






















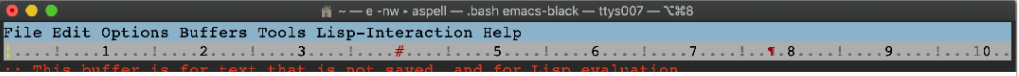








# Highlighting

Operation	Keystroke	Function	Note
<b>Highlighting</b>  See also: <a href="#">↷ Customize</a>	<p>Emacs provides different ways of highlighting text, lines, columns and cursor. It highlights on search operations but can also be instructed to highlight regular expressions, matched delimiters, the current line, the current column, the fill column where automatic text wrapping occurs.</p> <p> PEL provides the pel-pkg-for-highlight customization group to control some aspect of highlighting. The user options are:</p> <ul style="list-style-type: none"><li>•  <b>pel-use-auto-highlight-symbol</b> activates the <a href="#">auto-highlight-symbol</a> external package.</li><li>•  <b>pel-use-iedit</b> activates <a href="#">iedit</a> external package which allows you to interactively edit all symbols inside current scope or marked region.</li><li>•  <b>pel-use-fill-column-indicator</b> activates the <a href="#">fill-column-indicator</a> external package for Emacs versions earlier than 27.1.<ul style="list-style-type: none"><li>• This highlights a vertical line on the fill-column, the column where automatic line wrapping occurs (when the mode is active).</li></ul></li><li>•  <b>pel-use-highlight-indentation</b> activates the <a href="#">highlight-indentation</a> external package that helps see the indentation.</li><li>•  <b>pel-use-vline</b> activates <a href="#">vline.el</a> to create a vertical bar across the entire buffer that moves point (the cursor).</li><li>•  <b>pel-use-rainbow-delimiters</b> activates <a href="#">rainbow-delimiters.el</a> that highlights matching parens delimiters.</li><li>•  <b>pel-use-rainbow-mode</b> activates <a href="#">rainbow-mode</a> external package which <b>highlights strings containing color codes</b>.</li><li>• The built-in <b>show-paren-mode</b> provides matching parens highlighting.</li></ul> <p>PEL provides a set of key bindings to various highlight control commands that are regrouped under the pel:highlight prefix: <b>&lt;f11&gt; h</b>.</p> <ul style="list-style-type: none"><li>• They are listed in the table below along the native Emacs key bindings for highlight control.</li></ul>		
<b>Open this PDF file.</b> See also: <a href="#">↷ Help/Info</a>	<b>&lt;f11&gt; h &lt;f1&gt;</b>	<b>(pel-help-pdf</b> &optional OPEN-WEB-PAGE)	Open the <a href="#">↷ Highlight</a> local PDF. If the prefix argument (like <b>C-u</b> or <b>M--</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it's the other way around.
<a href="#">↷ Customize</a> PEL highlighting control	<b>&lt;f11&gt; h &lt;f2&gt;</b>	<b>(pel-customize-pel</b> &optional OTHER-WINDOW)	Customize the following PEL support: <ol style="list-style-type: none"><li>1. PEL support for highlighting (pel-pkg-for-highlight)</li><li>2. PEL support for mode line (pel-pkg-for-modeline)</li><li>3. PEL support for parens (pel-pkg-for-parens)</li></ol> <ul style="list-style-type: none"><li>• If OTHER-WINDOW is non-nil (use <b>C-u</b>) , display in other window.</li></ul>
<a href="#">↷ Customize</a> Emacs highlighting control	<b>&lt;f11&gt; h &lt;f3&gt;</b>	<b>(pel-customize-library</b> &optional OTHER-WINDOW)	Customize Emacs support for buffer highlight management: <a href="#">auto-highlight-symbol</a> , <a href="#">iedit</a> , <a href="#">highlight-indentation</a> , <a href="#">paren-showing</a> , <a href="#">rainbow-mode</a> , <a href="#">rainbow-delimiters</a> , <a href="#">smartparens</a> , <a href="#">vline</a> , <a href="#">fill-column-indicator</a> (for Emacs version earlier than 27.1)
<b>Font Locking Control</b>	Use these commands to toggle font locking in the buffer or globally.		
