


Shells

Description	Keystroke	Function	Note
Emacs Shells	Emacs provides multiple ways of executing shell commands or running programming language specialized shells.		
Run a shell command	<ul style="list-style-type: none"><li>M-!</li><li>• ⌘-L</li></ul>	(shell-command COMMAND &optional OUTPUT-BUFFER ERROR-BUFFER)	Prompts for the command in the minibuffer, show the command output in the next window in the "Shell Command Output" buffer in Fundamental mode.
Run a command on a marked region	M-	(shell-command-on-region START END COMMAND &optional OUTPUT-BUFFER REPLACE ERROR-BUFFER DISPLAY-ERROR-BUFFER)	Execute string COMMAND in inferior shell with region as input. <ul style="list-style-type: none"><li>Normally display output (if any) in temp buffer "Shell Command Output";</li><li>Prefix arg means replace the region with it. Return the exit code of COMMAND.</li><li>Mark the region first. Then type M- . Emacs prompts for the command to run. Use an argument to replace the region with the command output (ie. type C-u M- )</li></ul>
Run a shell command asynchronously	M-&	(async-shell-command COMMAND &optional OUTPUT-BUFFER ERROR-BUFFER)	Execute string COMMAND asynchronously in background. <ul style="list-style-type: none"><li>Like 'shell-command', but adds '&amp;' at the end of COMMAND to execute it asynchronously.</li><li>The output appears in the buffer "Async Shell Command".</li><li>That buffer is in shell mode.</li></ul>
Open an eshell	<f11> x e	(eshell &optional ARG)	<b>Implementation:</b> <ul style="list-style-type: none"><li>eshell is implemented in Emacs Lisp and implements several Unix commands, making them available to OS that do not natively have them (like Windows). If a command is not implemented it runs the one found in PATH.</li></ul> <b>Extra Features</b> <ul style="list-style-type: none"><li>Can redirect output into a buffer. The grep command output goes to a grep result buffer which can be used to open the various files.</li><li>Support lisp commands.</li></ul> <b>Supports</b> <ul style="list-style-type: none"><li>Cursor lateral cursor line beginning/end, kill, yank.</li><li>Meta-cursor word-move keys, but going left it does not stop at the prompt.</li><li>command tab expansion</li><li>command line re-direction</li><li>Is colouring (done by the eshell implementation), columns are aligned.</li><li>Command history (and shows history item # in mini-buffer)</li><li>Can run top, man, less (which start inside separate buffer)</li><li>Can run Python scripts.</li></ul> <b>Limitations:</b> <ul style="list-style-type: none"><li>Meta-cursor word-move keys going left does not stop at the prompt.</li><li>Clear screen does not work</li><li>No bash alias, however eshell can remember its own aliases and will prompt for commands often ran &amp; unfound.</li></ul> <p>To open another eshell: use the C-u prefix To open a numbered shel: use the C-u number prefix</p>
Open a shell	<f11> x s	(shell &optional BUFFER)	<b>Implementation</b> <p>The oldest emacs shell. Uses the comint-mode. Emacs keys are possible, the sub-process does not see the keys until &lt;RET&gt; is pressed.</p> <b>Supports</b> <ul style="list-style-type: none"><li>Can run multiple shell, each inside its own buffer/name</li><li>Cursor lateral cursor line beginning/end, kill, yank.</li><li>Meta-cursor word-move keys.</li><li>bash alias</li><li>Command history (but with Control Up/Down)</li><li>Can run Python scripts.</li><li>Can run Python REPL<ul style="list-style-type: none"><li>REPL is OK, echo is OK, no Python colouring, but each command is colored.</li></ul></li><li>Can run Common-Lisp (clisp) REPL</li></ul> <b>Limitations:</b> <ul style="list-style-type: none"><li>Clear screen does not work</li><li>Is colouring does not work, Is columns are misaligned.</li><li>Can start and stop top, but the output is incorrect and cannot be read.</li><li>The shell PS1 prompt is partially applied, remnants show up on the second line. TO-INVESTIGATE?.</li></ul> <b>Current Problems to resolve in PEL</b> <ul style="list-style-type: none"><li>When running Python REPL:<ul style="list-style-type: none"><li>REPL has trailing space which are highlighted by PEL. This should be disabled in shell mode.</li><li>would be nice to have Python mode activated.</li></ul></li></ul> <p>To Open a shell inside another window: use the C-u prefix</p>
Open an ANSI term shell  (Preferred)	<f11> x a	(ansi-term PROGRAM &optional NEW-BUFFER-NAME)	<b>Implementation</b> <ul style="list-style-type: none"><li>Prompts for shell to use. Default is /bin/bash. Can use others. Opens in current window.</li><li>A terminal emulator written in Emacs Lisp.</li><li>Newer implementation than term.</li><li>You can even run other editors within it (vi, emacs, others). But use character-mode.</li></ul> <b>Specificities:</b> <ul style="list-style-type: none"><li>C-x is mapped to term-escape-char</li></ul> <b>Supports:</b> <ul style="list-style-type: none"><li>Scroll up/down with M-&lt;up&gt;, M-&lt;down&gt;</li><li>Is colouring, columns are aligned</li><li>bash alias</li><li>bash tab expansion</li><li>command line redirection</li><li>clear screen</li><li>Command history</li><li>Can run Python scripts.</li><li>Running Python shell:<ul style="list-style-type: none"><li>REPL is OK, echo is OK</li></ul></li></ul> <b>Limitations:</b> <ul style="list-style-type: none"><li>Natively runs in character mode, which does not allow movement nor saving.</li><li>&lt;up&gt;, &lt;down&gt; cursor, C-n/C-p do not work as navigating: used as shell command history. Change to line mode (see above) to enable these.</li><li>⚠ Normally operates in character mode, in which up/down navigation and kill/yank is not possible. Change to line mode to do that:<ul style="list-style-type: none"><li>Use C-x C-j to change to line mode an allow movement, mark, saving.</li><li>When done use C-c C-k to switch to character mode.</li></ul></li><li>Have not yet found a way to control prompt (PS1 setup of .bash_profile does not seem to be used). TO-INVESTIGATE.</li></ul>

