn't really save any keystrokes over Emacs as is commonly claimed by religious Vi advocates. (It actually requires more in most cases - the debate is really whether you're in Insert Mode more than you are in Command Mode.)

What is really happening on the backend

Emacs is really a bunch of lisp routines programmed to operate on the text in the buffers. These complex routines are accessed via key bindings on the front end. For example, when you type an "s" you are really telling Emacs to run the "self-insert-command" on the letter "s" which inserts the letter "s" at the current Point of Insertion (POI). These backend commands can be accessed by typing M-x and then the command. For example, you open a file with the key binding "C-x C-f". You can begin the same operation by typing "M-x find-file".

Keyboard Shortcuts - (Key Bindings)

Key Binding	Backend Function	What it does
		Quiting Things
C-x C-c	M-x save-buffers-kill-emacs	**Quit Emacs**. Exits out of Emacs. If you have open buffers that have been modified you are prompted to save the changes or discard them.
C-g	M-x keyboard-quit	If you are in a key request sequence that you can't get out of, this returns you to the main buffer.
ESC ESC ESC	M-x keyboard-escape-quit	Does about the same thing as "C-g"
C-x k	M-x kill-buffer	Kills the current buffer. If the buffer is modified you are prompted to save it.
		Files and directories
C-x C-f	M-x find-file	**Open a file**. This prompts for the name of a file. Once you have typed the file it opens it in the current buffer. If the file doesn't exist, a blank buffer is created and Emacs allows you to begin typing (In this case the file will not exist on the system until you save it the first time). When you are typing the name of the file, you may use the tab key for filename completion. Also you may use M-p and M-n to go to the "previous" and "next" files that have been found using find-file.
C-x C-s	M-x save-buffer	**Save a file**. This saves the current buffer.
C-x d	M-x dired (directory edit)	More Here. This opens the directory specified (same way as find-file). It opens a buffer that lists the directory and allows you to perform operations on the files. listed in that directory. There are many features (to many to list here) that I will go into later. The most basic is to put the cursor on the file you want and hit ENTER. This does a find-file on that file.
C-x C-w	M-x write-file	Save the current file as...
C-x i	M-x insert-file	Insert a file
		Buffers
C-x C-b	M-x list-buffers	This lists all buffers that are open. Emacs will typically open a buffer for each file that you visit (if you visit a new file without closing an old one). This will give you a list of all of the currently open buffers and will allow you to choose one from the list to switch to. (You will notice there are a couple of buffers beginning with an *. These are buffers used by Emacs to store output. You may kill them if you like.) If you have a file opened multiple times, you will see it listed as "file" and "file <2>". This list of buffers is called the **buffer ring**
C-x b	M-x switch-to-buffer	**Switch buffers**. Does just that. The default buffer to switch to is the last one that you came from. (Typing C-x b ENTER repeatedly swaps between the current and the last buffer like ALT-TAB in windows.) You may type a TAB to complete the buffer name or TAB TAB to see a list of all of the buffer names you can type.

