

## Code/Markup Template Support

<b>Code Template Support</b>	PEL provides Emacs Tempo Skeletons specialized for some programming and markup languages, listed in this table. <ul style="list-style-type: none"> <li>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.</li> <li>The key bindings to insert text from the templates use the <code>&lt;f12&gt; &lt;f12&gt;</code> key prefix for the supported major modes.</li> </ul>
See also: <a href="#">                         ⌘ Inserting Text                     </a>	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 <code>&lt;f6&gt;</code> key prefix and are described in the <a href="#">                         ⌘ Inserting Text                     </a> table.
<b>Example code template</b>	Example of generated code are stored in example files stored in the repo under the <a href="#">example/template directory</a> . 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.

Programming Language	⌘ - C 🚧	⌘ - C++ 🚧		
<b>Key Sequence to use in a language file to access the customization group</b>	<b>&lt;f12&gt; &lt;f2&gt;</b> <ul style="list-style-type: none"> <li><b>Pel C Code Style</b></li> <li><b>Pel C Skeleton Control</b></li> </ul> 🚧 pel-c-skel-doc-markup is not used in the current implementation.			
See Also: <a href="#">                         ⌘ Customize                     </a>				
<b>0 - Separator Lines</b>	pel-c-skel-use-separators			
<b>0.1 - Secondary separators</b>				
<b>1 - Timestamp</b>	pel-c-skel-insert-file-timestamp			
<b>2 - License &amp; Copyright</b>	pel-c-skel-with-license			
<b>3 - Package Name</b>				
<b>4 - file variable</b>				
<b>5 - doc section titles</b>	pel-c-skel-module-section-titles			
<b>6 - doc markup</b>	pel-c-skel-doc-markup 🚧			
<b>7 - comment style</b>	pel-c-skel-comment-with-2stars			

Programming Language	⌘⌘ - Emacs Lisp	⌘ - Common Lisp 🚧		
<b>Key Sequence to use in a language file to access the customization group</b>	<b>&lt;f12&gt; &lt;f2&gt;</b> <ul style="list-style-type: none"> <li><b>Pel Elisp Code Style</b></li> </ul>	<b>&lt;f12&gt; &lt;f2&gt;</b> <ul style="list-style-type: none"> <li><b>Pel Clisp Code Style</b></li> </ul> 🚧 pel-clisp-skel-package-name is not used in the current implementation.		
See Also: <a href="#">                         ⌘ Customize                     </a>				
<b>0 - Separator Lines</b>	pel-elisp-skel-use-separator	pel-clisp-skel-use-separators		
<b>0.1 - Secondary separators</b>				
<b>1 - Timestamp</b>	pel-elisp-skel-insert-file-timestamp	pel-clisp-skel-insert-file-timestamp		
<b>2 - License &amp; Copyright</b>	pel-elisp-skel-with-license	pel-clisp-skel-with-license		
<b>3 - Package Name</b>	pel-elisp-skel-package-name	pel-clisp-skel-package-name 🚧		
<b>4 - file variable</b>		pel-clisp-emacs-filevar-line		
<b>5 - doc section titles</b>				
<b>6 - doc markup</b>				
<b>7 - comment style</b>				

Programming Language	⌘ - Erlang			
<b>Key Sequence to use in a language file to access the customization group</b>	<b>&lt;f12&gt; &lt;f2&gt;</b> <ul style="list-style-type: none"> <li><b>Pel Erlang Code Style</b></li> <li><b>Pel Erlang Skeleton Control</b></li> </ul>			
See Also: <a href="#">                         ⌘ Customize                     </a>				
<b>0 - Separator Lines</b>	pel-erlang-skel-use-separators			
<b>0.1 - Secondary separators</b>	pel-erlang-skel-use-secondary-separators			
<b>1 - Timestamp</b>	pel-erlang-skel-insert-file-timestamp			
<b>2 - License &amp; Copyright</b>	pel-erlang-skel-with-license			
<b>3 - Package Name</b>				
<b>4 - file variable</b>				
<b>5 - doc section titles</b>				
<b>6 - doc markup</b>	pel-erlang-skel-with-edoc			
<b>7 - comment style</b>				

Markup Language	⌘ reStructuredText 🚧			
<b>Key Sequence to use in a language file to access the customization group</b>				
See Also: <a href="#">                         ⌘ Customize                     </a>				

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	