preprocess-lambda-keywords *args* [*FUNCTION*]

Unused

preprocess-lisp-latex-clashes *args purge-doc* [*FUNCTION*]

This function is used to make the strings for the arguments of the form digestible for LaTeX, e.g. by removing '#' and '&'.

preprocess-character *c* [*FUNCTION*]

Low level processing of single characters, when passed as defaults to optional, key and aux parameters.

preprocess-specials *list-form purge-doc* [*FUNCTION*]

Processing of some 'special' forms. Only 'quote' and 'function' are treated for the time being.

split-string *string width* &optional *arglistp filled (trim-whitespace t)* [*FUNCTION*]

Splits a string into a list of strings, each of which is shorter than the specified width. Tries to be intelligent about where to split the string if it is an argument list. If filled is T, tries to fill out the strings as much as possible. This function is used to break up long argument lists nicely, and to break up wide lines of documentation nicely.

split-point *string max-length* &optional *arglistp filled* [*FUNCTION*]

Finds an appropriate point to break the string at given a target length. If arglistp is T, tries to find an intelligent position to break the string. If filled is T, tries to fill out the string as much as possible.

lambda-list-keyword-position *string* &optional *end trailer-only* [*FUNCTION*]

If the previous symbol is a lambda-list keyword, returns its position. Otherwise returns end.

balanced-parenthesis-position *string* &optional *end* [*FUNCTION*]

Finds the position of the left parenthesis which is closest to END but leaves the prefix of the string with balanced parentheses or at most 1 unbalanced left parenthesis.

parse-with-delimiter *line* &optional *(delim #\newline)* [*FUNCTION*]

Breaks LINE into a list of strings, using DELIM as a breaking point.

defclass *form* [*DOC-HANDLER*]

Documentation handler for classs.

failed-definition-types *nil* [*VARIABLE*]

List of definition types that create-user-manual couldn't handle.

create-user-manual *filename* &key *(output-format 'text) (output-stream *standard-output*) (purge-latex t)* [*FUNCTION*]

Automatically creates a user manual for the functions in a file by collecting the documentation strings and argument lists of the functions and formatting the output nicely. Returns a list of the definition types of the forms it couldn't handle. Output-format may be either 'TEXT, 'SCRIBE or 'LATEX. In this last case the extra keyword 'purge-latex' may be specified: if non nil the latex filter will try to substitute possible dangerous characters like '&', '\' and '#'.

handle-form-output *form* &optional *(output-format 'text) (stream *standard-output*) (purge-latex t)* [*FUNCTION*]

This function takes a form as input and outputs its documentation segment to the output stream.

output-text-documentation *name type args documentation args-tab-pos type-pos* &optional *(stream *standard-output*)* [*FUNCTION*]

Prints out the user guide entry for a form in TEXT mode.

output-scribe-documentation *name type args documentation* &optional *(stream *standard-output*)* [*FUNCTION*]

Prints out the user guide entry for a form in SCRIBE mode.

output-latex-documentation *name type args documentation* &optional *(stream *standard-output*) (purge-documentation t)* [*FUNCTION*]

Prints out the user guide entry for a form in LaTeX mode.

purge-string-for-latex *a-string purge-doc* [*FUNCTION*]

Tries to purge a string from characters that are potentially dangerous for LaTeX.

define-doc-handler *form* [*DOC-HANDLER*]

Documentation handler for doc-handlers.

defvar *form* [*DOC-HANDLER*]

Documentation handler for variables.

defconstant *form* [*DOC-HANDLER*]

Documentation handler for constants.

defparameter *form* [*DOC-HANDLER*]

Documentation handler for parameters.

defun *form* [*DOC-HANDLER*]

Documentation handler for functions.

defmacro *form* [*DOC-HANDLER*]

Documentation handler for macros.

defstruct *form* [*DOC-HANDLER*]

Documentation handler for structures.

define-condition *form* [*DOC-HANDLER*]

Documentation handler for conditions.

deftype *form* [*DOC-HANDLER*]

Documentation handler for types.

defsetf *form* [*DOC-HANDLER*]

Documentation handler for setf mappings.

defmethod *form* [*DOC-HANDLER*]

Documentation handler for methods.

defgeneric *form* [*DOC-HANDLER*]

Documentation handler for generic functions.

LaTeX output for `user-manual`

Marco Antoniotti
Robotics Lab
Courant Institute for Mathematical Science
New York University
marcoxa@robocop.nyu.edu

January 5, 1993

extract-documentation *body* [*MACRO*]

atom-or-car *list-or-atom* [*FUNCTION*]

documentation-handlers *(make-hash-table :test #'equal)* [*VARIABLE*]

Hash table of entries of the form (handler description), where definer is the car of the definition form handled (for example, DEFUN or DEFMACRO), handler is a function which takes the form as input and value-returns the name, argument-list and documentation string, and description is a one-word equivalent of definer (for example, FUNCTION or MACRO).

define-doc-handler *definer arglist description* &body *body* [*MACRO*]

Defines a new documentation handler. DEFINER is the car of the definition form handled (e.g., defun), DESCRIPTION is a one-word string equivalent of definer (e.g., "function"), and ARGLIST and BODY together define a function that takes the form as input and value-returns the name, argument-list, documentation string, and a list of any qualifiers of the form.

find-doc-handler *definer* [*FUNCTION*]

Given the car of a form, finds the appropriate documentation handler for the form if one exists.