<b><a href="#">Toggle syntax highlighting in current buffer</a></b> See also: <a href="#">↷ Faces/Fonts</a>	<b>&lt;f11&gt; h F</b>	<b>(font-lock-mode</b> &optional ARG)	Toggle syntax highlighting in this buffer (Font Lock mode). <ul style="list-style-type: none"><li>• With a prefix argument ARG, enable Font Lock mode if ARG is positive, and disable it otherwise.</li></ul>
<b><a href="#">Enable/disable highlight lock mode</a></b>	<b>&lt;f11&gt; h L</b>	<b>(hi-lock-mode</b> &optional ARG)	Toggle selective highlighting of patterns (Hi Lock mode). <ul style="list-style-type: none"><li>• With prefix argument, enable Hi Lock mode if ARG is positive, disable it otherwise.</li><li>• When text is already highlighted toggling the mode off will remove all highlighting; turning it back on will not restore the highlighting.</li></ul>
<b><a href="#">Enable highlight lock mode to all buffers</a></b>	<b>&lt;f11&gt; h M-L</b>	<b>(global-hi-lock-mode</b> &optional ARG)	Toggle Hi-Lock mode in all buffers. <ul style="list-style-type: none"><li>• With prefix ARG, enable Global Hi-Lock mode if ARG is positive; otherwise, disable it.</li></ul>
<b>Highlight Blocks</b>	These commands are used to highlight matching parentheses, but also braces and square brackets. Can be used in any editing mode, not restricted to Lisp only (but very useful in for editing Lisp code).		
<b><a href="#">Enable/Disable highlighting matching block (), {}, []</a></b>	<b>&lt;f11&gt; h (</b>	<b>(show-paren-mode</b> &optional ARG)	Toggle visualization of matching parens (Show Paren mode). <ul style="list-style-type: none"><li>• With a <b>C-u</b> prefix argument ARG, enable Show Paren mode if ARG is positive, and disable it otherwise. If called from Lisp, enable the mode if ARG is omitted or nil.</li></ul>
	<ul style="list-style-type: none"><li>• <b>&lt;f12&gt; M-9</b></li><li>• <b>&lt;M-f12&gt; M-9</b></li></ul>		 The <b>&lt;f12&gt; M-9</b> and <b>&lt;M-f12&gt; M-9</b> bindings are available for several programming modes.
<b><a href="#">Enable/Disable coloured highlight of nested blocks (), {}, []</a></b>	<b>&lt;f11&gt; h M-(</b>	<b>(rainbow-delimiters-mode</b> &optional ARG)	Highlight nested parentheses, brackets, and braces with different colours according to their depth. <ul style="list-style-type: none"><li>• Customize the depth and colours with M-x customize-group rainbow-delimiters or with PEL access its customization group with the <b>&lt;f11&gt; h &lt;f3&gt;</b> key sequence and select rainbow-delimiters.</li></ul>  <b>Requires:</b> <a href="#">rainbow-delimiters.el</a>  activated by <a href="#">pel-use-rainbow-delimiters</a>
<b>Highlight Cursor</b>  (Emacs >=27)	Requires  external <a href="#">beacon package</a> . PEL activates when  <b>pel-use-beacon-mode</b> is turned on from start or on request. If <b>pel-use-beacon-mode</b> is set to t: you must toggle the mode on explicitly. If it is set to <b>use-from-start</b> , PEL activates it automatically.		
<b>Toggle beacon-mode</b>	<b>&lt;f11&gt; h M-b</b>	<b>(beacon-mode</b> &optional ARG)	Toggle the beacon-mode. This highlights the cursor changing focus.  the key binding is only active when <b>pel-use-beacon-mode</b> is not nil.
