↖ Topic Index

Faces / Fonts

Operation	Keystroke	Function	Note

Emacs Faces, Text Faces and Fonts
○ Help & customization
• Get info about faces
• face-explorer
  • tooltip for face
  • simulate other display
• Query about point
• macOS color table
○ Change font size
• Set frame font
• Font lock
• Font Tools
  • font-lock-studio
	Emacs major mode code controls the various faces used to represent various text elements.
	• Each face corresponds to the way text is rendered according to the environment's ability and the selected default font, colours, etc...
	• If you do not like the default selected font, colour, style, etc... you can customize them to your liking.
	
	This table describes the various commands related to Emacs faces and fonts.
	
	PEL provides access to the following external packages when the corresponding **pel-use-** customizable user-option is turned on:		

	📦 **face-explorer**	**pel-use-face-explorer**	Library and tools for faces and text properties
	The following external packages are very useful when writing a major mode where you need to define faces for various keywords and concepts:		
	📦 **faceup**	**pel-use-faceup**	Regression test system for Emacs font-lock keywords
	📦 **font-lock-profiler**	**pel-use-font-lock-profiler**	Coverage and timing tool for font-lock keywords
	📦 **font-lock-studio**	**pel-use-font-lock-studio**	Debugger for Font Lock keywords
Last updated on:	2026-05-03		
Open this PDF file. See also: **Help/Info**	**<f11> C-f <f1>**	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the **Face/Font** 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.
Customize PEL Face/Font support. See also: **Customize**	**<f11> C-f <f2>**	(**pel-customize-pel** &optional OTHER-WINDOW)	Customize PEL face/font management. • If OTHER-WINDOW is non-nil (use **C-u**), display in other window.
Customize Emacs Face/Font control **See also: Customize**	**<f11> C-f <f3>**	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs support for the face/font management: face-explorer
Show/customize face at point ☝ To see or change the definition of a face used at point, type: **C-u <f11> C-f C-f**	• **<f11> C-f C-f** • **<f11> ? d a**	(**pel-show-face-at-point** &optional EDIT)	Display information about face at point in the echo area. • With **C-u** optional argument or EDIT non-nil, customize the face found at point. ☝ Note that a face can inherit from another. You may need to disable this inheritance to change colour of some faces or create inheritance to impose consistencies. This is done in the customization.
Show available colours	• **<f11> C-f c** • **<f11> ? d c**	(**list-colors-display** &optional LIST BUFFER-NAME CALLBACK	Display names of defined colors, and show what they look like.
See faces currently defined	• **<f11> C-f f** • **<f11> ? d F**	(**list-faces-display** &optional REGEXP)	List all available faces in a buffer. • Very useful to see the capabilities of text display in the emacs process used. This also helps to see if the background colour is OK with the faces being used. • The faces show the various colours and fonts used for various type of text. • By clicking on each face character list, another window shows up to provide more info on the face and provides ability for even more customization of each face.
Show hierarchy of all faces currently loaded	• **<f11> C-f h** • **<f11> ? e f**	(**pel-hier-face**)	Display all loaded faces in the current frame and their inheritance relationship. • The faces hierarchy is printed inside a hierarchy-tabulated buffer. • Each face name is a link to its implementation code.
face-explorer	📦 Requires **face-explorer** activated by **pel-use-face-explorer**		
