

# Naming Conventions

## File and Directory Names

Usual variable (suffix) names	Description	Examples
<ul style="list-style-type: none"> <li>file-path-name</li> <li>file-path</li> </ul>	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 <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. ~/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
<ul style="list-style-type: none"> <li>basename</li> <li>file-base</li> <li>file-basename</li> </ul>	The file name without its path and extension components.	1. pel-erlang 2. runner 3. parser 4. pel-user-manual
<ul style="list-style-type: none"> <li>extension</li> <li>file-extension</li> </ul>	The string after the <b>last</b> period in the file name.	1. ~/dev/elisp/pel/pel-erlang-el 2. erl 3. cpp 4. rst