Le package tutodoc - Documentation de type tutoriel

Christophe BAL

25 sept. 2024 - Version 1.3.0

Résumé

Le package tutodoc¹ est utilisé par son auteur pour produire de façon sémantique des documentations de packages et de classes IATEX dans un style de type tutoriel², et avec un rendu sobre pour une lecture sur écran.

Deux points importants à noter.

- Ce package impose un style de mise en forme. Dans un avenir plus ou moins proche, tutodoc sera sûrement éclaté en une classe et un package.
- Cette documentation est aussi disponible en anglais.

Abstract.

The tutodoc package³ is used by its author to semantically produce documentation of LATEX packages and classes in a tutorial style⁴, and with a sober rendering for reading on screen.

Two important points to note.

- This package imposes a formatting style. In the not-too-distant future, tutodoc will probably be split into a class and a package.
- This documentation is also available in French.

 $^{1. \ \ \}text{Le nom vient de} \ \textit{``tuto-rial-type doc-umentation"} \ \ \text{qui se traduit en } \ \textit{``documentation'} \ \ \textit{de type tutoriel'} \ .$

^{2.} L'idée est de produire un fichier PDF efficace à parcourir pour des besoins ponctuels. C'est généralement ce que l'on attend d'une documentation liée au codage.

^{3.} The name comes from « $tuto \cdot rial - type \ doc \cdot umentation$ ».

^{4.} The idea is to produce an efficient PDF file that can be browsed for one-off needs. This is generally what is expected of coding documentation.

Table des matières

I.	Mises en forme générales imposées	3
	1. Géométrie de la page	3
	2. Titre et table des matières	3
	3. Liens dynamiques	4
II.	Choisir la langue au chargement du package	4
III.	Cela veut dire quoi en « anglais »	4
IV.	Mettre en avant du contenu	4
	1. Du contenu dans le flot de la lecture	4
	i. Des exemples	4
	ii. Des remarques	5
	2. Du contenu tape-à-l'oeil	6
	i. Une astuce	6
	ii. Note informative	6
	iii. Un truc important	6
	iv. Avertir d'un point très délicat	6
	v. Avertir d'un danger	7
V.	Indiquer des packages, des classes, des macros ou des environnements	7
VI.	Origine d'un préfixe ou d'un suffixe	7
VII.	Un rendu en situation réelle	8
	1. Avec une bande colorée	8
	2. Sans bande colorée	9
	3. En important le code L ^A T _E X	10
VIII.	Cas d'utilisation en L ^A T _E X	10
	1. Codes « en ligne »	10
	2. Codes tapés directement	11
	3. Codes importés	12
	4. Codes importés et mis en situation	12
IX.	Indiquer les changements	14
	1. À quel moment?	14
	2. Quoi de neuf?	15
X.	Décorations	16
XI.	Historique	16

I. Mises en forme générales imposées

1. Géométrie de la page

Le package geometry est chargé avec les réglages suivants.

2. Titre et table des matières

Les packages titlesec et tocbasic sont réglés comme suit.

```
\RequirePackage[raggedright]{titlesec}
\ifcsundef{chapter}%
          {}%
          {\renewcommand\thechapter{\Alph{chapter}.}}
\renewcommand\thesection{\Roman{section}.}
\renewcommand\thesubsection{\arabic{subsection}.}
\renewcommand\thesubsubsection{\roman{subsubsection}.}
\titleformat{\paragraph}[hang] %
            {\normalfont\normalsize\bfseries}%
            {\theparagraph}{1em}%
            {}
\titlespacing*{\paragraph}%
              {0pt}%
              {3.25ex plus 1ex minus .2ex}%
              \{0.5em\}
% Source
* https://tex.stackexchange.com/a/558025/6880
\DeclareTOCStyleEntries[
 raggedentrytext,
 linefill = \hfill,
 indent = Opt,
 dynindent,
 numwidth = Opt,
 numsep = 1ex,
 dynnumwidth
]{tocline}{
 chapter,
 section,
 subsection,
 subsubsection,
 paragraph,
 subparagraph
\DeclareTOCStyleEntry[indentfollows = chapter]{tocline}{section}
```

3. Liens dynamiques

Le package hyperref est importé en coulisse avec les réglages ci-dessous.

```
\hypersetup{
  colorlinks,
  citecolor = orange!75!black,
  filecolor = orange!75!black,
  linkcolor = orange!75!black,
  urlcolor = orange!75!black
}
```

II. Choisir la langue au chargement du package

La présente documentation utilise le français via \usepackage[lang = french]{tutodoc} . Pour le moment, on a juste les deux choix suivants.

