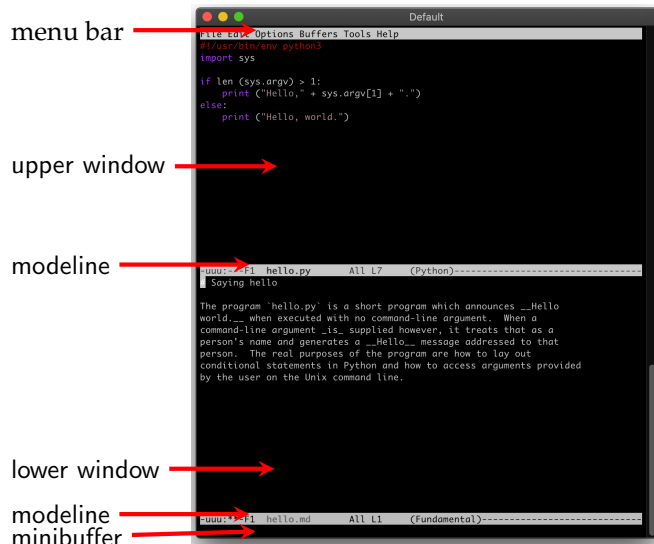


Emacs Reference Card

Major features

Cross-platform
Command and filename completion
Incremental search
Extensible and programmable
Syntax highlighting for C, Java, L^AT_EX, Markdown, Python...
Interfaces to symbolic debuggers



An Emacs frame containing two windows

Notation

hold control key, simultaneously type x C-x
hold meta key, simultaneously type x M-x
(meta is ALT on PCs and option on Macs, or you can type
e.g. ESC followed by x for M-x)

Starting Emacs

To edit file `myfile.txt` `emacs myfile.txt`
To run Emacs in its own window `emacs &`

Leaving Emacs

suspend Emacs C-z
exit Emacs permanently C-x C-c

Written by Adrian F. Clark (`alien@essex.ac.uk`) based on the Emacs reference card & > 30 years' use. [last edit: 2019/11/28]

Files

read a file into Emacs C-x C-f
save a file back to disk C-x C-s
insert contents of a file into the current buffer C-x C-i
write buffer to a specified file C-x C-w
toggle read-only status of buffer C-x C-q
view a file (read-only) C-x v
browse directory using dired C-x d
With tramp loaded and ssh available, you can edit a remote file as `/login@host:/remote/path/to/file`

Online help

get help C-h
tutorial on editing with Emacs C-h t
show commands matching a string C-h a
describe the function a key runs C-h k
describe a function C-h f
describe the current editing mode C-h m
online documentation M-x info

Error recovery

interrupt an executing command C-g
undo an unwanted change C-_
redraw screen centred on current line C-l
Type M-x `recover-session` to recover after a system crash. Type M-x `revert-buffer` to restore a buffer to its original content.

Incremental search

search forward C-s
search backward C-r
regular expression search forward C-M-s
regular expression search backward C-M-r
select previous search string M-p
select next later search string M-n
exit incremental search RET
undo effect of last-typed character DEL
abort current search C-g
Use C-s or C-r to repeat a search in the chosen direction.

Miscellaneous

numeric argument C-u *num*
negative argument M--
insert character *c* C-q *c*

Motion

entity to move over	backward	forward
character (or arrow keys)	C-b	C-f
word	M-b	M-f
line (or arrow keys)	C-p	C-n
start or end of line	C-a	C-e
sentence	M-a	M-e
paragraph	M-{	M-}
function	C-M-a	C-M-e
buffer	M-<	M->
scroll screen	M-v	C-v
scroll left/right	C-x <	C-x >

Marking

(‘point’ is the position of the cursor)

set mark at point	C-SPC
exchange point and mark	C-x C-x
mark paragraph	M-@
mark function	C-M-h
mark entire buffer	C-x h

Deleting, killing, pasting

entity to kill	backward	forward
delete character	DEL	C-d
word	M-DEL	M-d
rest of line	M-0	C-k
sentence	C-x DEL	M-k
kill region		C-w
copy region to kill ring		M-w
yank back last thing killed		C-y
replace last yank with previous kill		M-y

Change case

uppercase word	M-u
lowercase word	M-l
capitalize word	M-c
uppercase region	C-x C-u
lowercase region	C-x C-l

Transposition

transpose characters	C-t
transpose words	M-t
transpose lines	C-x C-t

Customisation

customize Emacs	M-x customize
load a library	M-x load-library
your initialization file	~/.emacs

Notes

Experienced Emacs users leave it running all the time and then edit files (C-x C-f) as desired.