<b>Blink beacon at cursor</b>	<b>&lt;f11&gt; h b</b>	<b>(beacon-blink)</b>	Blink the beacon at the position of the cursor. Even when beacon-mode is not active.
<b>Highlight Recent Changes</b>	This mode is a quick way to highlight what has been recently changed in the buffer.  All highlighting in a buffer is <b>non persistent</b> : it is forgotten when buffer is killed.		
<b><a href="#">Enable/Disable Highlight Changes Mode</a></b>	<b>&lt;f11&gt; h M-c</b>	<b>(highlight-changes-mode</b> &optional ARG)	Toggle highlighting changes in this buffer (Highlight Changes mode). <ul style="list-style-type: none"><li>• With a prefix argument ARG, enable Highlight Changes mode if ARG is positive, and disable it otherwise.</li><li>• In this mode the most recent changes in the buffer are highlighted.<ul style="list-style-type: none"><li>•  when enabled, syntax highlighting does not work properly for the new text.</li></ul></li></ul>
<b>Highlight Current Line</b>	Highlighting the current line may help to find the cursor when editing with big windows. These commands control line highlighting.		
<b><a href="#">Toggle line highlight mode</a></b>  See also: <a href="#">↷ Cursor</a>	<ul style="list-style-type: none"><li>• <b>&lt;f11&gt; h 0</b></li><li>• <b>&lt;f11&gt; 0</b></li></ul>	<b>(hl-line-mode</b> &optional ARG)	Toggle highlighting of the current line (HI-Line mode) in the current buffer. With prefix argument ARG, enable HI-Line mode if ARG is positive, disable it otherwise. <ul style="list-style-type: none"><li>• When same buffer is shown in several windows, the highlighting might show in each of them.<ul style="list-style-type: none"><li>• Change that with pel-toggle-hl-line-sticky with: <b>&lt;f11&gt; h s</b></li></ul></li><li>•  A quick way to find where your cursor is located is to hit <b>&lt;f11&gt; 0</b> quickly to toggle line highlighting on and off (that key binding is easier to type than the alternative, which exists for consistency, remember that you can use <b>&lt;f1&gt; k</b> to get help for a specific key binding and see all the key bindings for a command, allowing you to discover its other bindings.)</li></ul>
<b><a href="#">Toggle line highlighting affecting all windows</a></b>	<b>&lt;f11&gt; h s</b>	<b>(pel-toggle-hl-line-sticky)</b>	Toggle current line highlight to all windows with current buffer or just the current one. <ul style="list-style-type: none"><li>• Toggles the value of 'hl-line-sticky-flag' between t and nil.</li></ul>
<b><a href="#">Change color of highlight line for session</a></b>	<b>&lt;f11&gt; h c</b>	<b>(pel-set-highlight-color</b> COLORNAME)	Set the colour of the highlight line used in the line highlight mode (affects all buffers). <ul style="list-style-type: none"><li>• Prompt for color name, use tab completion to see available colour names.</li></ul>
	<ul style="list-style-type: none"><li>• The change does not persist when Emacs is closed.</li></ul>		To select a persistent color, then customize the <b>highlight</b> user option (see next row).
<a href="#">↷ Customize</a> highlight color and attributes permanently	<b>&lt;f11&gt; h &lt;f4&gt;</b>	<b>(pel-customize-highlight)</b>	Customize the <b>highlight</b> user option color and other attributes. <ul style="list-style-type: none"><li>• As with all customizations, you can activate the change for this Emacs session or save it to make it persist across Emacs sessions.</li><li>• With this you can set other attributes such as underlining (which will underline only text present in the buffer, useful to detect end-of-line whitespace), and other attributes.</li></ul>

Operation	Keystroke	Function	Note
<b>Visualize Color Codes</b> <b>Using rainbow mode:</b> highlight the color of color codes. 🍌 See color code names in rainbow customization group.			
Toggle rainbow mode on/off	<f11> h M-r	(rainbow-mode &optional ARG)	Colorize strings that represent colors. <ul style="list-style-type: none"> <li>This will fontify with colors the string like "#aabbcc" or "blue".</li> </ul>  Requires: <b>rainbow-mode</b>  activated when <b>pel-use-raibow-mode</b> is <b>t</b> .
