Marking

Operation	Keystroke	Function	Note
The Mark and the Region	In Emacs, several commands operate on arbitrary contiguous part of the current buffer, called the *region*. To understand Emacs in that regards, you must be aware that: • the cursor is called the "**point**", • the "**mark**" is another location, independent of point, • the area between the point and the mark is called the "**region**". Emacs has a minor mode called the "**Transient Mark Mode**", enabled by default, which highlights the region when the mark is active. This mode provides region highlighting similar to what most other editors do. While the region is highlighted you can execute commands that operate on the region. In transient mark mode the region becomes inactive as soon as you type another key. If you disable the Transient Mark Mode, you can set the mark, move point somewhere else, do lots of other things and then use a command that operate on the region, even though that region is not visible. This is not as easy, as you don't have much visible feedback, but in many situations it is useful. Some commands also behave differently when Transient Mark mode is off. The positions of "mark" can be pushed into the "***mark ring***" for later re-use. • Emacs maintains: 1. One **local mark ring** per buffer. The mark ring is a list of positional elements called "*markers*". The maximum length of each mark ring is controlled by the "*mark-ring-max*" customizable variable which is 16 by default. 2. One **global mark ring**, which holds the markers of the marks set inside each buffer last visited. The maximum length of that global mark ring is controlled by the customizable *"global-mark-ring-max"* variable which is also 16 by default. Emacs has: • Commands to set, unset, activate, deactivate mark (and therefore the region). • Commands to store the mark in the local buffer mark-ring, to use the top one and rotate that ring and to remove the top mark from the ring. • Note that several commands and functions use the word "pop": these do not remove the top element from a ring, it just get a copy from the top mark for use and then rotates the ring. Do not confuse the Emacs description of "popping the ring" with a stack pop operation. • Commands to move point to the current mark, to locations identified in local and global mark rings, exchange the mark and point. • Commands to "mark an object". That is to move and activate the mark (and potentially point too) to create a region over an area of text of interest, to prepare for a forthcoming command that will operate on that region only.		
Managing the Mark	The following commands are used to manage the mark and the mark rings. • Since Emacs undo command does not deal with navigation, and because several navigation commands set the mark, several of these commands help go back to past locations where significant operations where done.		
Show mark ring stats	• `<f11> . ?` • `<f11> ? .`	(pel-mark-ring-stats)	Show info about global and buffer local mark and mark rings; their current and maximum size, buffer and positions for each mark ring entry. • Use it to understand the impact of commands on the mark and mark rings.
Set mark & activate/deactivate it	• **C-SPC** • **C-@** • `<f11> . s`	(**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.
Push point to buffer's mark ring	`<f11> . SPC`	(pel-push-mark-no-activate)	Pushes point to the buffer's mark-ring *without activating the region*. • Equivalent to **C-SPC** when transient-mark-mode is disabled. • Useful to use the mark as a beacon so that later we can easily return to the marked location by typing **M-`** .
Jump to a mark popped off from the global mark ring	• **C-x C-SPC** • **C-x C-@** • `<f11> . g`	(**pop-global-mark**)	Pop off **global** mark ring and jump to the buffer and position of the top (latest entry) of the global mark ring. ☛ Although the docstring of that function describes it as '*popping off*' an element of the global mark ring, it really ***does not remove*** anything from the ring: it gets the top value but puts it back at the bottom effectively rotating the ring. ☛ The command removes entries that correspond to buffers that no longer exist.
Remove top level entry from local buffer mark ring	`<f11> . <Del>`	(pel-popoff-mark-ring)	Remove the top entry from the buffer's mark ring. ☛ Contrary to the commands named "pop" this command performs a destructive removal: it reduces the size of the local buffer's mark-ring.
Jump to location of next mark in buffer's mark ring	• `<f11> . `` • `M-`` • **C-u C-SPC**	(pel-jump-to-mark)	Move point to current mark, set mark to top of buffer's mark-ring, and then rotate the ring (by injecting old mark back at the bottom of mark ring). ☛ When using this command in sequence this effectively move point to all previously marked locations. ☛ This is the same as using the set-mark-command via **C-u C-SPC** (but easier to type and shows a more informative message). pel-jump-to-mark is simply executing `(set-mark-command 1)` and display a more informative message.
Exchange point and mark	• **C-x C-x** • ⌘-j • `<f11> . .`	(**exchange-point-and-mark** &optional ARG)	Exchange the point and mark and reactivate the last region. ☛When a region is marked and you are growing it by hitting more mark command keys, it grows away from the mark. Exchanging point and mark allows you to grow the region in the other direction. ☛The **<f11> . .** key binding is useful in text mode when cua-mode is used and a region is marked.
Exchange point and mark without activating the region	`<f11> . ,`	(pel-exchange-point-and-mark-no-activate)	Exchange point and the mark without activating the region. • Equivalent to above command but de-activates region.
De-activate current mark	• **C-g** • **<ESC><ESC><ESC>** • ⌘-.	(**keyboard-quit**)	De-activate the current mark by a quit command. ⌘-. works on macOS terminal mode, not in graphics mode. Programmatically, (**deactivate-mark**) can be used to deactivate the marked area.
Mark and Region	With most editors when you type over a "selected" region, the text in the selection is automatically replaced by the new text. By default Emacs does not behave like this; instead it allows typing text while there is an active a marked region. If you want Emacs behave like other editors and automatically replace the text activate the "*delete-selection-mode*" with the following command.		
Toggles delete selection mode (Also in: ∑ Text Modes, Cut & Paste)	`<f11> t m d`	(**delete-selection-mode**)	Toggles delete selection-mode on/off. • In delete-selection-mode typing a character while a region is active replaces the entire region with what is typed. By default delete selection-mode is off.
Toggle the Transient Mark Mode	`<f11> . M`	(**transient-mark-mode** &optional ARG)	Toggle Transient Mark mode. • With a prefix argument ARG, enable Transient Mark mode if ARG is • positive, and disable it otherwise. • Transient Mark mode is a global minor mode. When enabled, the region is highlighted with the 'region' face whenever the mark is active. The mark is "deactivated" after certain non-motion commands, including those that change the text in the buffer, and during shift or mouse selection by any unshifted cursor motion command (see Info node 'Shift Selection' for more details). • You can also deactivate the mark by typing **C-g** or **M-ESC ESC**. • Many commands change their behavior when Transient Mark mode is in effect and the mark is active, by acting on the region instead of their usual default part of the buffer's text. • To see the documentation of commands which are sensitive to the Transient Mark mode, invoke **C-h d** and type "transient" or "mark.*active" at the prompt.

