Rectangles

<u>Operation</u>	<u>Keystroke</u>	Function	Note	
Rectangles See also: Drawing	The rectangle can highlighted area s	be defined by the normal set-mark-comma	made of the area made of the opposite corners of the point and mark. and as shown in the first screen shot below. However, when you use that, the ngle. You just have to remember that the rectangle is made of the opposite corners the command.	
	The rectangle-ma second screenshots Remember that n would prevent you fi defining your rectang PEL binds <f1< th=""><th colspan="3">mark-mode command provides a better visual feedback as it only highlights the area that constitutes the rectangle as shown in</th></f1<>	mark-mode command provides a better visual feedback as it only highlights the area that constitutes the rectangle as shown in		
Set mark & activate/ deactivate it See also: Marking	• C-SPC • C-@ • <f11> . s</f11>	(set-mark-command ARG)	Set the mark where point is and toggle its activation. If mark was not active it activates it: moving the cursor further will show the marked area (the region) if transient mode is enabled (the default in Emacs). If the mark is active, de-activates it. Issuing the command twice (C-SPC C-SPC) sets the mark location and de-activates it. You can use this command to create a rectangle: the rectangle will not show explicitly, in the example below it is defined by the top-left and bottom-right corners of the marked area that is highlighted.	
		<pre>- /dev/elisp/pel — PE File Edit Options Buffers Too - no-indent - full-indent - indent-anchored-on-first-co - indent-to-scope</pre>		
		The styles are shown below. snippets is on the first colu are indented here to help sho - no-indent style:	- full indent:	
		<pre>void some_function(int some { #ifdef USING_WINDOWS #ifdef USING_CYGWIN do_some_unix_call(); #else do_some_windows_call(); #endif</pre>	_arg) void some_function(int some_arg) {	
		<pre>#elif_USING_MAC_OS some_macos_call(); #else do_some_unix_call(); #endif } </pre>	<pre>#elif USING_MAC_OS</pre>	
		- indent-anchored-on-first-co -UU-:F1 pel-ppindent.el	lumn: - indent-to-scope: 43% (79,33) (Emacs-Lisp WK Fly ² Anzu ElDoc)	
Toggle rectangle Mark Mode See also: Marking	C-x SPC	(rectangle-mark-mode &optional ARG)	 Toggle the region as rectangular. 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. See the screenshot below, where the mark was activated with C-x SPC on the first letter of the word "void" and then the cursor moved down right to highlight a rectangle. Nothing "bleeds" outside of the rectangle. 	
		- no-indent - indent-anchored-on-first-co-indent-to-scope		
			- full indent:	
		<pre>{ #ifdef USING_WINDOWS #ifdef USING_CYGWIN do_some_unix_call(); #else do_some_windows_call(); #endif #elif USING_MAC_OS some macos call();</pre>	<pre>#ifdef USING_WINDOWS #ifdef USING_CYGWIN do_some_unix_call(); #else do_some_windows_call(); #endif #elif USING_MAC_OS some_macos_call();</pre>	
		<pre>#else do_some_unix_call(); #endif }</pre>	#else	
Copy/Save rectangle text See also: <u>▼ Cut & Paste</u>	• C-x r M-w • <f11> = r</f11>	(copy-rectangle-as-kill START END)	Copy the region-rectangle and save it as the last killed one.	
Kill text in rectangle See also: • <u>∑ Cut & Paste</u>	• C-x r k • <f11> - r</f11>	(kill-rectangle START END &optional FILL)	Delete the region-rectangle and save it as the last killed one. • 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.)	

