

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}</code> sert à...	
<code>\tdocpack{monpackage}</code> est pour...	<code>maclasse</code> sert à...
<code>\tdocmacro{unemacro}</code> permet de...	<code>monpackage</code> est pour...
<code>\tdocenv{env}</code> produit...	<code>\unemacro</code> permet de...
On a aussi :	<code>\begin{env} ... \end{env}</code> produit...
	On a aussi :
<code>\tdocenv[[opt1]<opt2>]{env}</code>	<code>\begin{env}[opt1]<opt2> ... \end{env}</code>

Remarque. L'intérêt des macros précédentes vis à vis de l'usage de `\tdocinlatex`, voir la section ?? page ??, est l'absence de coloration. De plus, la macro `\tdocenv` demande juste de taper le nom de l'environnement¹ avec des éventuelles options en tapant les bons délimiteurs² à la main.

Avertissement. L'argument optionnel de la macro `\tdocenv` est copié-collé lors du rendu. Ceci peut donc parfois nécessiter d'utiliser des accolades protectrices comme dans l'exemple précédent.

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

Remarque. 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*.

1. De plus, `\tdocenv{monenv}` produit `\begin{monenv} ... \end{monenv}` avec des espaces afin d'autoriser des retours à la ligne si besoin.

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