






Emacs support for the Go Programming Language 🚧

Description	Keystroke	Function	Note
<h2>Go Support</h2> <ul style="list-style-type: none">Help & Customization<ul style="list-style-type: none">InfoNavigationIndentationSyntax Check			<p>Support for the Go programming language is described in this page.</p> <p> Go support requires the go-mode external package, which supports the classic go-mode and tree-sitter based go-ts-mode.</p> <p> PEL supports it when pel-use-go and pel-use-go-mode user options are turned on.</p> <ul style="list-style-type: none">On Emacs >= 30, PEL supports tree-sitter if pel-use-tree-sitter is set to t.<ul style="list-style-type: none">You can activate tree-sitter for Go by setting pel-use-go to 'with-tree-sitter' (as long as pel-use-tree-sitter is t and Emacs >= 30). See Tree SitterFiles with the .go extensions are recognized as Go source files and use the go-mode major mode,Speedbar support for .go files listing functions and types. See Speedbar for more info about it.Automatic execution of gofmt when saving a buffer into a file when  pel-use-gofmt-on-buffer-save user option is set to t.Generic programming language features like template text insertion handle Go comment style. See Inserting Text.Control of the tab width for all go files, via the pel-go-tab-width user-option (access the PEL customer buffer with the <f11> SCP g <f2> key sequence or <f12> <f2> from inside a buffer visiting a Go source code file.Support for syntax checking with either flymake or flycheck via the goflymake Go program when  pel-use-goflymake user option is set to t. <p>All support requires support for the Go programming language installed on your computer.</p> <ul style="list-style-type: none">See Go installation instructions or use Homebrew's command brew install go.
Last updated on:	2025-10-10		
Open this PDF file. See also: Tree Sitter Help/Info	<f11> SPC g <f1> <f12> <f1>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the PEL - Go 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 Go support	<f11> SPC g <f2> <f12> <f2>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL Go support. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
Customize Emacs Go support	<f11> SPC g <f3> <f12> <f3>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs Go support: go, go-cover, godoc, go-dot-mod. <ul style="list-style-type: none">If OTHER-WINDOW is non-nil (use C-u), display in another window.
Show PEL setup for Go	<f11> SPC g ? <f12> ?	(pel-go-setup-info)	Display Go setup information: <ul style="list-style-type: none">tab widthwhether gofmt is executed before saving buffer.
Describe expression at point.	C-c C-d	(godef-describe POINT)	Describe the expression at POINT.  This uses the godef executable , a Go program. <ul style="list-style-type: none">To install it, run the following command from a shell: <code>go get github.com/rogpeppe/godef</code>.The GOPATH environment variable must be setup and GOPATH/bin must be in the PATH to be able to run godef.
Set visual rendering of hard tabs for the current buffer	<f11> M-t	(pel-set-tab-width N)	Change the tab width of the current buffer, only affecting the display rendering of hard tabs inserted in the buffer text. Prompts for a new value in the [2, 8] range. <ul style="list-style-type: none">This modifies a buffer local value of the the tab-width user-option.The change is temporary and affects the current buffer only.To change the tab width used for all Go source code files, change the 'pel-go-tab-width' user-option variable instead. See Tree Sitter Indentation for more information.
Toggle gofmt run on buffer save	<f11> SPC g M-s <f12> M-s	(pel-go-toggle-gofmt-on-buffer-save &optional GLOBALLY)	Toggle automatic run of gofmt when saving Go buffer to file. <ul style="list-style-type: none">By default change behaviour for local buffer only.When GLOBALLY argument is non-nil, change it for all Go buffers for the current Emacs editing session (the change does not persist across Emacs sessions).To modify the global state permanently modify the customized value of the  pel-go-toggle-gofmt-on-buffer-save user option via the 'pel-pkg-for-go'group customize buffer.
Inserting code	See also: Tree Sitter Inserting Text generic commands that apply to go buffers.		
Add new import package to list of module package import statement	C-c C-a	(go-import-add ARG IMPORT)	Add a new IMPORT to the list of imports. Don't move point. <ul style="list-style-type: none">When called with a prefix ARG asks for an alternative name to import the package as.If no list exists yet, one will be created if possible.If an identical import has been commented, it will be uncommented, otherwise a new import will be added.
Navigation See also: Tree Sitter Navigation generic commands that apply to go buffers. The main commands are shown here but more are available and described there.			
Move to expression definition	C-c C-j	(godef-jump POINT &optional OTHER-WINDOW)	Jump to the definition of the expression at POINT. <ul style="list-style-type: none">after that command, use M- , to go back to original point.
Move to expression definition in other window	C-x 4 C-c C-j	(godef-jump-other-window POINT)	Jump to the definition of the expression at POINT but into the other window. <ul style="list-style-type: none">after that command, use M- , to go back to original point.
Move to current function arguments	C-c C-f a	(go-goto-arguments &optional ARG)	Go to the arguments of the current function. <ul style="list-style-type: none">If ARG is non-nil, anonymous functions are skipped.
Move to current function docstring	C-c C-f d	(go-goto-docstring &optional ARG)	Go to the top of the docstring of the current function. <ul style="list-style-type: none">If there is none, add one beginning with the name of the current function.Anonymous functions do not have docstrings, so when this is called interactively anonymous functions will be skipped. If called programmatically, an error is raised unless ARG is non-nil.
Move to function definition	C-c C-f f	(go-goto-function &optional ARG)	Go to the function definition (named or anonymous) surrounding point. <ul style="list-style-type: none">If we are on a docstring, follow the docstring down.If no function is found, assume that we are at the top of a file and search forward instead.If point is looking at the func keyword of an anonymous function, go to the surrounding function.If ARG is non-nil, anonymous functions are ignored.
Move to imports statement	C-c C-f i	(go-goto-imports)	Move point to the block of imports. <ul style="list-style-type: none">If using<pre>import ("foo" "bar")</pre>it will move point directly behind the last import.If using<pre>import "foo" import "bar"</pre>it will move point to the next line after the last import.If no imports can be found, point will be moved after the package declaration.
Move to current method receiver	C-c C-f m	(go-goto-method-receiver &optional ARG)	Go to the receiver of the current method. <ul style="list-style-type: none">If there is none, add parenthesis to add one.Anonymous functions cannot have method receivers, so when this is called interactively anonymous functions will be skipped. If called programmatically, an error is raised unless ARG is non-nil.
Move to current function name	C-c C-f n	(go-goto-function-name &optional ARG)	Go to the name of the current function. <ul style="list-style-type: none">If the function is a test, place point after 'Test'.If the function is anonymous, place point on the 'func' keyword.If ARG is non-nil, anonymous functions are skipped.

