

Naming Conventions

File and Directory Names

Usual variable (suffix) names	Description	Examples
<code>file-path-name</code> <code>file-path</code>	Complete name. Includes both path and file name. Path may be absolute or relative.	1. <code>~/dev/elisp/pel/pel-erlang.el</code> 2. <code>/Users/roup/dev/erlang/system_x/runner.erl</code> 3. <code>../code/parser.cpp</code> 4. <code>~/dev/elisp/pel/doc/pel-user-manual.1.rst</code>
<code>dir-path</code>	Complete absolute or relative path. Emacs considers path as a string that ends with a path separator <code>/</code>	1. <code>~/dev/elisp/pel/</code> 2. <code>/Users/roup/dev/erlang/system_x/</code> 3. <code>../code/</code> 4. <code>~/dev/elisp/pel/doc/</code>
<code>dir-path-name</code>	Complete absolute or relative path. What Emacs considers a <i>file name</i> for a directory: the path without the trailing path separator. Note that a directory is <i>also</i> a file.	1. <code>~/dev/elisp/pel</code> 2. <code>/Users/roup/dev/erlang/system_x</code> 3. <code>../code</code> 4. <code>~/dev/elisp/pel/doc</code>
<code>file-name</code>	The file name alone, without any path component.	1. <code>pel-erlang.el</code> 2. <code>runner.erl</code> 3. <code>parser.cpp</code> 4. <code>pel-user-manual.1.rst</code>
<code>dir-name</code>	The name of the parent directory.	1. <code>pel</code> 2. <code>system_x</code> 3. <code>code</code> 4. <code>doc</code>
<code>basename</code> <code>file-base</code> <code>file-basename</code>	The file name without its path and extension components.	1. <code>pel-erlang</code> 2. <code>runner</code> 3. <code>parser</code> 4. <code>pel-user-manual</code>
<code>extension</code> <code>file-extension</code>	The string after the last period in the file name.	1. <code>~/dev/elisp/pel/pel-erlang-el</code> 2. <code>erl</code> 3. <code>cpp</code> 4. <code>rst</code>

Source Code Naming Conventions - - Function Names

<code>pel-</code>	‘public’ function or command	A PEL function or interactive command that can be executed from anywhere.
<code>pel--</code>	‘private’ function	A PEL function that should not be used externally; these are meant to be used only by appropriate PEL code.
<code>pel-Σ-</code>	public’ hydra command	These are ‘public’ commands that can be used anywhere. The Σ symbol is used instead of the word ‘ hydra ’ in command names to shorten them a little. This helps showing a larger portion o the command name when the prefix key is typed and which-mode shows the list of commands in the minibuffer. 👉 Under macOS, typing Σ is done using the <code>⌘w</code> key combination. - Note that when Emacs runs in terminal mode you can use <code>⌘⌥⌘</code> to switch the meaning of the <code>⌘</code> key between Option and Meta.