Print information about all currently loaded faces	**<f11> C-f C-e C-e**	(**face-explorer-list-faces**)	List all faces, with samples for the selected frame and fictitious display. • Shows how a face is defined, for example if the currently active themes affect the face.
Print information about specified face	**<f11> C-f C-e C-d**	(**face-explorer-describe-face** FACE)	Print information about FACE, including the origin of all attributes. Prompt for face.
Print face property information for face at point	**<f11> C-f C-e C-p**	(**face-explorer-describe-face-prop** FACE-TEXT-PROPERTY)	Print information about FACE-TEXT-PROPERTY of face at point.
Show currently supported face features	**<f11> C-f C-e C-f**	(**face-explorer-list-display-features**)	Show which features the current display supports.
Print examples of faces specifications	**<f11> C-f C-e e**	(**face-explorer-list-face-prop-examples**)	List sample text with face spec in various variants.
Print examples of faces specifications with overlays	**<f11> C-f C-e E**	(**face-explorer-list-overlay-examples**)	List a number of examples with 'face' overlays and text properties.
Toggle mode to show tooltip of face at point	**<f11> C-f C-e C-t**	(**face-explorer-tooltip-mode** &optional ARG)	Minor mode to show tooltips for face-related text properties.
Toggle simulation of other display type	**<f11> C-f C-e C-s**	(**face-explorer-simulate-display-mode** &optional ARG)	Toggle a minor mode to simulate another display type.
• **Set number of colours for the fictitious display**	**C-c ! #**	(**face-explorer-set-number-of-colors** NUMBER-OF-COLORS)	Set number of the fictitious display to NUMBER-OF-COLORS.
• **Increase number of colours**	**C-c ! +**	(**face-explorer-increase-number-of-colors**)	Increase number of colors in the 'face-explorer' list. • The numbers are increased to the next of 8, 16, 256, and t (representing infinite).
• **Decrease number of colours**	**C-c ! -**	(**face-explorer-decrease-number-of-colors**)	Decrease number of colors in the 'face-explorer' list. • The numbers are decreased to 8, 16, and 256.
• **Toggle background mode**	**C-c ! b**	(**face-explorer-toggle-background-mode**)	Toggle background mode in the 'face-explorer' list.
• **Select next color class**	**C-c ! c**	(**face-explorer-next-color-class**)	Select next color class in the 'face-explorer' list. • The color class can be 'color', 'grayscale', or 'mono'.
• **Toggle between graphics and tty**	**C-c ! g**	(**face-explorer-toggle-window-system**)	Toggle window system between current and 'tty'. • On a tty, this does nothing.
• **Reset fictitious display**	**C-c ! r**	(**face-explorer-reset-fictitious-display**)	Make the fictitious display like the selected frame.
Query info about point **Show information about current character (including face information)** See also: **Help/Info** , **Input Method**	• **C-x =** • **<f11> ? d p**	(**what-cursor-position** &optional DETAIL)	Displays information about character at point in the echo area. ☝ With any prefix argument opens a *Help* buffer and show the complete information of character at point with all properties, face, encoding, etc. • Type: **C-u C-x =** • With PEL, you can also type: **C-- C-x =**
	<f11> ? d P	(**pel-what-cursor-position**)	Same as above but always display the complete information.
Open macOS color table dialog	**⌘-C**	(**ns-popup-color-panel** &optional FRAME)	On macOS in graphics mode only: opens the color table dialog.
Open macOS font table dialog	**⌘-t**	(**ns-popup-font-panel** &optional FRAME)	On macOS in graphics mode only: opens the font table dialog.

