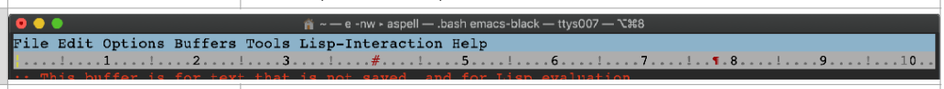
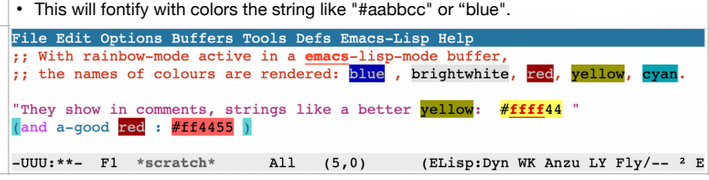
↖ Topic Index

Highlighting

Operation	Keystroke	Function	Note

Highlighting
• ruler mode
• toggle font locking
• highlight blocks
• blink cursor
• highlight recent changes
• highlight current line
• highlight current column
• highlight fill column
• highlight indentation
• highlight symbols
• highlight selected line(s)
• highlight with regexp
  • list highlighted text
• iedit mode
• rainbow-mode : color codes
• Variables
• Reference
 	
Last updated on:
	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.
	PEL provides the pel-pkg-for-highlight customization group to control some aspect of highlighting. The user options are:
	• **pel-use-auto-highlight-symbol** activates the **auto-highlight-symbol** external package.
	• **pel-use-beacon-mode** activates **beacon** external package automatically or under user action, to "*flash*" highlight the cursor.
	• **pel-use-iedit** activates **iedit** external package which allows you to interactively edit all symbols inside current scope or marked region.
	• **pel-use-fill-column-indicator** activates the **fill-column-indicator** external package for Emacs versions earlier than 27.1.
	  • This highlights a vertical line on the fill-column, the column where automatic line wrapping occurs (when the mode is active).
	• **pel-use-highlight-indentation** activates the **highlight-indentation** external package that helps see the indentation.
	• **pel-use-vline** activates **vline.el** external package to create a vertical bar across the entire buffer that moves point (the cursor).
	• **pel-use-rainbow-delimiters** activates **rainbow-delimiters.el** external package that highlights matching parens delimiters.
	• **pel-use-rainbow-mode** activates **rainbow-mode** external package which **highlights strings containing color codes**.
	• The builtin **show-paren-mode** provides matching parens highlighting.
	PEL provides a set of key bindings to various highlight control commands that are regrouped under the pel:highlight prefix: **<f11> h**.
	• They are listed in the table below along the native Emacs key bindings for highlight control.
	
