I. Specify packages, classes, macros or environments

Here's what you can type semantically.

```
\tdoccls{myclass} is for...

\tdocpack{mypackage} is for...

\tdocmacro{onemacro} is for...

\tdocenv{env} produces...

We also have :

\tdocenv[{[opt1]<opt2>}]{env}
```

myclass is for...
mypackage is for...
\onemacro is for...
\begin{env} ... \end{env} produces...
We also have :
\begin{env}[opt1]<opt2> ... \end{env}

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* [1] *with any options by typing the correct delimiters* [2] *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.

```
\tdocpre{sup} relates to...

\tdocprewhy{sup.erbe} means...

\emph{\tdocprewhy{sup.er} for...}
```

sup relates to...
sup·**erbe** means...
*sup·**er** for...*

Remark. *Le choix du point pour scinder un mot permet d'utiliser des mots avec un tiret comme dans* `\tdocprewhy{ca.sse-brique}` *qui donne* ***ca·sse-brique****. 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****.*

[1] In addition, `\tdocenv{monenv}` produces `\begin{monenv} ... \end{monenv}` with spaces to allow line breaks if necessary.

[2] Remember that almost anything is possible from now on.