Operation	Keystroke	Function	Note

Modify Font Size (Text Scale)	• The standard Emacs commands to change font size of the current buffer are the **C-x C-+**, **C-x C- -** and **C-x C-0**. When using those, after the first **C-+** (or **C-=**) you can continue to change the font size by repeating the **C-+**, **C-=** or **C-0**).
	  • These commands do not work when Emacs is used in text terminal mode.
	• The macOS terminal supports **⌘-+**, **⌘--** and **⌘-0** commands to control the font size of the terminal window. This allows changing the font size for everything in Emacs running in terminal mode under macOS.
	• The PEL package also
	  • binds **⌘-+**, **⌘--** and **⌘-0** to the font control commands (of current buffer) when Emacs runs in graphics mode, and binds the keypad equivalent keys to control the font size of **all** buffers.
	⚠️ When Emacs is running in graphics mode, these commands do not affect the font of other windows such as the echo area or the mini buffer. For that you must change the font of the entire frame.		

Increase font size of current buffer	• **C-x C-+** • **C-x C-=** • **s-+** • **⌘-+**	(**text-scale-adjust** INC)	Adjust the height of the default face by INC. • INC may be passed as a numeric prefix argument. • After the first **C-+** (or **C-=**) you can continue to change the font size by repeating the **C-+**, **C-=** or **C-0**).
Decrease font size of current buffer	• **C-x C- -** • **s--** • **⌘--**	(**text-scale-adjust** INC)	
Reset font size of current buffer to default	• **C-x C-0** • **s-0** • **⌘-0**	(**text-scale-adjust** INC)	
Increase font size of all buffers	• **s-<kp-add>** • **⌘-<kp-add>**	(**pel-font-increase-size-all-buffers** ALL)	Increase the font size of all buffers. For Emacs running on graphics mode only. • By default the font size of Minibuffers and Echo Area buffers are not changed. To change them as well, set the ALL argument using **C-u** prefix or any numeric argument.
Decrease font size of all buffers	• **s-<kp-subtract>** • **⌘-<kp-subtract>**	(**pel-font-decrease-size-all-buffers** ALL)	Decrease the font size of all buffers. For Emacs running on graphics mode only. • By default the font size of Minibuffers and Echo Area buffers are not changed. To change them as well, set the ALL argument using **C-u** prefix or any numeric argument.
Reset font size of all buffers	• **s-<kp-0>** • **⌘-<kp-0>**	(**pel-font-reset-size-all-buffers** ALL)	Reset the font size of all buffers to their default size. For Emacs running on graphics mode only. • By default the font size of Minibuffers and Echo Area buffers are not changed. To change them as well, set the ALL argument using **C-u** prefix or any numeric argument. ☝ Useful to set the same font size to all buffers when their font size differ, maybe even prior to increase or decrease the font size of all buffers.
Set Frame Font	With Emacs running in graphics mode, you can change the font of all windows with the menu-set-font command.		
Change font of current Frame See also: **Σ Faces/Fonts**	**<f11> F F**	(**menu-set-font**)	Interactively select a font and font size and make it the default on all frames. • The selected font will be the default on both the existing and future frames in the current session. • It is not persistent, so next time you start Emacs the default font is used.
Font Locking			

Toggle syntax highlighting in current buffer
 	
See also: **Σ Highlight**
	<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. If called from Lisp, enable the mode if ARG is omitted or nil.
			• When Font Lock mode is enabled, text is fontified as you type it:
			  • Comments are displayed in 'font-lock-comment-face';
			  • Strings are displayed in 'font-lock-string-face';
			  • Certain other expressions are displayed in other faces according to the value of the variable 'font-lock-keywords'.

Font lock block	**M-o M-o**	(**font-lock-fontify-block** &optional ARG)	Fontify some lines the way 'font-lock-fontify-buffer' would. The lines could be a function or paragraph, or a specified number of lines. • If ARG is given, fontify that many lines before and after point, or 16 lines if no ARG is given and 'font-lock-mark-block-function' is nil. • If 'font-lock-mark-block-function' non-nil and no ARG is given, it is used to delimit the region to fontify.
Font Tools	The following Emacs Lisp packages are very useful to debug font-locking code when you write major mode lisp code: • 📦 **font-lock-studio** • 📦 **font-lock-profiler** • 📦 **faceup** Each one can be installed by setting the corresponding **pel-use-** user-option to t. Use **<f11> C-f <f2>** to access the customization buffer for that.		
Analyze font rendering with font-lock-studio	**<f11> C-f C-s**	(**font-lock-studio** &optional ARG)	Interactively debug the font-lock keywords of the current buffer. • With C-u prefix (when ARG is non-nil), create a new, unique, interface buffer. 📦 Requires **font-lock-studio** activated by **pel-use-font-lock-studio**

Faces/Fonts — References

Topic & Link	Description
GNU Emacs Manual- Fonts	
GNU Emacs Manual - Text Faces	
Set Fonts @ EmacsWiki	The EmacsWiki and ErgoEmacs page provide a lot of valuable information.
Font Setup @ ErgoEmacs	
How to set the default font size?	Discussion on StackExchange/Emacs
How to set the font size in Emacs?	Discussion on StackOverflow
Cursor Management	
Changing Cursor Dynamically @ Emacs Wiki	Describes cursor_chg.el witch changes the cursor color & shape depending on context
* **cursor-chg.el**	
Major Mode Highlighting	**Overcolorization**, by Andrey Listopadov raises an interesting pointy about color highlighting done by major modes. Andrey promotes the idea that highlighting should be used to help you notice important aspects of the code, not just colorized it.