- 1. english est la valeur par défaut.
- 2. french est pour « français » en anglais.



Les noms des langues sont ceux proposés par le package babel.

III. Cela veut dire quoi en « anglais »

Penser aux non-anglophones est bien, même si ces derniers se font de plus en plus rares.

```
Cool et top signifient \tdocinEN*{cool} et \tdocinEN{top}.

Cool et top signifient « cool » et « top » en anglais.
```

La macro \tdocineN et sa version étoilée s'appuient sur \tdocquote : par exemple, « sémantique » s'obtient via \tdocquote{sémantique} .



Le texte « en anglais » est traduit dans la langue indiquée lors de l'importation de tutodoc.

IV. Mettre en avant du contenu



Les environnements présentés dans cette section ^a ajoutent un court titre indiquant le type d'informations fournies. Ce court texte sera toujours traduit dans la langue indiquée lors du chargement du package tutodoc.

a. La mise en forme provient du package keytheorems.

1. Du contenu dans le flot de la lecture

i. Des exemples

Des exemples numérotés, ou non, s'indiquent via l'environnement \begin{tdocexa} ...\end{tdocexa} qui propose deux arguments optionnels.

- 1. Le 1^{er} argument entre chevrons <...> peut prendre au choix les valeurs nb pour numéroter, réglage par défaut, et nonb pour ne pas numéroter.
- 2. Le 2^e argument entre crochets [...] sert à ajouter un mini-titre.

Voici différents emplois possibles.

```
Bla, bla, bla...
\begin{tdocexa}
    Ble, ble, ble...
\end{tdocexa}
                                                    Bla, bla, bla...
\begin{tdocexa} [Magnifique]
                                                    Exemple 1. Ble, ble, ble...
    Bli, bli, bli...
                                                    Exemple 2 (Magnifique). Bli, bli, bli...
\end{tdocexa}
                                                    Exemple. Blo, blo, blo...
\begin{tdocexa}<nonb>
   Blo, blo, blo...
                                                    Exemple (Superbe). Blu, blu, blu...
\end{tdocexa}
\begin{tdocexa}<nonb>[Superbe]
    Blu, blu, blu...
\end{tdocexa}
```

✓ Important.

La numérotation des exemples est remise à zéro dès qu'une section de niveau au moins égale à une \subsubsection est ouverte.

Astuce.

Il peut parfois être utile de revenir à la ligne dès le début du contenu. Le code suivant montre comment faire (ce tour de passe-passe reste valable pour l'environnement tdocrem présenté juste après). Noter au passage que la numérotation suit celle de l'exemple précédent comme souhaité.

ii. Des remarques

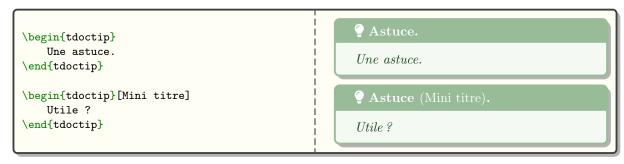
Tout se passe via l'environnement \begin{tdocrem} . . . \end{tdocrem} comme dans l'exemple suivant.

\begin{tdocrem} Juste une remarque \end{tdocrem}	Remarque. Juste une remarque
<pre>\begin{tdocrem}[Mini titre] Utile ? \end{tdocrem}</pre>	Remarque (Mini titre). Utile?

2. Du contenu tape-à-l'oeil

i. Une astuce

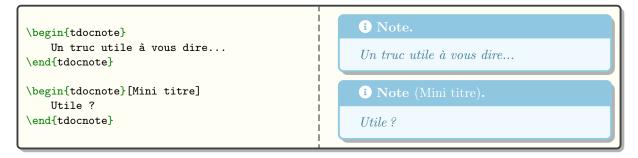
L'environnement \begin{tdoctip} ...\end{tdoctip} sert à donner des astuces. Voici comment l'employer.





ii. Note informative

L'environnement \begin{tdocnote} ...\end{tdocnote} sert à mettre en avant des informations utiles. Voici comment l'utiliser.



