

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

1. Codes « en ligne »

Exemple I.1 (Usage standard). La macro `\tdoclatexin`² 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.³

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

Exemple I.2 (Des options possibles). Comme la macro `\tdoclatexin` s'appuie sur `minted`, on peut utiliser toutes les options prises en compte par `minted`. Voici des exemples d'utilisation.

1: <code>\tdoclatexin{\$a~b = c\$}</code>	1: <code>\$a~b = c\$</code>
2: <code>\tdoclatexin[style = bw]{\$a~b = c\$}</code>	2: <code>\$a~b = c\$</code>
3: <code>\tdoclatexin[style = igor, showspaces]{\$a~b = c\$}</code>	3: <code>\$a~b_=_c\$</code>

Note.

La macro `\tdoclatexin` est utilisable dans une note de pied de page : voir ci-dessous.^a

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

2. Codes tapés directement

Exemple I.3 (Face à face). Afficher un code et son rendu côte à côte se fait comme suit où la macro `\tdoctcb` permet de juste taper `\tdoctcb{sbs}` au lieu de `listing side text` (`sbs` est pour « **s**-**i**de **b**.**y** **s**-**i**de » soit « côte à côte » en anglais, tandis que `tcb` est l'abréviation standard de `tcolorbox`). Bien noter l'emploi de chevrons, et non de crochets (nous revenons sur ceci un plus bas).

<code>\begin{tdoclatex}<\tdoctcb{sbs}></code>
<code>\$A = B + C\$</code>
<code>\end{tdoclatex}</code>

Ceci donne :

<code>\$A = B + C\$</code>	<code>A = B + C</code>
----------------------------	------------------------

Exemple I.4 (À la suite). `\begin{tdoclatex}... \end{tdoclatex}` produit le résultat suivant (ce réglage par défaut s'obtient aussi via l'emploi de `\tdoctcb{std}`).⁴

<code>\$A = B + C\$</code>
<code>A = B + C</code>

Exemple I.5 (Juste le code). Via `\tdoctcb{code}`, on aura juste le code comme ci-après.

<code>\$A = B + C\$</code>

Exemple I.6 (Personnaliser). L'environnement `tdoclatex` accepte deux types d'argument optionnel.

1. Entre de classiques crochets, on peut employer toute option prise en compte par `minted`.
2. Entre des chevrons, on peut employer toute option prise en compte par les environnements obtenus via `tcolorbox`.

Par exemple, on peut faire les modifications suivantes si besoin.⁵

-
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** L^AT_EX » soit « L^AT_EX en ligne » en anglais.
 3. Une couleur de fond est volontairement utilisée pour subtilement faire ressortir les codes `\LaTeX`.
 4. `std` fait référence au comportement « *standard* » de `tcolorbox` vis à vis de la librairie `minted`.
 5. Cette documentation utilise les options entre chevrons pour obtenir des rendus corrects de codes L^AT_EX produisant des

```

\begin{tdoclatex}%
  [linenos, style = igor, showspace]%
  <\tdoctcb{sbs},
  attach boxed title to top left = {yshift = -9pt},
  fonttitle                       = \bfseries,
  title                           = Modifications locales,
  top                             = 10pt>
% Parfois utile, mais ne pas en abuser !
$A = B + C$
% Fin de cette démonstration.
\end{tdoclatex}

```

Ceci donne :

Modifications locales

```

1  %_Parfois_utile,_mais_ne_pas_en_abuser_!
2  $A=_B_+_C$
3  %_Fin_de_cette_démonstration.

```

$A = B + C$

⚠ Avertissement.

Pour obtenir la mise en forme par défaut d'un code commençant par un crochet, ou un chevron, il faudra bidouiller un peu comme ci-dessous.

```

\begin{tdoclatex}[]
[Étrange... Ou pas !]
\end{tdoclatex}
OU.
\begin{tdoclatex}<>
<Étrange... Ou pas !>
\end{tdoclatex}

```

Ceci donne :

```

[Étrange... Ou pas !]
-----
[Étrange... Ou pas !]

```

OU.

```

<Étrange... Ou pas !>
-----
<Étrange... Ou pas !>

```

Une autre méthode consiste à utiliser la primitive `\string` comme ci-après.

```

\begin{tdoclatex}
\string[Étrange... Ou pas !]
\end{tdoclatex}
OU.
\begin{tdoclatex}
\string<Étrange... Ou pas !>
\end{tdoclatex}

```

Ceci donne :

```

[Étrange... Ou pas !]
-----
[Étrange... Ou pas !]

```

OU.

<Étrange... Ou pas !>

<Étrange... Ou pas !>