	2025-12-09		

Open this PDF file. See also: **Σ Help/Info**	**<f11> h <f1>**	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the **Σ Highlight** local PDF. If the prefix argument (like **C-u** or **M--**) is used, then it opens the remote GitHub hosted raw PDF instead. If the **pel-flip-help-pdf-arg** user-option is set it's the other way around.
Σ Customize PEL highlighting control	**<f11> h <f2>**	(**pel-customize-pel** &optional OTHER-WINDOW)	Customize the following PEL support: 1. PEL support for highlighting (pel-pkg-for-hightlight) 2. PEL support for mode line (pel-pkg-for-modeline) 3. PEL support for parens (pel-pkg-for-parens) • If OTHER-WINDOW is non-nil (use **C-u**), display in other window.
Σ Customize Emacs highlighting control	**<f11> h <f3>**	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs support for buffer highlight management: **auto-highlight-symbol**, **iedit**, **hl-line**, **highlight-indentation**, **paren-showing**, **rainbow-mode**, **rainbow-delimiters**, **smartparens**, **vline**, **fill-column-indicator** (Emacs < 27.1), **beacon** (Emacs >= 27.1)
Ruler Mode	Shows a ruler at the top of the window indicating several markers:		
Show the ruler	**<f11> b -**	(**ruler-mode** &optional ARG)	Toggle display of ruler in header line (Ruler mode) in window(s) of current buffer. • With prefix argument ARG, enable Ruler mode if ARG is positive, disable it otherwise.
The **ruler markers** are: See also: • **Σ Filling/Justification** • Ruler Mode @ Youtube	| Current cursor position. **!** Marks every 5 positions. **#** Current *comment-column* value, set by **C-x ;** It controls where the cursor goes when creating a new comment on a line. **¶** Current *fill-column* value, set by **C-x f**, controls where the cursor wraps. **G** Current *goal-column* value, controlled by **C-x C-n**. The horizontal position when going up and down lines via **C-p** and **C-n**. 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 set-goal-column command in the navigation table.		
Font Locking Control	Use these commands to toggle font locking in the buffer or globally. See also **Font Lock Mode** section of the **Emacs Lisp Manual**		
Toggle syntax highlighting in current buffer See also: **Σ Faces/Fonts**	**<f11> 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> h (** **<f12> h (**	(**show-paren-mode** &optional ARG)	Toggle visualization of matching parens (Show Paren mode). • With **C-u** prefix argument, enable Show Paren mode if positive, disable it otherwise. ☛ **<f12> h (** bindings are available for several modes.
Show paren-mode info	**<f11> h ? (**	(**pel-show-paren-info** &optional APPEND)	Show 'paren-mode' control user-options in *pel-highlight-info* buffer. • With non-nil optional APPEND argument, append the information to the buffer.
Enable/Disable coloured highlight of nested blocks: (), {}, []	**<f11> h)** **<f12> h)**	(**rainbow-delimiters-mode** &optional ARG)	Highlight nested parentheses, brackets and braces with depth-specific colours. • Customize the depth and colours with **M-x customize-group rainbow-delimiters:** with PEL access its customization group with **<f11> h <f3> 8** key sequence. Requires: **rainbow-delimiters.el** activated by **pel-use-rainbow-delimiters**
Blink Cursor ⚠ (Emacs >=27)	Requires external **beacon package**. PEL activates when **pel-use-beacon-mode** is turned on from start or on request. If **pel-use-beacon-mode** is set to **t**: you must toggle the mode on explicitly. If it is set to **use-from-start**, PEL activates it automatically.		
Toggle beacon-mode	**<f11> h M-b**	(**beacon-mode** &optional ARG)	Toggle the beacon-mode. This highlights the cursor changing focus. the key binding is only active when **pel-use-beacon-mode** is not nil.
Blink beacon at cursor	**<f11> h b**	(**beacon-blink**)	Blink the beacon at the position of the cursor. Even when beacon-mode is not active.
Highlight Recent Changes	This mode is a quick way to highlight what has been recently changed in the buffer. ⚠ All highlighting in a buffer is **non persistent**: it is forgotten when buffer is killed.		
Enable/Disable Highlight Changes Mode	**<f11> h M-c**	(**highlight-changes-mode** &optional ARG)	Toggle highlighting changes in this buffer (Highlight Changes mode). • With prefix argument, enable Highlight Changes mode if it's positive, else disable it. • In this mode the most recent changes in the buffer are highlighted. • ⚠ when enabled, 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 ☝ Use **<f11> 0** twice to locate your cursor. See also: **Σ Cursor**	• **<f11> h 0** • **<f11> 0**	(**hl-line-mode** &optional ARG)	Toggle highlighting of the current line (Hl-Line mode) in the current buffer. With prefix argument ARG, enable Hl-Line mode if ARG is positive, 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> h s**		
Toggle line highlighting affecting all windows	**<f11> h s**	(**pel-toggle-hl-line-sticky**)	Toggle current line highlight to all windows with current buffer or just the current one. • Toggles the value of '**hl-line-sticky-flag**' between t and nil.
Change color of highlight line for session	**<f11> h c**	(**pel-set-highlight-color** COLORNAME)	Set the colour of the highlight line used in the line highlight mode (affects all buffers). • Prompt for color name, use tab completion to see available colour names.
	• The change does not persist when Emacs is closed. To select a persistent color, then customize the **highlight** user option (see next row).		
Σ Customize highlight color and attributes permanently	**<f11> h <f4>**	(**pel-customize-highlight**)	Customize the **highlight** user option color and other attributes.
	• As with all customizations, you can activate the change for this Emacs session or save it to make it persist across Emacs sessions. • 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.		

