

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

Remarque I.1. Contrairement à `\tdocinlatex`, 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^a 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}</code> est relatif à...	<code>\\</code>	sup est relatif à...
<code>\tdocprewhy{sup.erbe}</code> signifie...	<code>\\</code>	sup·erbe signifie...
<code>\emph{\tdocprewhy{sup.er} pour...}</code>		<i>sup·er pour...</i>

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*.