Operation	Keystroke	Function	Note

Mark a specific region	Emacs supports several ways of explicitly marking a region:
	• One way is the **shift selection**: **hold the Shift key while moving point** with the cursor or any navigation command.
	  • This is why the Shift key (S-) is never used for any navigation command key binding.
	• The other is to use one of the commands described in the following rows:
	  • The first one selects the semantical element at point, and repeating the command selects a bigger element.
	  • The other commands select specific semantical or buffer element, such as word or line. You can select more of the same element by repeating the command or using a key indicated in the description.
	• Once a region is marked and active you can continue to expand it by using the navigation commands (and cursors), without holding the Shift key.
	• Also, while the region is marked you can type **C-x C-x** to swap point and mark to grow the region in the opposite direction.
	
	The PEL package provides complementary **<f11>** key bindings for all of these commands, even though some other bindings are available, to ensure that these commands are always available in all modes.		

Mark region by semantic unit, increase marked region on each invocation. **★ Powerful command ★**	• **M-=** • **<f11> . =**	(**er/expand-region** ARG)	Increase selected region by semantic units. • With prefix argument expands the region that many times. • If prefix argument is negative calls 'er/contract-region'. • If prefix argument is 0 it resets point and mark to their state before calling 'er/expand-region' for the first time. This command is very powerful: the first time it's typed it selects a word, if you type it again it will expand the selection, and again, and again. The expansions follow the semantics of the current major mode: it is aware of the semantics of several programming languages. ☛ Once **M-=** is typed, you can quickly type the following single keys in sequence: • **=** to expand the region, • **-** to contract the region, • **0** to reset the operation. If you wait too long, then you have to use **M-=** again to continue the expansion, otherwise the region is de-activated. Note that you can also use the following key chords to control the contraction of the selected text without having to worry about time: • **M- M-=** to contract the region • **M-0 M-=** to reset the operation. • Also you can use the cursor keys to expand or contract the region and **C-x C-x** to exchange mark and point to expand the other side of the region with cursors. 📦 This requires the expand-region package. ☛ Under PEL, activated with ***pel-use-expand-region*** customize variable. ☛ The PEL package uses this command and key binding for it, a popular binding for this command is **C-=** but that key does not work in text terminal mode. The standard Emacs binding for **M-=** is normally **count-words-region** used for counting words in region, but PEL provides **<f11> c r** for that.
Mark line(s) going down	• **M-S-<down>** • **<f11> . <down>**	(**pel-mark-line-down** &optional N)	Mark current line or N line forward for going down. • Set mark at beginning of line, move point to line end. • Without argument select the current line. With numeric argument N, selects the current line and N-1 lines below. ☛ Once the line is marked this way, pressing the same keys or **<down>** key alone grows the region by one more line downward.
Mark line(s) going up	• **M-S-<up>** • **<f11> . <up>**	(**pel-mark-line-up** &optional N)	Mark current line or N previous lines for going up. • Move point to start of line, set mark at end of line. • Without argument select the current line. With numeric argument N, selects the current line and N-1 lines above. ☛ Once the line is marked this way, pressing the the same keys or **<up>** key alone grows the region by one more line downward.
Set mark args words away	• **M-@** • **<f11> . w**	(**mark-word** &optional ARG ALLOW-EXTEND)	Set mark ARG words away from point. The place mark goes is the same place **M-f** would move to with the same argument. Interactively, if this command is repeated or (in Transient Mark mode) if the mark is active, it marks the next ARG words after the ones already marked. • Use numerical prefix to specify ARG, which can be negative to indicate backward direction. For example: • to mark the next 23 words use: **M-2 3 M-@** • to mark the previous 6 words, use: **M- - 6 M-@**
