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. ⚠ 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) **Register names:** • The registers (identified as *<reg>* in the table below) are identified by single character name. Even control codes. • ⚠ Restriction: do not use ‘**?**’ (as ‘?’ is a prefix to identify characters in Elisp) nor **Esc** nor **C-g** for register name. **Register Commands:** • All **<f11> 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 **C-x r** commands (native Emacs commands) do not provide this type of protection.		
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: **p**oint, **k**eyboard-macro, **w**indow, **f**rame	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>	• **C-x r SPC** *<reg>* • *<f11> r p <reg>*	• (**point-to-register** REGISTER &optional ARG) • (**pel-point-to-register** REGISTER &optional ARG)	Record the position of point and the current buffer in register r. With prefix argument store frame configuration.
Save keyboard macro in register <reg>	• **C-x C-k x** *<reg>* • *<f11> r k <reg>*	• (**kmacro-to-register** REGISTER) • (**pel-kmacro-to-register** REGISTER)	
Save Window layout in register <reg>	• **C-x r w** *<reg>* • *<f11> r w <reg>*	• (**window-configuration-to-register** REGISTER) • (**pel-window-configuration-to-register** REGISTER)	See Preserve window layout in Emacs @ StackOverflow.
“Jump” to register<reg>to: • **move to point** • **execute keyboard-macro** • **restore window layout** • **restore frame layout**	• **C-x r j** *<reg>* • *<f11> r j <reg>*	(**jump-to-register** REGISTER &optional DELETE)	Used to restore position, windows and frames (and execution keyboard-macros). • 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: **f**ilename, **n**umber, rectangle, **t**ext	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>	• **C-x r f** *<reg>* • *<f11> r f <reg>*	• (**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 <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>	• **C-x r n** *<reg>* • *<f11> r n <reg>*	• (**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 <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	• **C-x r +** *<reg>* • *<f11> r + <reg>*	(**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: **M-3 M-5 <f11> r + a**
Copy region rectangle into register <reg>	• **C-x r r** *<reg>* • *<f11> r r <reg>*	• (**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 <reg>	• **C-x r s** *<reg>* • *<f11> r t <reg>*	• (**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 <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: • **filename** • **number** • **rectangle** • **text**	• **C-x r i** *<reg>* • *<f11> r i <reg>*	(**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.