

I. Cas d'utilisation en L^AT_EX

Documenter un package ou une classe se fait efficacement via des cas d'utilisation montrant à la fois du code et le résultat correspondant.

1. Codes « en ligne »

La macro `\tdocinlatex`¹ permet de taper du code en ligne via un usage similaire à `\verb`. Voici des exemples d'utilisation.

1: <code>\tdocinlatex \$a^b = c\$ </code>	1: $a^b = c$
2: <code>\tdocinlatex+\tdocinlatex \$a^b = c\$ +</code>	2: <code>\tdocinlatex \$a^b = c\$ </code>

Note. La macro `\tdocinlatex` est utilisable dans une note de pied de page : voir le bas de cette page². De plus, une couleur de fond est volontairement utilisée pour subtilement faire apparaître les codes `\LaTeX`.

2. Codes tapés directement

Exemple 1 (Face à face). Via `\begin{tdoclatex}[sbs] ... \end{tdoclatex}`, on affichera un code et son rendu côte à côte. Indiquons que `sbs` est pour « `s·ide b·y s·ide` » soit « côte à côte » en anglais. Considérons le code suivant.

```
\begin{tdoclatex}[sbs]
  $A = B + C$
\end{tdoclatex}
```

Ceci produira ce qui suit.

$A = B + C$	$A = B + C$
-------------	-------------

Exemple 2 (À la suite). `\begin{tdoclatex} ... \end{tdoclatex}` produit le résultat suivant qui correspond à l'option par défaut `std`³.

```
$A = B + C$
```

 $A = B + C$

Exemple 3 (Juste le code). Via `\begin{tdoclatex}[code] ... \end{tdoclatex}`, on aura juste le code comme ci-après.

```
$A = B + C$
```

Danger. Avec la mise en forme par défaut, si le code commence par un crochet ouvrant, il faudra indiquer explicitement l'option par défaut. Considérons le code suivant.

```
\begin{tdoclatex}[std]
  [Étrange... Ou pas !]
\end{tdoclatex}
```

Ceci produira ce qui suit.

```
[Étrange... Ou pas !]
```

1. Le nom de la macro `\tdocinlatex` vient de « *in-line L^AT_EX* » soit « *L^AT_EX en ligne* » en anglais.
2. `$minted = TOP$` a été tapé `\tdocinlatex+$minted = TOP$+` dans cette note de bas de page..
3. `std` fait référence au comportement « *standard* » de `tcolorbox` vis à vis de la librairie `minted`.

[Étrange... Ou pas !]

Une autre méthode consiste à utiliser la primitive `\string`. Considérons le code suivant.

```
\begin{tdoclatex}  
  \string[Étrange... Ou pas !]  
\end{tdoclatex}
```

Ceci produira ce qui suit.

[Étrange... Ou pas !]

[Étrange... Ou pas !]