C-x k	M-x kill-buffer	Kills the current buffer. If the buffer is modified you are prompted to save it.
Windows and frames		
C-x 0	M-x delete-window	Kills the current window if more than one is visible. The buffer contained in the window is put back to the buffer ring.
C-x 1	M-x delete-other-windows	If more than one window is visible in the current viewing frame, it removes all but the current window. Other buffers remain open but unviewed.
C-x 2	M-x split-window-vertically	Splits the current viewing window into two windows placed one over the other.
C-x 3	M-x split-window-horizontally	Splits the current viewing window into two windows placed side to side.
C-x o	M-x other-window	Switches between two windows.
Moving Around		
C-p (up arrow)	M-x previous-line	Moves to the previous line.
C-n (down arrow)	M-x next-line	Moves to the next line.
C-f (right arrow)	M-x forward-char	Move forward one character
C-f (left arrow)	M-x backward-char	Move backward one character
M-f	M-x forward-word	Move forward one word.
M-b	M-x backward-word	Move backward one word.
C-a	M-x beginning-of-line	Move to the beginning of the line.
C-e	M-x end-of-line	Move to the beginning of the line.
M-a	M-x backward-sentence	Move backward one sentence.
M-e	M-x forward-sentence	Move forward one sentence.
C-v	M-x scroll-up	Page down one page.
M-v	M-x scroll-down	Page up one page.
M->	M-x end-of-buffer	Moves to the end of the buffer.
M-<	M-x beginning-of-buffer	Moves to the beginning of the buffer.
C-x]	M-x forward-page	In most modes does the same as M->
C-x [	M-x backward-page	In most modes does the same as M-<
	M-x goto-line	**Goto line**. Does what it says. I can't believe that there is not a default key binding for this (there is in XEmacs). I usually set up in my .emacs file the line (global-set-key "\M-g" 'goto-line) which tells emacs to use M-g to run the command.
C-x o	M-x other-window	Switches between two windows.
C-x b	M-x switch-to-buffer	**Switch buffers**. Does just that. The default buffer to switch to is the last one that you came from. (Typing C-x b ENTER repeatedly swaps between the current and the last buffer like ALT-TAB in windows.) You may type a TAB to complete the buffer name or TAB TAB to see a list of all of the buffer names you can type.
C-SPC	M-x set-mark-command	**Set Mark**. Sets the mark to the current cursor location (point). Many commands are run on the region in between the point and mark.
C-x C-x	M-x exchange-point-and-mark	Exchanges the point (cursor location) and the mark.
Copying and Deleting stuff		

C-_	M-x undo	**Undo**. Undoes the last command. Can be done repeatedly.
C-d	M-x delete-char	**Delete**. Deletes the character under the point (after the cursor location).
BACKSPC	M-x delete-backward-char	**Delete**. Deletes the character before the point (before the cursor location).
M-d	M-x kill-word	Kills the word after the cursor location. Places on the kill ring.
M-BACKSPC	M-x backward-kill-word	Kills the word before the cursor location. Places on the kill ring.
C-k	M-x kill-line	**Kill**. Kills from the point to the end of the line. If repeated in a row, all of kills go into the same kill ring entry. Doing this once does not kill the new line at the end of the line. Repeating kills the newline. (for example, to kill the entire current line you would type C-a to move to the beginning of the line, and C-k C-k to kill the line and the newline).
C-y	M-x yank	**Yank**. Inserts the last entry from the kill ring. This can be done repeatedly.
M-y	M-x yank-pop	Cycle through the kill ring. Operates only if the last command was C-y
C-SPC	M-x set-mark-command	**Set Mark**. Sets the mark to the current cursor location (point). Many commands are run on the region in between the point and mark.
C-w	M-x kill-region	Kills the text between the mark and the point. Places on the kill ring. If the buffer is read only, places text on kill ring but warns about not being able to modify buffer.
M-w	M-x kill-ring-save	Places text between mark and point on the kill ring without killing text.
M-z	M-x zap-to-char	Deletes the text from the point up to the first instance of the specifed character
Search and Replace		
C-s	M-x isearch-forward	Searches forward for the typed in text. C-s repeatedly searches forward multiple times. C-r switches direction of search.
C-r	M-x isearch-backward	Searches backward for the typed in text. C-r repeatedly searches backward multiple times. C-s switches direction of search.
M-%	M-x query-replace	Requests two patterns. Searches for first pattern and replaces with second pattern. Prompts at each match. (y for "yes replace", n for "no next", and ! for "replace all").
C-u C-s	M-x isearch-forward-regexp	Search using a regular expression (as you type it).
M-C-%	M-x query-replace-regexp	Query replaces using a regular expression as the match.
Fixing up the text		
TAB	M-x indent-relative-maybe	Indent the text according to the program mode you are in.
M-C-\	M-x indent-region	Runs the "TAB" on all lines in the region.
M-u	M-x upcase-word	Upper casifies from the point to the end of the word.
M-l	M-x downcase-word	Lower casifies from the point to the end of the word.
C-t	M-x transpose-chars	Swaps characters before and after point with one another.
M-t	M-x transpose-words	Swaps current and next word with one another.
Dired - Directory Editor		
C-x d	M-x dired (directory edit)	This opens the directory specified (same way as find-file). It opens a buffer that lists the directory and allows you to perform operations on the files. listed in that directory. There are many features

		(to many to list here) that I will go into later. The most basic is to put the cursor on the file you want and hit ENTER. This does a find-file on that file.