	🍌 When <b>rainbow-mode</b> is active inside a buffer you can easily build a command line that sets the color elements for that command as shown at right (in Emacs running inside a terminal). <ul style="list-style-type: none"> <li>With <b>overwrite-mode</b> on you can change the hex color values and see the impact right away!</li> </ul> 		
<b>Highlight current column</b>	The following command provide a vertical line across the entire window at the cursor location. <ul style="list-style-type: none"> <li>Useful when creating tables or checking indentation manually.</li> <li>vline also provides the vline-global-mode to activate the vertical line in all buffers; PEL has no binding for it because it slows Emacs too much.</li> </ul>		
<b>Toggle Vline Mode</b> See also: <a href="#">🔗 Cursor</a> <a href="#">🔗 Hide/Show</a>	<ul style="list-style-type: none"> <li>&lt;f11&gt; h  </li> <li>&lt;f11&gt; 9</li> </ul>	(vline-mode &optional ARG)	Toggle the display of a vertical line spanning the entire window at the cursor column.  Requires: <b>vline.el</b>  activated when <b>pel-use-vline</b> user option is <b>t</b> .
<b>Highlight Fill Column</b>	The fill column identifies the end of line where line wrapping occurs. <ul style="list-style-type: none"> <li>Some development groups impose a style where a line limited to a specified length. This helps see this limit as you type.</li> </ul>		
Toggle ruler line on fill column  See also: <ul style="list-style-type: none"> <li><a href="#">🔗 Filling/Justification</a></li> <li><a href="#">🔗 Spell Checking</a></li> </ul>	<ul style="list-style-type: none"> <li>&lt;f11&gt; h \</li> <li>&lt;f11&gt; &lt;f5&gt; \</li> <li>&lt;f11&gt; 8</li> </ul>	<ul style="list-style-type: none"> <li>(fci-mode &amp;optional ARG)</li> <li>(display-fill-column-indicator-mode &amp;optional ARG)</li> </ul>	Toggle a vertical ruler line shown at the buffer's fill-column. <ul style="list-style-type: none"> <li>For Emacs earlier than 27.1,  it requires the <b>fill-column-indicator external package</b>  activated by PEL <b>use-fill-column-indicator</b> user option set to <b>t</b>.</li> <li>For Emacs 27.1 or later, the built-in <b>display-fill-column-indicator-mode</b> is used.</li> <li>⚠️ <b>fci-mode</b> interferes with pop-up menu displays in terminal-mode, at least with the one used by <b>flyspell-correct-word-before-point</b>: the menu lines become all jagged, they do not line up vertically. The problem does not affect Emacs running in graphics mode.</li> </ul>
<b>Highlight Indentation</b>	 Both require <a href="#">highlight-indentation</a> external package.  PEL activates it when <b>pel-use-highlight-indentation</b> user option is <b>t</b> .		
Highlight indentation based on spaces	<f11> h M-i	(highlight-indentation-mode &optional ARG)	Highlight indentation minor mode highlights indentation based on spaces.
Highlight indentation of current line	<f11> h M-c	(highlight-indentation-current-column-mode &optional ARG)	Highlight Indentation minor mode displays a vertical bar corresponding to the indentation of the current line
<b>Ruler</b>	Shows a ruler at the top of the window indicating several markers: 		
<b>Show the ruler</b>  See also: <a href="#">🔗 Filling/Justification</a>	<f11> b -	(ruler-mode &optional ARG)	Toggle display of ruler in header line (Ruler mode) in window(s) of current buffer. <ul style="list-style-type: none"> <li>With prefix argument ARG, enable Ruler mode if ARG is positive, disable it otherwise.</li> </ul>
	The ruler markers are: <ul style="list-style-type: none"> <li>  Current cursor position</li> <li>! Marks every 5 positions.</li> <li># Current <i>comment-column</i> value, set by <b>C-x</b> ; It controls where the cursor goes when creating a new comment on a line.</li> <li>¶ Current <i>fill-column</i> value, set by <b>C-x f</b>, controls where the cursor wraps.</li> <li>G Current <i>goal-column</i> value, controlled by <b>C-x C-n</b>. The horizontal position when going up and down lines via <b>C-p</b> and <b>C-n</b>. If nil, its G symbol will not show on the ruler and it will not have any impact on the navigation commands. See the information about the <a href="#">set-goal-column</a> command in the navigation table.</li> </ul>		
<b>Automatic Symbol Highlighting</b>	 The following require the <a href="#">auto-highlight-symbol</a> external package.  PEL activates it when <b>pel-use-auto-highlight-symbol</b> user option is <b>t</b> .		