Make sure program source code files have the right extension when you visit them (C-x C-f) so that Emacs switches into the right mode for the language. Visit a file that doesn't exist to create it.

To select a region of text, type C-x SPC at one extreme of it, then move the cursor to the other end and type C-w to cut the region or M-w to copy it. Paste using C-y. C-_ to undo.

On a window system, you can move the cursor and highlight regions with the mouse. If your fingers need those Windows shortcuts to copy, cut and paste, explore cua-mode.

Start a macro with C-x (, then edit a single line as desired, move to the start of the next line and finish the macro with C-x). Type C-x e to execute the macro on the next line. When you're happy that the macro is right, use C-u 100 C-x e to execute it on the next 100 lines, for example.

You can change the appearance of Emacs by selecting a theme in its customization interface.

Emacs has an extensive package system, and there is probably a package that does any unusual functionality that you want. For example, there are packages for calendar, diary, web and email. **L^AT_EX users** will find the AucTeX package very useful. You might like to read about **org mode** online.

Programmers should simply type TAB at the beginning of each line of code. If the line doesn't indent correctly, there's a syntax error on the preceding one. You can customize the way Emacs lays out code of course. Using the symbolic debugger is described in the online documentation: M-x info.

You can edit files remotely over ssh connections via tramp.

You can set up shells to accept many Emacs motion keystrokes when editing commands. A lightweight Unix editor that uses the Emacs keybindings is zile; I carry it on a memory stick.

Buffers

select another buffer	C-x b
list all buffers	C-x C-b
kill a buffer	C-x k

Windows and frames

A 'window' is a region with its own minibuffer inside an operating system window; a 'frame' is a separate operating system window (see diagram on front).

operation	window	frame
delete all others	C-x 1	C-x 5 1
split horizontally	C-x 2	C-x 5 2
delete this one	C-x 0	C-x 5 0
split vertically		C-x 3
move point to another	C-x o	C-x 5 o
find file in other	C-x 4 f	C-x 5 f

Formatting

indent current line (mode dependent)	TAB
indent region (mode dependent)	C-M-\
indent region rigidly <i>N</i> columns	C-u <i>N</i> C-x TAB
indent for comment	M-;
insert new line at point	C-o
fill paragraph	M-q
set fill width to <i>N</i>	C-u <i>N</i> C-x f

Interactive replacement

interactively replace text	M-%
valid responses while replacing are:	
replace this time	SPC or y
don't replace this time	DEL or n
replace all remaining matches	!
back up to the previous match	^
exit interactive replacement	q or RET
enter recursive edit (C-M-c to exit)	C-r

Spelling

check spelling of current word	M-\$
check spelling of whole buffer	M-x ispell-buffer
check spelling while typing	M-x flyspell-mode

Registers

save region in register	C-x r s
insert register contents into buffer	C-x r i
save value of point in register	C-x r SPC
jump to point saved in register	C-x r j

Rectangles

kill rectangle to register	C-x r k
yank rectangle	C-x r y
open rectangle, shifting text right	C-x r o
blank out rectangle	C-x r c
prefix each line with a string	C-x r t

Shells

execute a shell command	M-!
execute a shell command asynchronously	M-&
run a shell command on this region	M-
filter region through a shell command	C-u M-
start a shell in window *shell*	M-x shell
compile programs	M-x compile

Keyboard macros

start defining a keyboard macro	C-x (
finish a keyboard macro	C-x)
execute last-defined keyboard macro	C-x e

Regular expressions

any single character except a newline	.	(dot)
zero or more repeats	*	
one or more repeats	+	
zero or one repeat	?	
at word break	\b	
not at word break	\B	
quote special character <i>c</i>	\c	
alternative ("or")	\	
grouping	\(...\)	
shy grouping	\(:?...\)	
match entity		start end
line	^	\$
word	\<	\>
buffer	\'	\>