Naming Conventions 🚧

File and Directory Names		
Usual variable (suffix) names	**Description**	**Examples**
• **file-path-name** • **file-path**	Complete name. Includes both path and file name. Path may be absolute or relative.	1. `~/dev/elisp/pel/pel-erlang.el` 2. `/Users/roup/dev/erlang/system_x/runner.erl` 3. `../code/parser.cpp` 4. `~/dev/elisp/pel/doc/pel-user-manual.1.rst`
dir-path	Complete absolute or relative path. Emacs considers path as a string that ends with a path separator /	1. `~/dev/elisp/pel/` 2. `/Users/roup/dev/erlang/system_x/` 3. `../code/` 4. `~/dev/elisp/pel/doc/`
dir-path-name	Complete absolute or relative path. What Emacs considers a *file name* for a directory: the path without the trailing path separator. Note that a directory is *also* a file.	1. `~/dev/elisp/pel` 2. `/Users/roup/dev/erlang/system_x` 3. `../code` 4. `~/dev/elisp/pel/doc`
file-name	The file name alone, without any path component.	1. `pel-erlang.el` 2. `runner.erl` 3. `parser.cpp` 4. `pel-user-manual.1.rst`
dir-name	The name of the parent directory.	1. `pel` 2. `system_x` 3. `code` 4. `doc`
• **basename** • **file-base** • **file-basename**	The file name without its path and extension components.	1. `pel-erlang` 2. `runner` 3. `parser` 4. `pel-user-manual`
• **extension** • **file-extension**	The string after the **last** period in the file name.	1. `~/dev/elisp/pel/pel-erlang-el` 2. `erl` 3. `cpp` 4. `rst`
Source Code Naming Conventions - - Function Names		
pel-	'public' function or command	A PEL function or interactive command that can be executed from anywhere.
pel--	'private' function	A PEL function that should not be used externally; these are meant to be used only by appropriate PEL code.
pel-∑-	public' **hydra** command	These are 'public' commands that can be used anywhere. The ∑ 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 ∑ is done using the ⌥**w** key combination. • Note that when Emacs runs in terminal mode you can use ⌘⌥**o** to switch the meaning of the ⌥ key between Option and Meta.