Toggle auto-highlight-symbol mode	<f11> h a	(auto-highlight-symbol-mode &optional ARG)	Toggle Auto Highlight Symbol Mode: in that mode the symbol at point and all its instances in the visible part of the buffer is highlighted after some delay.  More testing and documentation needed for this.
<b>Interactive Highlighting</b>	Highlight Specified Text or Text Matching Regexp (Hi Lock Mode)		
Highlight/un-highlight current line	<f11> h h	(pel-highlight-line &optional CHANGE-COLOR)	Toggle highlighting of the current line with the current PEL line highlight color <ul style="list-style-type: none"> <li>Default color defined by <b>pel-highlight-color-default</b> user-option.</li> <li>Modify current value by pressing command with any key prefix, like <b>C-u</b></li> </ul>
Remove all highlight	<f11> h M-H	(pel-remove-line-highlight)	Remove all highlighting in buffer. <ul style="list-style-type: none"> <li>⚠️ Prompts before proceeding as it also removes bm bookmark highlighting.</li> </ul>
<b>Regexp-based interactive highlighting</b>	The following commands highlight text selected by regular expressions selected in various ways. <ul style="list-style-type: none"> <li>The highlighting is adjusted as text is modified in the buffer. The <b>unhighlight-regexp</b> command removes the previously activated highlighting.</li> </ul>		
<b>Highlight text that match regexp</b>	<ul style="list-style-type: none"> <li>M-s h r</li> <li>&lt;f11&gt; h r</li> </ul>	(highlight-regexp REGEXP &optional FACE)	Set face of each match of REGEXP to FACE. <ul style="list-style-type: none"> <li>Interactively, prompt for REGEXP using 'read-regexp', then FACE. Use the global history list for FACE.</li> </ul>
<b>Highlight entire line that contain text that matches regexp</b>	<ul style="list-style-type: none"> <li>M-s h l</li> <li>&lt;f11&gt; h l</li> </ul>	(highlight-lines-matching-regexp REGEXP &optional FACE)	Set face of all lines containing a match of REGEXP to FACE. <ul style="list-style-type: none"> <li>Interactively, prompt for REGEXP using 'read-regexp', then FACE. Use the global history list for FACE.</li> </ul>
<b>Highlight matches of phrase</b>	<ul style="list-style-type: none"> <li>M-s h p</li> <li>&lt;f11&gt; h p</li> </ul>	(highlight-phrase REGEXP &optional FACE)	Set face of each match of phrase REGEXP to FACE. <ul style="list-style-type: none"> <li>Interactively, prompt for REGEXP using 'read-regexp', then FACE. Use the global history list for FACE.</li> <li>When called interactively, replace whitespace in user-provided regexp with arbitrary whitespace, and make initial lower-case letters case-insensitive, before highlighting with 'hi-lock-set-pattern'.</li> </ul>
<b>Highlight symbol found near point</b>	<ul style="list-style-type: none"> <li>M-s h .</li> <li>&lt;f11&gt; h .</li> </ul>	(highlight-symbol-at-point)	Highlight each instance of the symbol at point. <ul style="list-style-type: none"> <li>Uses the next face from 'hi-lock-face-defaults' without prompting, unless you use a prefix argument.</li> </ul>
<b>Un-highlight</b>	<ul style="list-style-type: none"> <li>M-s h u</li> <li>&lt;f11&gt; h u</li> </ul>	(unhighlight-regexp REGEXP)	Remove highlighting of each match to REGEXP set by hi-lock. <ul style="list-style-type: none"> <li>Interactively, prompt for REGEXP, accepting only regexps previously inserted by hi-lock interactive functions. If REGEXP is t (or if <b>C-u</b> was specified interactively), then remove all hi-lock highlighting.</li> </ul>

Operation	Keystroke	Function	Note
<a href="#">List Highlighted Text</a>	Write a list of currently highlighted text at point with the first command below, then edit if needed and then use the second command to highlight more of the text. These commands can be used to store the highlighting info in comments for persistency, and later re-highlight them using these comments.		