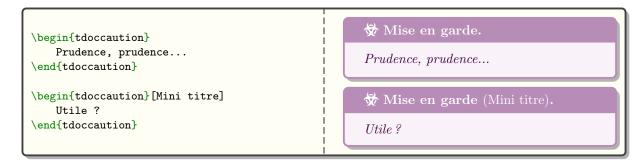
iii. Un truc important

L'environnement $\begin{tdocimp} \dots \end{tdocimp} \ permet d'indiquer quelque chose d'important mais sans danger.$



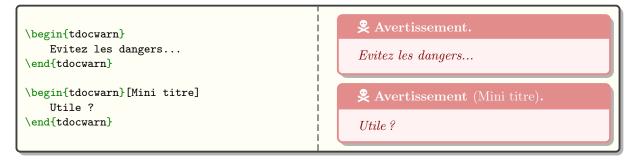
iv. Avertir d'un point très délicat

L'environnement \begin{tdoccaution} ...\end{tdoccaution} sert à indiquer un point délicat à l'utilisateur. Voici comment l'employer.



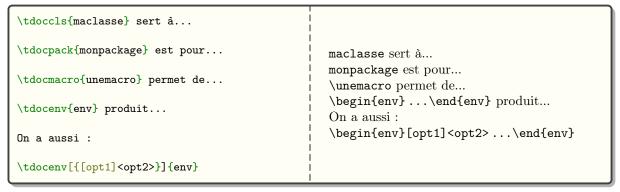
v. Avertir d'un danger

L'environnement \begin{tdocwarn} ...\end{tdocwarn} sert à avertir l'utilisateur d'un piège à éviter. Voici comment l'employer.



V. Indiquer des packages, des classes, des macros ou des environnements

Voici ce qu'il est possible de taper de façon sémantique.



Remarque. L'intérêt des macros précédentes vis à vis de l'usage de \tdocinlatex, voir la section 1. page 10, est l'absence de coloration. De plus, la macro \tdocenv demande juste de taper le nom de l'environnement 5 avec des éventuelles options en tapant les bons délimiteurs 6 à la main.



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.

VI. 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.

^{5.} De plus, \tdocenv{monenv} produit \begin{monenv} ...\end{monenv} avec des espaces afin d'autoriser des retours à la ligne si besoin.

^{6.} Se souvenir que tout est possible ou presque dorénavant.

```
\tdocpre{sup} est relatif à...
sup est relatif à...
sup est relatif à...
sup erbe signifie...
sup erbe signifie...
sup er 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.

VII. Un rendu en situation réelle

Il est parfois utile d'obtenir directement le rendu d'un code dans la documentation. Ceci nécessite que ce type de rendu soit dissociable du texte donnant des explications.

1. Avec une bande colorée

Exemple 1 (Avec les textes par défaut). Il peut être utile de montrer un rendu réel directement dans un document. 7 Ceci se tape via \begin{tdocshowcase} ... \end{tdocshowcase} comme suit.

```
\begin{tdocshowcase}
\bfseries Un peu de code \LaTeX.
\bigskip
\emph{\large Fin de l'affreuse démo.}
\end{tdocshowcase}
```

On obtient alors le rendu suivant. 8

■ Début du rendu réel ■

Un peu de code PTEX.

Fin de l'affreuse démo.

Fin du rendu réel

Remarque. Voir la section 4. page 12 pour obtenir facilement un code suivi de son rendu réel comme dans l'exemple précédent.

i Note.

Les textes explicatifs s'adaptent à la langue choisie lors du chargement de tutodoc.

Exemple 2 (Changer la couleur et/ou les textes par défaut).

Ceci produira ce qui suit.

^{7.} Typiquement lorsque l'on fait une démo.

^{8.} En coulisse, la bande est créée sans effort grâce au package clrstrip.

Vous avez

Vous avez sûrement remarqué que le rouge sert de base pour obtenir les couleurs utilisées.

- La couleur de fond est fournie par \tdocbackcolor.
- La couleur des titres et des lignes est fournie par \tdocdarkcolor.

Ces macros développables et à un seul argument, la couleur choisie, admettent les codes suivants.

Il faut également savoir qu'en coulisses, la macro \tdocruler est utilisée.

