Code/Markup Template Support

Code Templates	
Code Templates **Emacs:** • **Using Skeletons** • **Skeleton Language** • **Tempo** See also: ∑ **Inserting Text**	PEL provides **Emacs Tempo Skeletons** specialized for some programming and markup languages, listed in this table. • The format of the text inserted by these templates is specialized to each context but is also customizable through a set of PEL user-option variables that are listed in the tables. Some of the concepts apply to several languages, some only apply to a very specific one. • The key bindings to insert text from the templates use the **\<f12\> \<f12\>** key prefix for the supported major modes. • The key binding to access the customization group that defines the layout of the skeleton is **\<f12\> \<f12\> \<f2\>** PEL also supports a set of templates that are more generic and are distinguished only by the comment style used by the language. These commands are available through the **\<f6\>** key prefix and are described in the ∑ **Inserting Text** table.
Example code template	Example of generated code are stored in example files stored in the repo under the **example/template directory**. The example files were generated with me as the author and for all combinations of the user-options. They also contain text that would normally not be generated. This text describes the values of each user-option that has an impact on the generated text. The file names end with a set of dash separated number with correspond to the value of the user-options.

Generic Templates	Generic Templates, available for all major modes
Key Sequences for Generic Templates See Also: ∑ **Inserting Text**	• To insert file header block: **\<f6\> h** • To access the **pel-pkg-generic-code-style-group**: **\<f6\> \<f2\>**
Template Source	By default, PEL controls the content of the generic templates. If you prefer a different content fo these generic templates, then you can create your own tempo skeleton files and identify them in the following user-option: • **pel-generic-skel-module-header-block-style**
0 - Separator Lines	**pel-generic-skel-use-separators**
1 - Timestamp	**pel-generic-skel-insert-file-timestamp**
2 - License & Copyright	**pel-generic-skel-with-license**
3 - Package Name	
4 - file variable	
5 - doc section titles	**pel-generic-skel-module-section-titles**

Programming Language	𝔓𝔩 - C 🚧	𝔓𝔩 - C++ 🚧		
Key Sequence to use in a language file to access the customization group See Also: ∑ **Customize**	**\<f12\> \<f12\> \<f2\>** • **Pel C Skeleton Control** • **Pel C Module Header Skeleton Control** • **Pel C Function Header Skeleton Control**	**\<f12\> \<f12\> \<f2\>** • **Pel C++ Skeleton Control** • **Pel C++ Module Header Skeleton Control** • **Pel C++ Function Header Skeleton Control**		
0 - Separator Lines	pel-c-skel-use-separators	pel-c++-skel-use-separators		
0.1 - Secondary separators				
1 - Timestamp	pel-c-skel-insert-file-timestamp	pel-c++-skel-insert-file-timestamp		
2 - License & Copyright	pel-c-skel-with-license	pel-c++-skel-with-license		
3 - Package Name				
4 - file variable				
5 - doc section titles - code	pel-c-skel-cfile-section-titles	pel-c++-skel-cppfile-section-titles		
5.1 - doc section titles - header	pel-c-skel-hfile-section-titles	pel-c++-skel-hppfile-section-titles		
6 - doc markup	pel-c-skel-doc-markup 🚧	pel-c++-skel-doc-markup 🚧		
7 - comment style	pel-c-skel-comment-with-2stars			
8 - UUID include guard	pel-c-skel-use-include-guards	pel-c++-skel-use-include-guards		

Programming Language	ℰ𝔓𝔩 - Emacs Lisp	𝔓𝔩 - Common Lisp 🚧		
Key Sequence to use in a language file to access the customization group See Also: ∑ **Customize**	**\<f12\> \<f12\> \<f2\>** • **Pel Elisp Code Style**	**\<f12\> \<f12\> \<f2\>** • **Pel Clisp Code Style**		
0 - Separator Lines	pel-elisp-skel-use-separator	pel-clisp-skel-use-separators		
0.1 - Secondary separators				
1 - Timestamp	pel-elisp-skel-insert-file-timestamp	pel-clisp-skel-insert-file-timestamp		
2 - License & Copyright	pel-elisp-skel-with-license	pel-clisp-skel-with-license		
3 - Package Name	pel-elisp-skel-package-name	pel-clisp-skel-package-name 🚧		
4 - file variable		pel-clisp-emacs-filevar-line		
5 - doc section titles				
6 - doc markup				
7 - comment style				

Programming Language	𝔅𝔩 - Erlang			
Key Sequence to use in a language file to access the customization group See Also: ∑ **Customize**	**\<f12> \<f12> \<f2>** • **Pel Erlang Code Style** • **Pel Erlang Skeleton Control**			
0 - Separator Lines	pel-erlang-skel-use-separators			
0.1 - Secondary separators	pel-erlang-skel-use-secondary-separators			
1 - Timestamp	pel-erlang-skel-insert-file-timestamp			
2 - License & Copyright	pel-erlang-skel-with-license			
3 - Package Name				
4 - file variable				
5 - doc section titles				
6 - doc markup	pel-erlang-skel-with-edoc			
7 - comment style		2		

Markup Language	𝔐 **reStructuredText**			
Key Sequence to use in a language file to access the customization group See Also: ∑ **Customize**	**\<f12> \<f2>** • **Pel reST style**			
0 - Separator Lines				
0.1 - Secondary separators				
1 - Timestamp	pel-rst-skel-insert-file-timestamp			
2 - License & Copyright	pel-rst-skel-with-license			
3 - Package Name				
4 - file variable				
5 - doc section titles				
6 - doc markup				
7 - comment style				

Templates— References

Topic & Link	Notes
Emacs Manual - Updating Time Stamps Automatically	Describe how to make emacs automatically update a time stamp when a file is saved.
Yasnippet GitHub repo/home page	
Yasnippet documentation page	
Yasnippet Menu	
Yasnippet Reference	
Textmate Snippets	
Emacs Wiki - Category Templates	
Emacs Wiki - Yasnippet	Also list packages that create snippets.
Unix & Linux - Code Template with Emacs	
Stack Overflow - What is the best code template facility for Emacs?	
Yasnippets - video - good presentation	
Yasnippets - screencast	
Yasnippets - video	
Emacs Rocks! Episode 06: Yeah! Snippets!	
EMACSulation	