Operation	Keystroke	Function	Note
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; PEL has no binding for it because it slows Emacs too much.		
Toggle Vline Mode See also: Σ **Cursor** Σ **Hide/Show**	• **<f11> h |** • **<f11> 9**	(**vline-mode** &optional ARG)	Toggle the display of a vertical line spanning the entire window at the cursor column. 📦 Requires: **vline.el** activated when **pel-use-vline** user option is **t**.
Highlight Fill Column	The fill column identifies the end of line where line wrapping occurs. • Some development groups impose a style where a line limited to a specified length. This helps see this limit as you type.		
Toggle ruler line on fill column See also: • Σ **Filling/Justification** • Σ**Spell Checking**	• **<f11> h ** • **<f11> <f5> ** • **<f11> 8**	• (**fci-mode** &optional ARG) • (**display-fill-column-indicator-mode** &optional ARG)	Toggle a vertical ruler line shown at the buffer's fill-column. • For Emacs earlier than 27.1, 📦 it requires the **fill-column-indicator external package** activated by PEL **use-fill-column-indicator** user option set to **t**. • For Emacs 27.1 or later, the built-in **display-fill-column-indicator-mode** is used. • ⚠ **fci-mode** interferes with pop-up menu displays in terminal-mode, at least with the one used by **flyspell-correct-word-before-point:** the menu lines become all jagged, they do not line up vertically. The problem does not affect Emacs running in graphics mode.
Highlight Indentation	📦 Require **highlight-indentation** external package. PEL activates it when **pel-use-highlight-indentation** user option is **t**.		
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
Symbol Highlighting	📦 Require **auto-highlight-symbol** external package. PEL activates it when **pel-use-auto-highlight-symbol** user option is **t**.		
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.
Highlight Interactively	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 • Default color defined by **pel-highlight-color-defaul**t user-option. • Modify current value by pressing command with any key prefix, like **C-u**
Remove all highlight in buffer	**<f11> h M-H**	(**pel-remove-line-highlight**)	Remove all highlighting in buffer. ⚠ Also removes visible bookmarks. • Prompts before proceeding as it also removes visible bm bookmark highlighting.
Regexp-based interactive highlighting	The following commands highlight text selected by Emacs Lisp regular expressions selected in various ways. • The highlighting is adjusted as text is modified in the buffer. The **unhighlight-regexp** command removes the previously activated highlighting.		
Enable/disable highlight lock mode - highlight text matching specified regexp	**<f11> h L**	(**hi-lock-mode** &optional ARG)	Toggle selective highlighting of patterns (Hi Lock mode). • With prefix argument, enable Hi Lock mode if ARG is positive, 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> h M-L**	(**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> 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> 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> 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> 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> 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	When using regex-based interactive highlighting, 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> 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'.
Extract regexp/face pairs from comments in current buffer and highlight them.	• **M-s h f** • **<f11> 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.
iEdit mode	iEdit Mode - **Edit multiple regions in the same way simultaneously.** ☝ This mode is **very useful**. 📦Requires the **iedit** external package. PEL downloads, installs it when any of the **pel-use-iedit** or **pel-use-lispy** user options is set to **t**.		
Toggle iedit mode See also: • Σ **Cursor** • Σ **Search/Replace**	• **C-;** • **<f11> e** • **<f11> h e** • **<f11> m e**	(**iedit-mode** &optional ARG)	Toggle iEdit mode: **edit all symbols in scope or region simultaneously**. ⚠ 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 Σ **Search/Replace** where all the iedit-mode commands are described.
Visualize Color Codes	**Using rainbow mode:** highlight the color of color codes. ☝ See color code names in rainbow customization group.		
Toggle rainbow mode on/off 📦 Requires: **rainbow-mode** activated when **pel-use-raibow-mode** is **t**.	**<f11> h M-r**	(**rainbow-mode** &optional ARG)	Colorize strings that represent colors. • This will fontify with colors the string like "#aabbcc" or "blue". 
	☝ When **rainbow-mode** 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). • With **overwrite-mode** on you can change the hex color values and see the impact right away!		

Operation	Keystroke	Function	Note
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. 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.
e2ansi, syntax highlighting for less, powered by Emacs	Using Emacs to syntax highlight less output.