```
\tdocruler{Un pseudo-titre décoré}{red}

Un pseudo-titre décoré
```

Avertissement.

Avec les réglages par défaut, si le code \LaTeX à mettre en forme commence par un crochet ouvrant, il faudra user de \string comme dans l'exemple suivant.

```
\begin{tdocshowcase}
\string[Cela fonctionne...]
\end{tdocshowcase}
```

Ceci produira ce qui suit.

```
[Cela fonctionne...]

Fin du rendu réel
```

2. Sans bande colorée

Le rendu de \begin{tdocshowcase}\end{tdocshowcase} avec une bande colorée peut ne pas convenir, ou parfois ne pas être acceptable malgré le travail fait par clrstrip. Il est possible de ne pas utiliser une bande colorée comme nous allons le voir tout de suite.

Exemple 1. L'emploi de \begin{tdocshowcase} [nostripe] ... \end{tdocshowcase} demande de ne pas faire appel à clrstrip. Voici un exemple d'utilisation.

Exemple 2 (Changer la couleur et/ou les textes par défaut).

```
\begin{tdocshowcase}[nostripe,
before = Mon début,
after = Ma fin à moi,
color = green]
```

■ Fin du rendu réel **■**

3. En important le code LATEX

Pour obtenir des rendus en important le code depuis un fichier externe, au lieu de le taper, il suffit d'employer la macro \tdocshowcaseinput dont l'option reprend la syntaxe de celle de \begin{tdocshowcase} ...\end{tdocshowcase} et l'argument obligatoire correspond au chemin du fichier.

Exemple. Ce qui suit a été obtenu via \tdocshowcaseinput{external.tex}.

Début du rendu réel
Blablobli, blablobli, blablobli, blablobli, blablobli
Fin du rendu réel
$Quant \ a \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
Début du rendu réel
Blablobli, blablobli, blablobli, blablobli, blablobli, blablobli

VIII. Cas d'utilisation en I⁴T_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. 9

Avertissement.

La version 3 de minted ne peut pas être prise en compte pour le moment car elle comporte des bugs : voir https://github.com/gpoore/minted/issues/401. On force donc l'usage de la version 2 de minted.

1. Codes « en ligne »

La macro \t docinlatex 10 permet de taper du code en ligne via un usage similaire à \t verb. Voici des exemples d'utilisation.

```
1: \tdocinlatex|\$a^b = c\$|
2: \tdocinlatex+\tdocinlatex|\$a^b = c\$|+
2: \tdocinlatex|\$a^b = c\$|
```

1 Note.

La macro \tdocinlatex est utilisable dans une note de pied de page : voir ci-dessous. De plus, une couleur de fond est volontairement utilisée pour subtilement faire ressortir les codes \LaTeX.

^{9.} La mise en forme des codes se fait via le package minted.

^{10.} Le nom de la macro \tdocinlatex vient de « in·line LATEX » soit « LATEX en ligne » en anglais.

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

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

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 . 11

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

```
\$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]
    [Etrange... Ou pas !]
\end{tdoclatex}
```

Ceci produira ce qui suit.

```
[Étrange... Ou pas !]
```

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

```
\begin{tdoclatex}
\string[\tilde{E}trange... Ou pas !]
\end{tdoclatex}
```

Ceci produira ce qui suit.

^{11.} std fait référence au comportement « standard » de tcolorbox vis à vis de la librairie minted.

```
[Étrange... Ou pas !]
```

3. Codes importés

Pour les codes suivants, on considère un fichier de chemin relatif examples-listing-xyz.tex, et ayant le contenu suivant.

```
% Juste une démo.
$x y z = 1$
```

La macro \tdoclatexinput, présentée ci-dessous, attend le chemin d'un fichier et propose les mêmes options que l'environnement \begin{tdoclatex}...\end{tdoclatex}.

Exemple 1 (Face à face).

```
\to tdoclatexinput[sbs]{examples-listing-xyz.tex}
```

Ceci produit la mise en forme suivante.

```
% Juste une démo.
$x y z = 1$
xyz = 1
```

Exemple 2 (À la suite).

Ceci produit la mise en forme suivante où l'option employée par défaut est std.

Exemple 3 (Juste le code).

