

# I. Cas d'utilisation en $\LaTeX$

Documenter un package ou une classe se fait efficacement via des cas d'utilisation montrant à la fois du code et le résultat correspondant.<sup>1</sup>

## 1. Codes « en ligne »

La macro `\tdoclatexin`<sup>2</sup> permet de taper du code en ligne via un usage similaire à `\verb` ou bien comme une macro standard (voir la gestion des accolades dans le dernier cas ci-dessous). Voici des exemples d'utilisation.

1: <code>\tdoclatexin \$a^b = c\$ </code> <code>\\</code>	1: $a^b = c$
2: <code>\tdoclatexin+\tdoclatexin \$a^b = c\$ +</code> <code>\\</code>	2: <code>\tdoclatexin \$a^b = c\$ </code>
3: <code>\tdoclatexin{\tdoclatexin \$a^b = c\$ }</code>	3: <code>\tdoclatexin \$a^b = c\$ </code>

### Note.

La macro `\tdoclatexin` est utilisable dans une note de pied de page : voir ci-dessous.<sup>a</sup> De plus, une couleur de fond est volontairement utilisée pour subtilement faire ressortir les codes `\LaTeX`.

a. `$minted = TOP$` a été tapé `\tdoclatexin+$minted = TOP$+` dans cette note de bas de page..

## 2. Codes tapés directement

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

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

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

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

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

### ⚠ Avertissement.

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 !]

-----  
[Étrange... Ou pas !]

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

1. La mise en forme des codes se fait via les packages `minted` et `tcolorbox`.  
2. Le nom de la macro `\tdoclatexin` vient de « *in·line*  $\LaTeX$  » soit «  $\LaTeX$  en ligne » en anglais.  
3. `std` fait référence au comportement « *standard* » de `tcolorbox` vis à vis de la librairie `minted`.

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

*Ceci produira ce qui suit.*

[Étrange... Ou pas !]

[Étrange... Ou pas !]