mark paragraph	• **M-h** • **<f11> . h**	(**mark-paragraph** &optional ARG ALLOW-EXTEND)	Put point at beginning of this paragraph, mark at end (creating a region that covers the entire current paragraph.) • The paragraph marked is the one that contains point or follows point. • With argument ARG, puts mark at end of a following paragraph, so that the number of paragraphs marked equals ARG. • If ARG is negative, point is put at end of this paragraph, mark is put at beginning of this or a previous paragraph. ☛ See GNU Emacs Paragraphs section for more information about how Emacs defines and handles paragraphs.
mark page	• **C-x C-p** • **<f11> . p**	(**mark-page** &optional ARG)	Put point at beginning of this paragraph, mark at end. • The paragraph marked is the one that contains point or follows point. • With argument ARG, puts mark at end of a following paragraph, so that the number of paragraphs marked equals ARG. • If ARG is negative, point is put at end of this paragraph, mark is put at beginning of this or a previous paragraph.
mark sexp and balanced expressions	• **Esc C-@** • **C-M-@** • **C-M-SPC** • **<f11> . x**	(**mark-sexp** &optional ARG ALLOW-EXTEND)	Set mark ARG sexps (and balanced expressions) from point. • The place mark goes is the same place **C-M-f** would move to with the same argument. • If this command is repeated or (in Transient Mark mode) if the mark is active, it marks the next ARG sexps after the ones already marked. • This command assumes point is not in a string or comment.
• **mark function** • **mark C function** • **mark reStructuredText section**	**C-M-h**	(**mark-defun** &optional ALLOW-EXTEND)	Put mark at end of this defun, point at beginning. Used in Lisp modes. • The defun marked is the one that contains point or follows point. • With positive ARG, mark this and that many next defuns; with negative ARG, change the direction of marking. • If the mark is active, it marks the next or previous defun(s) after the one(s) already marked.
		(**c-mark-function**)	Mark complete function. Used in CC Modes (like c-mode, c++-mode, d-mode). • Put mark at end of the current top-level declaration or macro, point at beginning. • If point is not inside any then the closest following one is chosen. Each successive call of this command extends the marked region by one function. • A mark is left where the command started, unless the region is already active (in Transient Mark mode). • As opposed to C-M-a and C-M-e, this function does not require the declaration to contain a brace block.
		(**rst-mark-section** &optional COUNT ALLOW-EXTEND)	Select COUNT sections around point. Used in reStructuredText mode. • Mark following sections for positive COUNT or preceding sections for negative COUNT.

Operation	Keystroke	Function	Note
mark entire buffer	• **C-x h** • **<f11> . b** • **⌘-a**	(**mark-whole-buffer**)	Put point at beginning and mark at end of buffer. If narrowing is in effect, only uses the accessible part of the buffer.
Rectangle Mark Mode	Emacs also support a mode where operations can be done inside a rectangle shape identified by the mark and point: the rectangle mark mode.		
Toggle rectangle Mark Mode (See also: ∑ Rectangles)	**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.

Marking — References

Topic & Link	Notes
GNU Emacs Manual: The Mark and the Region	
GNU Emacs Manual: The Mark Region - Setting the Mark	
GNU Emacs Manual: The Mark and the Region - Marking Objects	
GNU Emacs Manual: The Mark and the Region - The Mark Ring	
GNU Emacs Manual: The Global Mark Ring	
GNU Emacs Manual: The Mark and the Region - Disabling Transient Mark Mode	
GNU Emacs Manual: Text - Pages	
show-marks.el @ Emacs Wiki	
Fixing the mark commands in transient mark mode	Mickey Petersen's article that describes the use and code of pel-push-mark-no-activate
GNU Emacs Lisp Manual: Markers	Internal information about the marker data type and the Emacs Lisp functions used to manipulate and use them.
GNU Emacs Lisp Manual: Positions	