

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

Here's what you can type semantically.

<code>\tdoccls{myclass}</code> is for... <code>\</code>	myclass is for...
<code>\tdocpack{mypackage}</code> is for... <code>\</code>	mypackage is for...
<code>\tdocmacro{onemacro}</code> is for... <code>\</code>	\onemacro is for...
<code>\tdocenv{env}</code> produces... <code>\</code>	\begin{env} ... \end{env} produces...
Just <code>\tdocenv*{env}</code> ...	Just env...

**Remark I.1.** Unlike `\tdoclatexin`, the `\tdocmacro`, `\tdocenv` and `\tdocenv*` macros don't color the text they produce. In addition, `\tdocenv{monenv}` produces `\begin{monenv} ... \end{monenv}` with breakable spaces to allow line breaks if required.

## 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... <code>\</code>	sup relates to...
<code>\tdocprewhy{sup.erbe}</code> means... <code>\</code>	sup·erbe means...
<code>\emph{\tdocprewhy{sup.er} for...}</code>	<i>sup·er for...</i>

**Remark II.1.** 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*.