

## I. Indiquer des packages, des classes, des macros ou des environnements

Voici ce qu'il est possible de taper de façon sémantique.

<code>\tdoccls{maclasse} sert à...</code>	<code>\\</code>	maclasse sert à...
<code>\tdocpack{monpackage} est pour...</code>	<code>\\</code>	monpackage est pour...
<code>\tdocmacro{unemacro} permet de...</code>	<code>\\</code>	\unemacro permet de...
<code>\tdocenv{env} produit...</code>	<code>\\</code>	\begin{env} ... \end{env} produit...
<code>\tdocenv[[{opt1}&lt;opt2&gt;]]{env}</code>	<code>\\</code>	\begin{env}[opt1]<opt2> ... \end{env}
Juste <code>\tdocenv*{env}...</code>	<code>\\</code>	Juste env...
Enfin <code>\tdocenv*[[{opt1}&lt;opt2&gt;]]{env}...</code>		Enfin env...
<code>% Pour les copier-coller.</code>		

**Remarque I.1.** Contrairement à `\tdoclatexin`, les macros `\tdocenv` et `\tdocenv*` ne colorent pas le texte produit. De plus, `\tdocenv{monenv}` produit `\begin{monenv} ... \end{monenv}` avec des espaces afin d'autoriser des retours à la ligne si besoin.



### Avertissement.

L'argument optionnel de la macro `\tdocenv` est copié-collé<sup>a</sup> lors du rendu. Ceci peut donc parfois nécessiter d'utiliser des accolades protectrices comme dans l'exemple ci-dessus.

a. Se souvenir que tout est possible ou presque dorénavant.

## II. Origine d'un préfixe ou d'un suffixe

Pour expliquer les noms retenus, rien de tel que d'indiquer et expliciter les courts préfixes et suffixes employés. Ceci se fait facilement comme suit.

<code>\tdocpre{sup} est relatif à...</code>	<code>\\</code>	sup est relatif à...
<code>\tdocprewhy{sup.erbe} signifie...</code>	<code>\\</code>	sup·erbe signifie...
<code>\emph{\tdocprewhy{sup.er} pour...}</code>		sup·er pour...

**Remarque II.1.** 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*.