

Registers

Description	Keystroke	Function	Note
Emacs Registers	The Emacs manual states: “Emacs registers are compartments where you can save text, rectangles, positions, and other things for later use.” Aside from the native Emacs commands, the PEL package adds a set of convenience functions. <div> ⚠ By default, Emacs does not save register content on exit; this information does not persist across Emacs sessions. However, there are several packages that implement project management that store/restore the registers (such as desktop.el) </div> Register names: <ul style="list-style-type: none"> The registers (identified as <code><reg></code> in the table below) are identified by single character name. Even control codes. ⚠ Restriction: do not use <code>’?</code> (as <code>’?</code> is a prefix to identify characters in Elisp) nor Esc nor C-g for register name. Register Commands: <ul style="list-style-type: none"> All <code><f11> r</code> commands (pel- functions) check if the assignment to a register is attempting to use a register that already contain something, and then prompt if so. The C-x r commands (native Emacs commands) do not provide this type of protection. 		
Open this PDF file. See also: 🔗 Help/Info	<code><f11> r <f1></code>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the 🔗 Registers local PDF. If the prefix argument (like C-u or M--) is used, then it opens the remote GitHub hosted raw PDF instead. If the pel-flip-help-pdf-arg user-option is set it's the other way around.
Display register content			
View Register content	<code><f11> r v <reg></code>	(view-register REGISTER)	Opens a small temporary window showing what the specified register holds.
List all Registers	<code><f11> r l</code>	(list-registers)	Opens a window that lists all register names and their content
Record data in register: point, keyboard-macro, window, frame	Once information related to point, keyboard-macro, window or frame is recorded into a register, use the <code><f11> r j</code> command to restore it.		
Copy point position in register <code><reg></code>	<ul style="list-style-type: none"> C-x r SPC <code><reg></code> <code><f11> r p</code> <code><reg></code> 	<ul style="list-style-type: none"> (point-to-register REGISTER &optional ARG) (pel-point-to-register REGISTER &optional ARG) 	Record the position of point and the current buffer in register <i>r</i> . With prefix argument store frame configuration.
Save keyboard macro in register <code><reg></code>	<ul style="list-style-type: none"> C-x C-k x <code><reg></code> <code><f11> r k</code> <code><reg></code> 	<ul style="list-style-type: none"> (kmacro-to-register REGISTER) (pel-kmacro-to-register REGISTER) 	
Save Window layout in register <code><reg></code>	<ul style="list-style-type: none"> C-x r w <code><reg></code> <code><f11> r w</code> <code><reg></code> 	<ul style="list-style-type: none"> (window-configuration-to-register REGISTER) (pel-window-configuration-to-register REGISTER) 	See Preserve window layout in Emacs @ StackOverflow .
“Jump” to register <code><reg></code> to: <ul style="list-style-type: none"> move to point execute keyboard-macro restore window layout restore frame layout 	<ul style="list-style-type: none"> C-x r j <code><reg></code> <code><f11> r j</code> <code><reg></code> 	(jump-to-register REGISTER &optional DELETE)	Used to restore position, windows and frames (and execution keyboard-macros). <ul style="list-style-type: none"> When restoring frames, any frame not included in the configuration become invisible. To delete these frames use: C-u C-x r j R
Record data in register: filename, number, rectangle, text	Once information relate to a file name, a number, a rectangle or text is recorded into a register, use the <code><f11> r i</code> command to restore that data.		
Store the state of all frames and their windows in register <code><reg></code>	<ul style="list-style-type: none"> C-x r f <code><reg></code> <code><f11> r f</code> <code><reg></code> 	<ul style="list-style-type: none"> (frameset-to-register REGISTER) (pel-frameset-to-register REGISTER) 	Store the state (position and sizes) of all frames and all their windows in register.
Store file name in register <code><reg></code>	<code><f11> r F <reg></code>	(pel-filename-to-register REGISTER)	Store the file name (with full path) of the file edited in current buffer in register.
Store NUMBER into register <code><reg></code>	<ul style="list-style-type: none"> C-x r n <code><reg></code> <code><f11> r n</code> <code><reg></code> 	<ul style="list-style-type: none"> (number-to-register NUMBER REGISTER) (pel-number-to-register NUMBER REGISTER) 	Use an argument to the command to specify the number value.
Store 0 into Register <code><reg></code>			If no argument specified the number 0 is stored in the register. To specify another value enter it as a numeric argument.
Increment value stored in register <code><reg></code> by NUMBER	<ul style="list-style-type: none"> C-x r + <code><reg></code> <code><f11> r +</code> <code><reg></code> 	(increment-register PREFIX REGISTER)	If no argument, increment by 1. To increment by a larger amount, specify the number via a numeric argument. For example, to increment register <i>a</i> by 35, use: M-3 M-5 <f11> r + a
Copy region rectangle into register <code><reg></code>	<ul style="list-style-type: none"> C-x r r <code><reg></code> <code><f11> r r</code> <code><reg></code> 	<ul style="list-style-type: none"> (copy-rectangle-to-register REGISTER START END &optional DELETE-FLAG) (pel-copy-rectangle-to-register REGISTER START END &optional DELETE-FLAG) 	If a prefix (C-u) argument is used, delete the rectangle from buffer.
Copy region text in register <code><reg></code>	<ul style="list-style-type: none"> C-x r s <code><reg></code> <code><f11> r t</code> <code><reg></code> 	<ul style="list-style-type: none"> (copy-to-register REGISTER START END &optional DELETE-FLAG REGION) (pel-copy-to-register REGISTER START END &optional DELETE-FLAG REGION) 	If a prefix (C-u) argument is used, delete the text from buffer.
Prepend region text to text in register <code><reg></code>	<code><f11> r , <reg></code>	(prepend-to-register REGISTER START END &optional DELETE-FLAG)	F11 Mnemonic: the <code>,</code> key is on the same key as <code><</code> (which points toward the end of text).
Append region text to text in register <code><reg></code>	<code><f11> r . <reg></code>	(append-to-register REGISTER START END &optional DELETE-FLAG)	F11 Mnemonic: the <code>.</code> key is on the same key as <code>></code> (which points toward the end of text).
Insert register data at point: <ul style="list-style-type: none"> filename number rectangle text 	<ul style="list-style-type: none"> C-x r i <code><reg></code> <code><f11> r i</code> <code><reg></code> 	(insert-register REGISTER &optional ARG)	Normally it leaves point after the inserted text and the mark before. With a numeric argument it puts the point before the text and the mark after.

Registers — References

Topic & Link	Description
GNU Emacs - Registers	
Register names	Single characters. Control and escape sequences can be used.
Stack Overflow - Preserve Window layout	Describes a set of packages that can also be used. Registers seems to be the best way to store named layouts though.