Highlighting Text

Operation	Keystroke	Function	Note
Highlighting	Note: All text highlight commands listed in this table affects the current buffer, therefore PEL binds the keys to the pel:buffer prefix: **<f11> b**		
Toggle syntax highlighting in current buffer (see also: ∑ Faces/Fonts)	**<f11> b h F**	(**font-lock-mode** &optional ARG)	Toggle syntax highlighting in this buffer (Font Lock mode). • With a prefix argument ARG, enable Font Lock mode if ARG is positive, and disable it otherwise.
Highlight Blocks	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).		
Enable/Disable highlighting matching block (),{},[]	**<f11> b h (**	(**show-paren-mode** &optional ARG)	Toggle visualization of matching parens (Show Paren mode). • With a **C-u** 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.
Enable/Disable coloured highlight of nested blocks (),{},[]	**<f11> b h R**	(**rainbow-delimiters-mode** &optional ARG)	Highlight nested parentheses, brackets, and braces with different colours according to their depth. • Customize the depth and colours with **M-x customize-group rainbow-delimiters** 📦 **Requires:** rainbow-delimiters.el PEL activates this when the **pel-use-rainbow-delimiters** user option is set to **t**.
Highlight Modes	Note: All highlighting in a buffer is non persistent: it is forgotten when buffer is killed.		
Enable/Disable Highlight Changes Mode	**<f11> b h C**	(**highlight-changes-mode** &optional ARG)	Toggle highlighting changes in this buffer (Highlight Changes mode). • With a prefix argument ARG, enable Highlight Changes mode if ARG is positive, and disable it otherwise. • In this mode the most recent changes in the buffer are highlighted. • However, when this is enabled, the syntax highlighting does not work properly for the new text.
Highlight Current Line	Highlighting the current line may help to find the cursor when editing with big windows. These commands control line highlighting.		
Toggle line highlight mode	• **<f11> b h -** • **<f11> 0**	(**hl-line-mode** &optional ARG)	Toggle highlighting of the current line (Hl-Line mode). • With a prefix argument ARG, enable Hl-Line mode if ARG is positive, and disable it otherwise. • When same buffer is shown in several windows, the highlighting might show in each of them. Change that with pel-toggle-hl-line-sticky with: **<f11> b h s** ☝ A quick way to find where your cursor is located is to hit **<f11> 0** 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 **<f1> k** to get help for a specific key binding and see all the key bindings for a command, allowing you to discover its other bindings.)
Set color of highlight line	**<f11> b h c**	(**pel-set-highlight-color** COLORNAME)	Set the colour of the highlight line used in the line highlight mode. • Use **<f11> ? d c** to list all colours and their names.
Toggle line highlighting affecting all windows	**<f11> b h s**	(**pel-toggle-hl-line-sticky**)	Toggle current line highlight to all windows or just the current one. • Toggles the value of 'hl-line-sticky-flag' between t and nil.
Highlight current column	The following command provide a vertical line across the entire window at the cursor location. Useful when creating tables or checking indentation manually. ☛ vline also provides the vline-global-mode to activate the vertical line in all buffers, but PEL does not provide a binding for it because it slows Emacs down too much.		
Toggle Vline Mode (See also: ∑ Hide/Show Code)	**<f11> b h |**	(**vline-mode** &optional ARG)	Toggle the display of a vertical line spanning the entire window at the cursor column. 📦 Requires: vline.el PEL activates this when **pel-use-vline** user option is **t**.
Interactive Highlighting	Highlight Specified Text or Text Matching Regexp (Hi Lock Mode)		
Enable/disable highlight lock mode	**<f11> b h L**	(**hi-lock-mode** &optional ARG)	Toggle selective highlighting of patterns (Hi Lock mode). • With a prefix argument ARG, enable Hi Lock mode if ARG is positive, and disable it otherwise. • When text is already highlighted toggling the mode off will remove all highlighting; turning it back on will not restore the highlighting.
Enable highlight lock mode to all buffers	**<f11> b h G**	(**global-hi-lock-mode** &optional ARG)	Toggle Hi-Lock mode in all buffers. • With prefix ARG, enable Global Hi-Lock mode if ARG is positive; otherwise, disable it.
Highlight text that match regexp	• **M-s h r** • **<f11> b h r**	(**highlight-regexp** REGEXP &optional FACE)	Set face of each match of REGEXP to FACE. • Interactively, prompt for REGEXP using 'read-regexp', then FACE. Use the global history list for FACE.
Highlight entire line that contain text that matches regexp	• **M-s h l** • **<f11> b h l**	(**highlight-lines-matching-regexp** REGEXP &optional FACE)	Set face of all lines containing a match of REGEXP to FACE. • Interactively, prompt for REGEXP using 'read-regexp', then FACE. Use the global history list for FACE.
Highlight matches of phrase	• **M-s h p** • **<f11> b h p**	(**highlight-phrase** REGEXP &optional FACE)	Set face of each match of phrase REGEXP to FACE. • Interactively, prompt for REGEXP using 'read-regexp', then FACE. Use the global history list for FACE. • 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'.
Highlight symbol found near point	• **M-s h .** • **<f11> b h .**	(**highlight-symbol-at-point**)	Highlight each instance of the symbol at point. • Uses the next face from 'hi-lock-face-defaults' without prompting, unless you use a prefix argument.
Un-highlight	• **M-s h u** • **<f11> b h u**	(**unhighlight-regexp** REGEXP)	Remove highlighting of each match to REGEXP set by hi-lock. • Interactively, prompt for REGEXP, accepting only regexps previously inserted by hi-lock interactive functions. If REGEXP is t (or if **C-u** was specified interactively), then remove all hi-lock highlighting.
List Highlighted Text	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.		
Insert current highlighting regexp/face pairs in buffer at point in comments.	• **M-s h w** • **<f11> b h w**	(**hi-lock-write-interactive-patterns**)	Write interactively added patterns, if any, into buffer at point. • 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'.

Operation	Keystroke	Function	Note
Extract regexp/face pairs from comments in current buffer and highlight them.	• **M-s h f** • **<f11> b h f**	(**pel-hi-lock-find-patterns**)	Add patterns from the current buffer to the list of hi-lock patterns. • Does nothing if the current major mode's symbol is a member of the list hi-lock-exclude-modes.  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.
Variables			
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. Note that 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
Gnu Emacs - 14.13 - Interactive Highlighting	Describes interactive highlighting, a topic for most of the commands listed above.
The Definitive Guide To Syntax Highlighting	A blog, written in 2104, that provides an overview of the highlighting that can be done with Emacs.