Vank last killed rectangle C-x r y yank-rectangle) C-x r o open-rectangle STATT END Application Insert line numbers to left or rectangle number-lines STATT END rectangle numbers to left or rectangle number-lines STATT END STATE NO Application C-x r o pen-rectangle STATT END Application Insert line numbers to left or rectangle number-lines STATT END STATE NO Application C-x r c please-rectangle order C-x r t please-rectangle STATT END STATE NO Application C-x r t please-rectangle order C-x r t please-rectangle order C-x r t please-rectangle STATT END STATE NO STATE NO State or	<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>	
Per cotangle with space Crit P Crit P Fill Per commended Per cotangle	Delete rectangle text	C-x r d		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.	
The Last provided print he last past in first constraints by the basins, but intended to the country.	Yank last killed rectangle	С-х г у	(yank-rectangle)	Yank the last killed rectangle with upper left corner at point.	
rectangle Colar reclangle of TATE PLA SOCIONITY Colar reclangle STATE PLA SOCIONITY Colar reclangle strip in the region is overwhere with bitance in parts of the mechangle with specified using on each late Figure as extension contract with specified using on each late Colar t t (phing-rectangle STATE PLA SOCIONITY Figure as extension contract with a reclangle contract Figure as extension contract with a reclangle reclangle strip in the region in the region is contract as an above the specified using on each late Colar t to the rectangle strip in the rectangle ince Colar thing on adult in the rectangle strip	Fill rectangle with space	С-х г о		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	
FILLI The lest previously in the region is coveration with blanks. The lest previously in the region is coveration with blanks. The lest proviously in the region is coveration with blanks. Out of the section of the rectangle of the rectangle content with specified string on each like. Cr z t Outing-rectangle STAPT END STRING) Fill argument of STRING on each like share as the rectangle with. What called interactively and option rectangle previously in provincing and provided interactively and option rectangle previous in provincing the section of the section rectangle previous in provincing the section of the section rectangle previous in section in each line. Chief the section rectangle previous in section is set in an extension of the section rectangle previous in section in section of the section rectangle previous and the section of the		C-x r N			
** The length of STRING mode on to be the same as the rectangle worth. **Production interacting on each interacting and inter		C-x r c	, ,	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	
END Application FILLy The field edge of the rectangle specifies the protion in each inter at which which which go color and the protion of the protecting of the protion of the protocol of the protion of the proteon of the protion of the protion of the protion of the protion of the proteon of the protion of the proteon of the proteon of the protion of the proteon of the pro	with specified string on each	C-x r t	(string-rectangle START END STRING)	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. This command can be used to draw vertical lines in tables, using as the character	
This command does not delete or overwrite any existing text.	-			 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. 	
* Use other rectangle commands, you must first mark an extangle area using the set-mark-command (C-SPC). See above. * In this mode the following commands are valiable. A. However the other standard exchangle commands are valiable. A. However the other standard exchangle commands are valiable. A. However the other standard exchangle commands are valiable. A. However the other standard exchangle value of the mode. Toggle the ledit-rectangle-mode & optional EFC. END) Redit-rectangle-mode & optional EFC. END) Restricted a region cashing marking. Use set mark command (bound to C-SPC) Betting the rectangle of the rectangle of whitespace inside the text to un-indent inside code. Once the rectangle is spiglided unit on ord the rectangle of specific column. Use the EFC. May to delete it rectangle commands, any key typed inside the rectangle of specific column. Use the EFC. May to delete it rectangle commands, any key typed inside the rectangle of specific column. Use the EFC. May to delete it rectangle column at a time. You can move the cursor inside the rectangle commands, any key typed inside the rectangle of specific column. Use the EFC. May to delete it rectangle column at a time. You can move the cursor inside the rectangle of specific column. Use the EFC. May to delete it rectangle column at a time. You can move the cursor inside the rectangle column and a time. You can move the cursor inside the rectangle column and a time.					
Rectangle editing support is sead on ladd mechanism. First stelect arginous using marking. Use ext-mark-command (bound to C-SPC) and then move point. Substitut to insend or move vertical spacing or editing tabular data. Delta control of the properties of t	iedit-rectangle-mode	Like other rectangIn this mode the formula	this mode the following commands are available. However the other standard rectangle commands above do not always work.		
code. Once the rectangle is highlighted (using a color different from other rectangle) commands), any key typed inside the rectangle is applied on all row of the rectangle on specific column. Use the DEL key to delete 1 rectangle column at a time. You can move the cursor inside the rectangle. "Indevidingole — PEL — any pel-coloridant is appell—bash emiscablook — typ 016 — YM7 "Indevidence — PEL — any pel-coloridant is appell—bash emiscablook — typ 016 — YM7 "Indevidence — PEL — any pel-coloridant is appell—bash emiscablook — typ 016 — YM7 "Indevidence — PEL — any pel-coloridant is appell—bash emiscablook — typ 016 — YM7 "Indevidence — PEL — any pel-coloridant is appell—bash emiscablook — typ 016 — YM7 "Indevidence — Pel-coloridant is appell—bash emiscablook — typ 016 — YM7 "Indevidence — Pel-coloridant indent — anchored — on-first — column indent — anchored — on-first — column indent — to — scope indent — to — to pel		C-x r RET		 First select a region using marking. Use set-mark-command (bound to C-SPC) and then move point. Useful to insert or remove vertical spacing or editing tabular data. PEL activates this key binding on startup when either pel-use-iedit or pel-use- 	
