

Operation	Keystroke	Function	Note
Rectangles	The following commands operate on the rectangle area that is made of the area made of the opposite corners of the point and mark.		
Kill text in rectangle	<ul style="list-style-type: none"> C-x r k <f11> - r 	(kill-rectangle START END &optional FILL)	Delete the region-rectangle and save it as the last killed one. <ul style="list-style-type: none"> If the buffer is read-only, Emacs will beep and refrain from deleting the rectangle, but put it in 'killed-rectangle' anyway. This means that ou can use this command to copy text from a read-only buffer. (If the variable 'kill-read-only-ok' is non-nil, then this won't even beep.)
Save rectangle text	<ul style="list-style-type: none"> C-x r M-w <f11> = r 	(copy-rectangle-as-kill START END)	Copy the region-rectangle and save it as the last killed one.
Delete rectangle text	C-x r d	(delete-rectangle START END &optional FILL)	Delete (don't save) text in the region-rectangle. <ul style="list-style-type: none"> The same range of columns is deleted in each line starting with the line where the region begins and ending with the line where the region ends. With a prefix (or a FILL) argument, also fill lines where nothing has to be deleted.
Yank last killed rectangle	C-x r y	(yank-rectangle)	Yank the last killed rectangle with upper left corner at point.
Fill rectangle with space	C-x r o	(open-rectangle START END &optional FILL)	Blank out the region-rectangle, shifting text right. <ul style="list-style-type: none"> The text previously in the region is not overwritten by the blanks, but instead winds up to the right of the rectangle. With a prefix (or a FILL) argument, fill with blanks even if there is no text on the right side of the rectangle.
Insert line numbers to left or rectangle	C-x r N	(rectangle-number-lines START END START-AT &optional FORMAT)	Insert numbers in front of the region-rectangle. <ul style="list-style-type: none"> With a prefix argument, prompt for <u>START-AT</u> and <u>FORMAT</u>.
Clear rectangle - replace text with space	C-x r c	(clear-rectangle START END &optional FILL)	Blank out the region-rectangle. <ul style="list-style-type: none"> The text previously in the region is overwritten with blanks. With a prefix (or a FILL) argument, also fill with blanks the parts of the rectangle which were empty.
Delete whitespace in rectangle lines		(delete-whitespace-rectangle START END &optional FILL)	Delete all whitespace following a specified column in each line. <ul style="list-style-type: none"> The left edge of the rectangle specifies the position in each line at which whitespace deletion should begin. On each line in the rectangle, all contiguous whitespace starting at that column is deleted. With a prefix (or a FILL) argument, also fill too short lines.
Replace rectangle content with specified string on each line	C-x r t	(string-rectangle START END STRING)	Replace rectangle contents with STRING on each line. <ul style="list-style-type: none"> The length of STRING need not be the same as the rectangle width. When called interactively and option 'rectangle-preview' is non-nil, display the result as the user enters the string into the minibuffer.
Insert string on each rectangle line		(string-insert-rectangle START END STRING)	Insert STRING on each line of region-rectangle, shifting text right. <ul style="list-style-type: none"> This command does not delete or overwrite any existing text.
Toggle rectangle Mark Mode (See also: Σ Marking)	C-x SPC	(rectangle-mark-mode &optional ARG)	Toggle the region as rectangular. <ul style="list-style-type: none"> Activates the region if needed. Only lasts until the region is deactivated. When this mode is active, the region-rectangle is highlighted and can be shrunk/grown, and the standard kill and yank commands operate on it.
Picture Mode Rectangle Commands	The following commands allow drawing rectangles in the buffer as well as copy and remove them. They also allow storing the rectangles in registers and restore them from rectangles.		
Draw rectangle around region	C-c C-r	(picture-draw-rectangle START END)	Draw a rectangle around region.
Clear & save rectangle	C-c C-k	(picture-clear-rectangle START END &optional KILLP)	Clear and save rectangle delineated by point and mark. <ul style="list-style-type: none"> The rectangle is saved for yanking by C-c C-y and replaced with whitespace. The previously saved rectangle, if any, is lost. With prefix argument, the rectangle is actually killed, shifting remaining text.
Clear reactangle	C-c C-w	(picture-clear-rectangle-to-register START END REGISTER &optional KILLP)	Clear rectangle delineated by point and mark into REGISTER. <ul style="list-style-type: none"> The rectangle is saved in REGISTER and replaced with whitespace. With prefix argument, the rectangle is actually killed, shifting remaining text.
Yank and overlay saved rectangle	C-c C-y	(picture-yank-rectangle &optional INSERTP)	Overlay rectangle saved by C-c C-k <ul style="list-style-type: none"> The rectangle is positioned with upper left corner at point, overwriting existing text. With prefix argument, the rectangle is inserted instead, shifting existing text. Leaves mark at one corner of rectangle and point at the other (diagonally opposed) corner.
Overlay rectangle saved in register	C-c C-x	(picture-yank-rectangle-from-register REGISTER &optional INSERTP)	Overlay rectangle saved in REGISTER. <ul style="list-style-type: none"> The rectangle is positioned with upper left corner at point, overwriting existing text. With prefix argument, the rectangle is inserted instead, shifting existing text. Leaves mark at one corner of rectangle and point at the other (diagonally opposed) corner.

Topic & Link	Notes
GNU Emacs Manual — Rectangles	