<a href="#">Insert current highlighting regexp/face pairs in buffer at point in comments.</a>	<ul style="list-style-type: none"> <li>M-s h w</li> <li>&lt;f11&gt; h w</li> </ul>	(hi-lock-write-interactive-patterns)	Write interactively added patterns, if any, into buffer at point. <ul style="list-style-type: none"> <li>Interactively added patterns are those normally specified using 'highlight-regexp' and 'highlight-lines-matching-regexp'; they can be found in variable 'hi-lock-interactive-patterns'.</li> </ul>
<a href="#">Extract regexp/face pairs from comments in current buffer and highlight them.</a>	<ul style="list-style-type: none"> <li>M-s h f</li> <li>&lt;f11&gt; h f</li> </ul>	(pel-hi-lock-find-patterns)	Add patterns from the current buffer to the list of hi-lock patterns. <ul style="list-style-type: none"> <li>Does nothing if the current major mode's symbol is a member of the list hi-lock-exclude-modes.</li> </ul>  pel-hi-lock-find-patterns is a thin wrapper over hi-lock-find-patterns that displays a warning if executed when the buffer is not already in hi-lock-mode.
<a href="#">iEdit mode</a>	iEdit Mode - <b>Edit multiple regions in the same way simultaneously.</b> 🙌 This mode is <b>very useful</b> .  Requires the <a href="#">iedit</a> external package.  PEL downloads, installs it when any of the <a href="#">pel-use-iedit</a> or <a href="#">pel-use-lispy</a> user options is set to <b>t</b> .		
Toggle iedit mode See also: <ul style="list-style-type: none"> <li>⌘ Cursor</li> <li>⌘ Search/Replace</li> </ul>	<ul style="list-style-type: none"> <li>C-;</li> <li>&lt;f11&gt; e</li> <li>&lt;f11&gt; h i</li> <li>&lt;f11&gt; m i</li> </ul>	(iedit-mode &optional ARG)	Toggle iEdit mode: <b>edit all symbols in scope or region simultaneously.</b>  Both iEdit and Flyspell use the C-; key as their default binding. PEL detects and reports that situation: modify the binding of one of them if you see it. ➤ See <a href="#">⌘ Search/Replace</a> where all the iedit-mode commands are described.
<a href="#">Variables</a> hi-lock-file-patterns-policy	Controls whether Hi Lock mode should automatically extract and highlight patterns found in a file when it is visited. Its value can be nil (never highlight), ask (query the user), or a function. If it is a function, hi-lock-find-patterns calls it with the patterns as argument; if the function returns non-nil, the patterns are used. The default is ask. Patterns are always highlighted if you call hi-lock-find-patterns directly, regardless of the value of this variable.		

### Highlight — References

Topic & Link	Description & Notes
<a href="#">Gnu Emacs - 14.13 - Interactive Highlighting</a>	Describes interactive highlighting, a topic for most of the commands listed above.
<a href="#">The Definitive Guide To Syntax Highlighting</a>	A blog, written in 2104, that provides an overview of the highlighting that can be done with Emacs.
<a href="#">e2ansi, syntax highlighting for less, powered by Emacs</a>	Using Emacs to syntax highlight less output.