ident-to-scope) (prog1			code. Once the rectangle is highlighted (in rectangle is applied on all row of the rectangle is applied in a control of the rectangle in the rectangle is applied in the rectangle in the rectangle in the rectangle is applied in the rectangle in the rectangle in the rectangle is applied in the rectangle in the rectangle in the rectangle is applied in the rectangle in the rectangle in the rectangle is applied in the rectangle in the rectangle is applied in the rectangle in the rectangle is applied in the rectangle is applied in the rectangle in the rectangle is applied in the rectangle is applied in the rectangle in the rectangle in the rectangle is applied in the rectangle in the rec	using a color different from other rectangle commands), any key typed inside the angle on specific column. Use the DEL key to delete 1 rectangle column at a time. angle. EL — e -nw pel-ppindent.el • aspell — .bash emacs-black — ttys016 — \times 7 Emacs-Lisp Help indent	
"Prompt user and select indentation style for C preprocessor directives."			in in (prog1 (setq pel-ppindent-us (message "Now using %S	dent-anchored-on-first-column dent-to-scope)) ed-style ppindent-style) C preprocessor style for current buffer." pel-ppindent-used-style\$	
Commands			"Prompt user and select inden		
Fel-ppindent-used-style (function pel-ppindent-set-style)) -UU-:**F1 pel-ppindent_el 66% (140,26) (Emacs_Lisp uTr WK Fly 2 Anzu ElDoc Tedit-rect:4) -			(pel-select-from "C preproces '((?n '(?f	"no-indent" no-indent)	
Kill text in rectangle M-K (iedit-kill-rectangle & optional FILL) Kill the rectangle: delete the region-rectangle and save it as the last killed one. • The behavior is the same as 'kill-rectangle' in rect mode. Replace all rectangle text with space characters Insert line numbers in each line of rectangle M-N (iedit-number-occurrences START-AT & optional FORMAT-STRING) M-N (iedit-number-occurrences START-AT & optional FORMAT-STRING) Insert numbers in front of each line of the rectangle • START-AT, if non-nil, should be a number from which to begin counting. FORMAT non-nil, should be a format string to pass to 'format-string' along with the line count. • When called interactively with a prefix argument, prompt for START-AT and FORMAT. Quit edit-mode C-g (iedit-quit) Quit edit-rectangle-mode. Must be typed inside the rectangle to take effect. 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. • To use them you must activate Picture mode first. With PEL use <fil> p Draw rectangle around region. Draw a rectangle around region.</fil>			pel-ppindent	-used-style	
Peplace all rectangle text with space characters Insert line numbers in each line of rectangle Insert line of rectangle Insert line numbers in each line of rectangle Insert numbers in front of each line of the rectangle Insert numbers in front o			-UU-:**Fl pel-ppindent.el	66% (140,26) (Emacs-Lisp uTr WK Fly 2 Anzu ElDoc Iedit-rect:4) -	
Insert line numbers in each line of rectangle Insert line numbers in each line of rectangle Insert line numbers in each line of rectangle Insert numbers in front of each line of the rectangle START-AT, if non-nil, should be a number from which to begin counting. FORMAT non-nil, should be a format string to pass to 'format-string' along with the line count. When called interactively with a prefix argument, prompt for START-AT and FORMAT. Quit edit-mode C-g (iedit-quit) Quit edit-rectangle-mode. Must be typed inside the rectangle to take effect. 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. To use them you must activate Picture mode first. With PEL use <f11> D p Draw rectangle around region.</f11>	Kill text in rectangle	м-к	(iedit-kill-rectangle &optional FILL)		
line of rectangle & optional FORMAT-STRING) START-AT, if non-nil, should be a number from which to begin counting. FORMAT non-nil, should be a format string to pass to 'format-string' along with the line count. When called interactively with a prefix argument, prompt for START-AT and FORMAT. Quit edit-mode C-g (iedit-quit) Quit edit-rectangle-mode. Must be typed inside the rectangle to take effect. 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. To use them you must activate Picture mode first. With PEL use <f11> D p Draw rectangle around region.</f11>		M-SPACE	(iedit-blank-occurrences)	Replace occurrences with blank spaces.	
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. • To use them you must activate Picture mode first. With PEL use <f11> p Draw rectangle around region C-c C-r (picture-draw-rectangle START END) Draw a rectangle around region.</f11>		M-N	1 '	 START-AT, if non-nil, should be a number from which to begin counting. FORMAT, if non-nil, should be a format string to pass to 'format-string' along with the line count. When called interactively with a prefix argument, prompt for START-AT and 	
• They also allow storing the rectangles in registers and restore them from rectangles. • To use them you must activate Picture mode first. With PEL use <f11> p Draw rectangle around region C-c C-r (picture-draw-rectangle START END) Draw a rectangle around region.</f11>	Quit edit-mode	C-g	(iedit-quit)	Quit edit-rectangle-mode. Must be typed inside the rectangle to take effect.	
Draw rectangle around region (picture-draw-rectangle START END) Draw a rectangle around region.		They also allow st	commands allow drawing rectangles in the buffer as well as copy and remove them. w storing the rectangles in registers and restore them from rectangles.		
	•	,		•	
 Coptional KILLP) 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. 	-	C-c C-k		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	
Clear reactangle (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) (picture-clear-rectangle-to-register START END REGISTER & optional KILLP) (Clear rectangle delineated by point and mark into REGISTER. • The rectangle is saved in REGISTER and replaced with whitespace. • With prefix argument, the rectangle is actually killed, shifting remaining text.	Clear reactangle	C-c C-w	(1	The rectangle is saved in REGISTER and replaced with whitespace.	

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Yank and overlay saved rectangle	С-с С-у	(picture-yank-rectangle &optional INSERTP)	Overlay rectangle saved by C-c C-k 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	С-с С-ж	(picture-yank-rectangle-from-register REGISTER &optional INSERTP)	Overlay rectangle saved in REGISTER. 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.

Rectangle - References

Topic & Link	Notes
GNU Emacs Manual — Rectangles	