

Code/Markup Template Support

<div> Code Templates </div> <div> Emacs: <ul style="list-style-type: none"> • Using Skeletons • Skeleton Language • Tempo </div> <div> See also: 🔗 Inserting Text </div>	<p>PEL provides Emacs Tempo Skeletons specialized for some programming and markup languages, listed in this table.</p> <ul style="list-style-type: none"> • 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. <p>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.</p>
<div> Example code template </div>	<p>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.</p>



Generic Templates	Generic Templates, available for all major modes
<div> Key Sequences for Generic Templates </div> <div> <ul style="list-style-type: none"> • To insert file header block: <f6> h • To access the pel-pkg-generic-code-style-group: <f6> <f2> </div> <div> See Also: 🔗 Inserting Text </div>	
<div> Template Source </div>	<p>By default, PEL controls the content of the generic templates. If you prefer a different content for these generic templates, then you can create your own tempo skeleton files and identify them in the following user-option:</p> <ul style="list-style-type: none"> • pel-generic-skel-module-header-block-style
<div> 0 - Separator Lines </div>	pel-generic-skel-use-separators
<div> 1 - Timestamp </div>	pel-generic-skel-insert-file-timestamp
<div> 2 - License & Copyright </div>	<ul style="list-style-type: none"> • pel-generic-skel-with-copyright • pel-generic-skel-with-license
<div> 3 - Package Name </div>	
<div> 4 - file variable </div>	
<div> 5 - doc section titles </div>	<ul style="list-style-type: none"> • pel-generic-skel-insert-module-sections • pel-generic-skel-module-section-titles

Programming Language	📄 - C 🛠️	📄 - C++ 🛠️		
<div> Key Sequence to use in a language file to access the customization group </div> <div> <ul style="list-style-type: none"> • <f12> <f2> • Pel C Code Style • Pel C Skeleton Control </div> <div> See Also: 🔗 Customize </div>	<p>🛠️ <f12> <f2></p> <ul style="list-style-type: none"> • Pel C Code Style • Pel C Skeleton Control <p>🛠️ pel-c-skel-doc-markup is not used in the current implementation.</p>	<p>🛠️ Nothing is implemented for C++ yet. What is shown here are the planned names of user-options.</p>		
<div> 0 - Separator Lines </div>	pel-c-skel-use-separators	pel-c++-skel-use-separators		
<div> 0.1 - Secondary separators </div>				
<div> 1 - Timestamp </div>	pel-c-skel-insert-file-timestamp	pel-c++-skel-insert-file-timestamp		
<div> 2 - License & Copyright </div>	pel-c-skel-with-license	pel-c++-skel-with-license		
<div> 3 - Package Name </div>				
<div> 4 - file variable </div>				
<div> 5 - doc section titles </div>	pel-c-skel-module-section-titles	pel-c++-skel-module-section-titles		
<div> 6 - doc markup </div>	pel-c-skel-doc-markup 🛠️	pel-++c-skel-doc-markup 🛠️		
<div> 7 - comment style </div>	pel-c-skel-comment-with-2stars			

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

Programming Language	📄 - Erlang			
<div> Key Sequence to use in a language file to access the customization group </div> <div> <ul style="list-style-type: none"> • <f12> <f2> • Pel Erlang Code Style • Pel Erlang Skeleton Control </div> <div> See Also: 🔗 Customize </div>				
<div> 0 - Separator Lines </div>	pel-erlang-skel-use-separators			
<div> 0.1 - Secondary separators </div>	pel-erlang-skel-use-secondary-separators			
<div> 1 - Timestamp </div>	pel-erlang-skel-insert-file-timestamp			
<div> 2 - License & Copyright </div>	pel-erlang-skel-with-license			
<div> 3 - Package Name </div>				

4 - file variable				
5 - doc section titles				
6 - doc markup	pel-erlang-skel-with-edoc			
7 - comment style				

Markup Language	M reStructuredText 			
Key Sequence to use in a language file to access the customization group				
See Also:  Customize				

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	