```
\top the thing-the thing
```

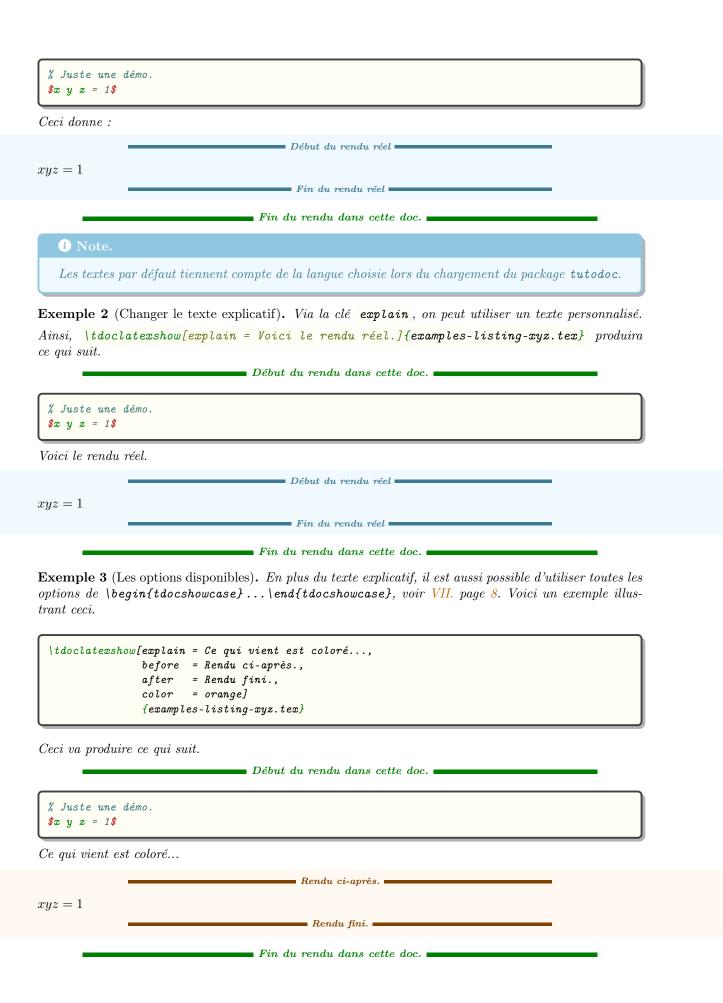
Ceci produit la mise en forme suivante.

```
\% Juste une démo. \$x\ y\ z\ =\ 1\$
```

4. Codes importés et mis en situation

Exemple 1 (Showcase). Ce qui suit s'obtient via \tdoclatexshow{examples-listing-xyz.tex}.

Début du rendu dans cette doc.



IX. Indiquer les changements

Afin de faciliter le suivi d'un package, il est indispensable de fournir un historique indiquant les changements effectués lors de la publication d'une nouvelle version.

1. À quel moment?

On peut au choix dater quelque chose, ou bien le versionner, dans ce second cas le numéro de version pourra éventuellement être daté.

Exemple 1 (Dater des nouveautés). La macro \tdocdate permet d'indiquer une date dans la marge comme dans l'exemple suivant.

Ceci donne:

■ Début du rendu réel |

2023-09-24 2020-05-08

Fin du rendu réel

Exemple 2 (Versionner des nouveautés en les datant événtuellement). Associer un numéro de version à une nouveauté se fait via la macro \tdocversion, la couleur et la date étant des arguments optionnels.

Ceci donne:

10.2.0-beta 01/12/2023

10.2.0-alpha

Fin du rendu réel

✓ Important.

- 1. Les macros \tdocdate et \tdocversion nécessitent deux compilations.
- 2. Comme la langue indiquée pour cette documentation est le français, la date dans le rendu final est au format JJ/MM/AAAA alors que dans le code celle-ci devra toujours être saisie au format anglais AAAA-MM-JJ.

& Avertissement.

Seul l'emploi du format numérique YYYY-MM-DD est vérifié, ^a et ceci est un choix! Pourquoi cela ? Tout simplement car dater et versionner des explications devrait se faire de façon semi-automatisée afin d'éviter tout bug humain.

a. Techniquement, vérifier la validité d'une date, via LATEX3, ne présente pas de difficulté.

2. Quoi de neuf?

tutodoc propose différents environnements pour indiquer rapidement et clairement ce qui a été fait lors des derniers changements. 12

Exemple 1 (Pour les nouveautés).

\begin{tdocnew} \item Info 1 \item Info 2 \end{tdocnew}	 ♦ Nouveau. • Info 1 • Info 2 	
---	--	--

Exemple 2 (Pour les mises à jour).

\begin{tdocupdate}	₩ Mise à jour.	
\item Info 1 \item Info 2	• Info 1	
$\ensuremath{\setminus} end\{tdocupdate\}$	• Info 2	

Exemple 3 (Pour les bifurcations).

\begin{tdocbreak}	🎖 Bifurcation.
\item Info 1 \item Info 2	• Info 1
\end{tdocbreak}	• Info 2

Exemple 4 (Pour les problèmes).

\begin{tdocprob} \item Info 1 \item Info 2 \end{tdocprob}	 Ряовіёме. Info 1 Info 2 	·
---	---	---

^{12.} L'utilisateur n'a pas besoin de tous les détails techniques.

Exemple 5 (Pour les réparations).

Exemple 6 (Thématiques aux choix avec une icône).

