

## I. Specify packages, classes, macros or environments

Here's what you can type semantically.

<code>\tdoccls{myclass}</code> is for...	myclass is for...
<code>\tdocpack{mypackage}</code> is for...	mypackage is for...
<code>\tdocmacro{onemacro}</code> is for...	\onemacro is for...
<code>\tdocenv{env}</code> produces...	\begin{env} ... \end{env} produces...
Just <code>\tdocenv*{env}</code> or	Just <code>\begin{*} ... \end{*}env</code> or
<code>\tdocenv[[opt1]&lt;opt2&gt;]{env}</code>	<code>\begin{env}[opt1]&lt;opt2&gt; ... \end{env}</code>

**Remark.** The advantage of the previous macros over the use of `\tdocinlatex`, see the section ?? page ??, is the absence of colouring. Furthermore, the `\tdocenv` macro simply asks you to type the name of the environment <sup>1</sup> with any options by typing the correct delimiters <sup>2</sup> by hand.

### Warning.

The optional argument to the `\tdocenv` macro is copied and pasted during rendering. This can sometimes require the use of protective braces, as in the previous example.

## II. Origin of a prefix or suffix

To explain the names chosen, there is nothing like indicating and explaining the short prefixes and suffixes used. This is easily done as follows.

<code>\tdocpre{sup}</code> relates to...	sup relates to...
<code>\tdocprewhy{sup.erbe}</code> means...	sup•erbe means...
<code>\emph{\tdocprewhy{sup.er} for...}</code>	sup•er for...

**Remark.** The choice of a full stop to split a word allows words with a hyphen to be used, as in `\tdocprewhy{bric.k-breaker}` which gives *bric•k-breaker*.

<sup>1</sup>In addition, `\tdocenv{monenv}` produces `\begin{monenv} ... \end{monenv}` with spaces to allow line breaks if necessary.

<sup>2</sup>Remember that almost anything is possible from now on.