

Description	Keystroke	Function	Note
<b>Emacs Registers</b>	<p>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.</p> <p>⚠ 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)</p> <p><b>Register names:</b></p> <ul style="list-style-type: none"> <li>The registers (identified as &lt;reg&gt; in the table below) are identified by single character name. Even control codes.</li> <li>⚠ Restriction: do not use ‘?’ (as ‘?’ is a prefix to identify characters in Elisp) nor <b>Esc</b> nor <b>C–g</b> for register name.</li> </ul> <p><b>Register Commands:</b></p> <ul style="list-style-type: none"> <li>All &lt;f11&gt; r 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 <b>C–x r</b> commands (native Emacs commands) do not provide this type of protection.</li> </ul>		
Display register content			
View Register content	<f11> r v <reg>	(view-register REGISTER)	Opens a small temporary window showing what the specified register holds.
List all Registers	<f11> r l	(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 <f11> r j command to restore it.		
Copy point position in register <reg>	<ul style="list-style-type: none"> <li><b>C–x r SPC</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r p &lt;reg&gt;</li> </ul>	<ul style="list-style-type: none"> <li>(point-to-register REGISTER &amp;optional ARG)</li> <li>(pel-point-to-register REGISTER &amp;optional ARG)</li> </ul>	Record the position of point and the current buffer in register r. With prefix argument store frame configuration.
Save keyboard macro in register <reg>	<ul style="list-style-type: none"> <li><b>C–x C–k x</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r k &lt;reg&gt;</li> </ul>	<ul style="list-style-type: none"> <li>(kmacro-to-register REGISTER)</li> <li>(pel-kmacro-to-register REGISTER)</li> </ul>	
Save Window layout in register <reg>	<ul style="list-style-type: none"> <li><b>C–x r w</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r w &lt;reg&gt;</li> </ul>	<ul style="list-style-type: none"> <li>(window-configuration-to-register REGISTER)</li> <li>(pel-window-configuration-to-register REGISTER)</li> </ul>	See <a href="#">Preserve window layout in Emacs @ StackOverflow</a> .
“Jump” to register<reg>to: <ul style="list-style-type: none"> <li>move to point</li> <li>execute keyboard-macro</li> <li>restore window layout</li> <li>restore frame layout</li> </ul>	<ul style="list-style-type: none"> <li><b>C–x r j</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r j &lt;reg&gt;</li> </ul>	(jump-to-register REGISTER &optional DELETE)	Used to restore position, windows and frames (and execution keyboard-macros). <ul style="list-style-type: none"> <li>When restoring frames, any frame not included in the configuration become invisible.</li> <li>To delete these frames use: <b>C–u C–x r j R</b></li> </ul>
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 <f11> r i command to restore that data.		
Store the state of all frames and their windows in register <reg>	<ul style="list-style-type: none"> <li><b>C–x r f</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r f &lt;reg&gt;</li> </ul>	<ul style="list-style-type: none"> <li>(frameset-to-register REGISTER)</li> <li>(pel-frameset-to-register REGISTER)</li> </ul>	Store the state (position and sizes) of all frames and all their windows in register.
Store file name in register <reg>	<f11> r F <reg>	(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 <reg>	<ul style="list-style-type: none"> <li><b>C–x r n</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r n &lt;reg&gt;</li> </ul>	<ul style="list-style-type: none"> <li>(number-to-register NUMBER REGISTER)</li> <li>(pel-number-to-register NUMBER REGISTER)</li> </ul>	Use an argument to the command to specify the number value.
Store 0 into Register <reg>			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<reg>by NUMBER	<ul style="list-style-type: none"> <li><b>C–x r +</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r + &lt;reg&gt;</li> </ul>	(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 a by 35, use: <b>M–3 M–5 &lt;f11&gt; r + a</b>
Copy region rectangle into register <reg>	<ul style="list-style-type: none"> <li><b>C–x r r</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r r &lt;reg&gt;</li> </ul>	<ul style="list-style-type: none"> <li>(copy-rectangle-to-register REGISTER START END &amp;optional DELETE-FLAG)</li> <li>(pel-copy-rectangle-to-register REGISTER START END &amp;optional DELETE-FLAG)</li> </ul>	If a prefix (C-u) argument is used, delete the rectangle from buffer.
Copy region text in register <reg>	<ul style="list-style-type: none"> <li><b>C–x r s</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r t &lt;reg&gt;</li> </ul>	<ul style="list-style-type: none"> <li>(copy-to-register REGISTER START END &amp;optional DELETE-FLAG REGION)</li> <li>(pel-copy-to-register REGISTER START END &amp;optional DELETE-FLAG REGION)</li> </ul>	If a prefix (C-u) argument is used, delete the text from buffer.
Prepend region text to text in register <reg>	<f11> r , <reg>	(prepend-to-register REGISTER START END &optional DELETE-FLAG)	F11 Mnemonic: the , key is on the same key as < (which points toward the end of text).
Append region text to text in register <reg>	<f11> r . <reg>	(append-to-register REGISTER START END &optional DELETE-FLAG)	F11 Mnemonic: the . key is on the same key as > (which points toward the end of text).
Insert register data at point: <ul style="list-style-type: none"> <li>filename</li> <li>number</li> <li>rectangle</li> <li>text</li> </ul>	<ul style="list-style-type: none"> <li><b>C–x r i</b> &lt;reg&gt;</li> <li>&lt;f11&gt; r i &lt;reg&gt;</li> </ul>	(insert-register REGISTER &optional ARG)	Normally it leaves point after the inserted text and the mark before. With a <b>numeric argument</b> 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.