Description	Keystroke	Function	Note
<b>Open a term shell</b>	<f11> x t	(term PROGRAM)	<div> Prompts for shell to use. Default is /bin/bash. Can use others. Opens in current window. </div> <div> <b>Implementation:</b>  Shell implemented in Emacs Lisp. The keys are sent directly to the sub-process, which means they are not interpreted by Emacs. </div> <div> Same access as normal shell: can use the bash alias, tab-autocomplete, clear screen, can use less and indirection, can execute python scripts. Can even run other terminal editors like vim, synaptic, etc... </div> <div> <b>Supports</b> <ul style="list-style-type: none"> <li>Cursor lateral cursor line beginning/end, kill, yank.</li> <li>Meta-cursor keys, but only in terminal Emacs, not in GUI Emacs.</li> <li>Is colouring, columns are aligned</li> <li>bash alias</li> <li>bash tab expansion</li> <li>command line redirection</li> <li>clear screen</li> <li>Command history</li> <li>Can run Python scripts.</li> <li>Running Python shell: <ul style="list-style-type: none"> <li>REPL is OK, echo is OK</li> </ul> </li> </ul> </div> <div> <b>Limitations:</b> <ul style="list-style-type: none"> <li>In GUI Emacs: Meta-left/right cursor word move do not work. Use Esc-b and Esc-f here instead.</li> <li>Normal Emacs keystrokes does not always work, it depends on the programs that are executed from the shell. When it stops working, either use <b>C-c b</b> to switch to another buffer or exit the shell to gain control to Emacs keys in this buffer.</li> <li>Vertical cursor history works only with Control-Up and Control-Down</li> <li>Emacs keys with Meta do not work. The ones with Control do work.</li> <li>Can run top in the buffer, but then C-c does not stop it. To stop it split the buffer in 2, kill the buffer with C-x k, confirm, close the buffer.</li> </ul> </div>
<b>Specialized REPL</b>	You can run several read eval run loop programming shells in Emacs. The first two are part of Emacs. Others are available, such as the Erlang shell which comes with Erlang support.		
<b>Emacs Lisp shell</b>  (See also: ⓘ - Emacs Lisp)	<f11> x i	(ielm)	<div> Open the Interactive Emacs Lisp Mode buffer where you can interactively evaluate Emacs Lisp expressions, a REPL for Emacs Lisp. </div> <ul style="list-style-type: none"> <li>Switches to the buffer <b>"*ielm*</b>", or creates it if it does not exist.</li> </ul>
<b>Start Python shell</b>  (See Also: ⓘ Python)	<f11> x p	(run-python &optional CMD DEDICATED SHOW)	<div> Run an inferior Python process. </div> <ul style="list-style-type: none"> <li>Argument CMD defaults to 'python-shell-calculate-command' return value. When called interactively with 'prefix-arg', it allows the user to edit such value and choose whether the interpreter should be DEDICATED for the current buffer. When numeric prefix arg is other than 0 or 4 do not SHOW.</li> <li>For a given buffer and same values of DEDICATED, if a process is already running for it, it will do nothing. This means that if the current buffer is using a global process, the user is still able to switch it to use a dedicated one.</li> </ul>
<b>Start Erlang Shell</b>	<ul style="list-style-type: none"> <li>&lt;f11&gt; x r</li> <li>C-c C-z</li> </ul>	(erlang-shell)	<div> Start a new Erlang shell. </div> <ul style="list-style-type: none"> <li>The variable 'erlang-shell-function' decides which method to use, default is to start a new Erlang host. It is possible that, in the future, a new shell on an already running host will be started.</li> <li>C-c C-z starts the Erlang Shell from the Erlang Mode.</li> <li>&lt;f11&gt; x r starts it anytime, as long as it was installed.</li> </ul> <div>  Under PEL this command is available only when the <b>pel-use-erlang</b> customize variable is set to <b>t</b>. </div>

## Shells — References

Topic & Link	Extra Notes
<b>GNU Emacs - Running Shell Commands</b>	
<b>Eshell manual</b>	
<b>Difference between various emacs shells</b>	
<b>Difference between various emacs shells</b>	
<b>How to run multiple shells on Emacs</b>	
<b>EmacsWiki: Ansi Term</b>	Quick overview
<b>Emacswiki: Ansi Term Hints</b>	Several hints
<b>Copy/Paste in Ansi Term</b>	Quick overview of the capability for cut/paste.
<b>Launch GUI emacs from command line in OSX</b>	This describes a solution on how to start the GUI emacs in OSX, but not in the background
<b>How to launch GUI Emacs from command line in OSX?</b>	This one describes the solution for handling it in the background
<b>Run commands in background</b>	Describes the & and the disown
<b>Executing commands in background from bash scripts</b>	
<b>Pass command arguments to bash scripts</b>	