```
\begin{tdoctopic}{Ne pas}

→ regarder}[\faEyeSlash]

% Une icône venant de fontawesome5.

\item Info 1...
\item Info 2...
\end{tdoctopic}

NE PAS REGARDER.

• Info 1...

• Info 2...
```

Exemple 7 (Thématiques aux choix sans icône).

```
\begin{tdoctopic}{La fin des icônes}

% Ici le point s'impose.
\item Info 1...
\item Info 2...
\end{tdoctopic}

LA FIN DES ICÔNES.

• Info 1...
• Info 2...
```

X. Décorations

Finissons cette documentation avec de petits outils de mise en forme pouvant rendre de grands services.

```
Bla, bla, bla...
\tdocsep % Pratique pour délimiter.
Ceci fonctionne avec des énumérations.
                                                     Bla, bla, bla...
\begin{itemize}
    \item Point souligné.
                                                     Ceci fonctionne avec des énumérations.
    \item Autre chose utile.
\end{itemize}
                                                         • Point souligné.
                                                         • Autre chose utile.
\tdocsep % Un comportement uniforme.
Ble, ble, ble...
                                                     Ble, ble, ble...
Bli, bli, bli...
                                                     Bli, bli, bli...
                                                     Blo, blo, blo...
\tdocxspace % Espace subtile
                                                     Blu, blu, blu...
             % mais utile.
Blo, blo, blo...
Blu, blu, blu...
```

XI. Historique

1.3.0 25/09/2024

O PROBLÈME.

• La version 3 de minted ne peut pas être prise en compte pour le moment car elle comporte des bugs : voir https://github.com/gpoore/minted/issues/401. On force donc l'usage de la version 2 de minted.

P BIFURCATION.

• L'environnement tdocimportant a été renommé tdocimp pour une saisie simplifiée.

Nouveau.

- Journal des changements : les environnements proposés utilisent des icônes.
- Mise en avant de contenus : des cadres colorés avec des icônes sont proposés pour les environnements suivants.

1. tdoccaution

3. tdocnote

5. tdocwarn

2. tdocimp

4. tdoctip

1.2.0-a 23/08/2024

Mise à jour.

- \tdocversion
 - 1. Le numéro de version est au-dessus de la date.
 - 2. L'espacement est mieux géré lorsque la date est absente.

F RÉPARATION.

 $\bullet\,$ Mise en avant de contenus : les traductions françaises de « caution » et « danger » étaient erronées.

 $\frac{1.1.0}{06/01/2024}$

Nouveau.

- Journal des changements : deux nouveaux environnements.
 - 1. $\begin{tdocbreak} \dots \end{tdocbreak} pour les « bifurcations », soit les modifications non rétrocompatibles.$
 - 2. \begin{tdocprob} ...\end{tdocprob} pour les problèmes repérés.
- \tdocinlatex : un jaune léger est utilisé comme couleur de fond.

08/12/2023

F RÉPARATION.

- \tdocenv : l'espacement est maintenant correct, même si le paquet babel n'est pas chargé avec la langue française.
- \begin{tdocshowcase} [nostripe] ...\end{tdocshowcase} : les sauts de page autour des lignes « cadrantes » devraient être rares dorénavant.

 $\frac{1.0.0}{29/11/2023}$

Première version publique du projet.