All commands in this section will only operate when you are in dired mode.	ENTER	M-x dired-advertised-find-file	Runs a C-x C-f (find-file) on the currently highlighted file. If the file is a directory, that directory will be opened in dired mode in a new buffer.
	+	M-x dired-create-directory	Prompts for a new directory name and creates it.
	d	M-x dired-flag-file-deletion	Flags the current file for deletion.
	~	M-x dired-flag-backup-files	Flags all files ending with **~** for deletion (backup files)
	x	M-x dired-do-flagged-delete	Delete all files marked for deletion (lines beginning with a **D**
	D	M-x dired-do-delete	Deletes the currently highlighted file(s)
	C	M-x dired-do-copy	Copies the currently highlighted file(s). Prompts for the destination.
	R	M-x dired-do-rename	Renames the currently highlighted file(s). Prompts for the destination.
	% R	M-x dired-do-rename-regex	Renames the currently highlighted file(s) using regular expressions. Prompts for a regex for from and to. (example - from regex **\(.+\).txt**, to regex **\1.text**)
	m	M-x dired-mark	Mark the current file (marks with a *****)
	u	M-x dired-unmark	Unmark the current file (marks with **SPC** - SPC is blank)
	* c	M-x dired-change-marks	Changes marks from first prompted letter to second prompted letter (example - *** c D SPC** - this will clear the deletion flag)
	!	M-x dired-do-shell-command	Execute command on files marked with *****. If your command contains a *****, the list of marked files will be substituted in place and executed. Otherwise, the command will be played once with each marked file as the last argument.
	g	M-x revert-buffer	Refreshes the current dired directory.

	Help - Finding it	
C-h t	M-x help-with-tutorial	Begins the Emacs built in Tutorial - highly recommended
C-h k	M-x describe-key	Allows you to type in key strokes and see the backend function that they are running.
C-h f	M-x describe-function	Allows you to type a function name and receive an explanation of what it does.
C-h v	M-x describe-variable	Allows you to type a Emacs variable name and see its contents
C-h b	M-x describe-bindings	Show a table of all of the current key bindings.
C-h i	M-x info	This one is possibly the most important. It enters into the Emacs help browser which allows you to read in depth about just about any subject relating to Emacs. (Basic commands are "l" for last, "n" for next, and "p" for previous.) I'll talk more about this later.
	Text Registers	
C-x r s	M-x copy-to-register	Prompts for register letter. Saves text between point and mark in the register.
C-x r i	M-x insert-register	Prompts for register letter. Inserts the text saved in that register at point.
	Position Registers	
C-x r SPC	M-x point-to-register	Prompts for register letter. Saves point in register.
C-x r j	M-x jump-to-register	Prompts for register letter. Jumpts to point saved in that register.
	Bookmarks	

C-x r m	M-x bookmark-set	Prompts for bookmark name (default is buffer name). Saves point in a bookmark. This information is saved even if you close Emacs.
C-x r b	M-x bookmark-jump	Prompts for bookmark name. Jumpts to point saved in that bookmark (Even if you had closed your emacs session).
C-x r l	M-x bookmark-bmenu-list	Lists all saved bookmarks. Typing "j" when on a bookmark jumps to the file. Dired commands are used to delete bookmarks (Type "d" when on file - then "x" to delete bookmark).
Rectangles - (Priceless)		
C-x r o	M-x open-rectangle	Opens rectangle of whitespace between point and mark.
C-x r t	M-x string-rectangle	Replaces text between point and mark with rectangle of text (prompts for text). This is a major timesaver. Typing C-x r t "Other Text" RET would replace <mark>Some Text Some Text Some Text<point> With <mark>Other Text Other Text Other Text<point>
C-x r k	M-x kill-rectangle	Kills text between point and mark and places in rectangle kill ring.
C-x r y	M-x yank-rectangle	Inserts last killed rectangle at point.