

I. Introduction

Le package `bdoc`¹ facilite la saisie sémantique de documentations de packages L^AT_EX avec un rendu sobre pour une lecture sur écran², mais pratique pour les utilisateurs (ce package propose, ou impose, un style de mise en forme, mais il n'a pas pour vocation à être transformé en une classe).

II. Choisir la langue au chargement du package

La présente documentation utilise le français via `\usepackage[lang = french]{bdoc}`. Pour le moment, on a juste deux choix où la convention de nommage suit celle de `babel`.

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

III. Mises en forme générales imposées

1. Géométrie de la page

Le package `geometry` est chargé avec les réglages suivants.

```
\RequirePackage[
  top          = 2.5cm,
  bottom       = 2.5cm,
  left         = 2.5cm,
  right        = 2.5cm,
  marginparwidth = 2cm,
  marginparsep  = 2mm,
  heightrounded
]{geometry}
```

2. Titre et table des matières

Les packages `titlesec`, `tocbasic` et `bookmark` sont réglés comme suit où la redéfinition de `\thechapter` n'est utile que pour des packages agrégateurs (dont les documentations sont de type livre).

```
\RequirePackage[raggedright]{titlesec}

% ...
\ifcsundef{chapter}%
  {}%
  {\renewcommand\thechapter{\Alph{chapter}.}}

\renewcommand\thesection{\Roman{section}.}
```

1. Le nom vient de « *b·asic doc·umentation* » qui ne nécessite aucune traduction.
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.

```

\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   = 0pt,
    dynindent,
    numwidth = 0pt,
    numsep   = 1ex,
    dynnumwidth
]{tocline}{
    chapter,
    section,
    subsection,
    subsubsection,
    paragraph,
    subparagraph
}

\DeclareTOCStyleEntry[indentfollows = chapter]{tocline}{section}

```

3. Liens dynamiques

Les packages `hyperref` et `hyperref` sont aussi importés 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
}

```

IV. Indiquer des packages, des macros ou des environnements

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

<code>\bdocpack{monpackage}</code> est pour...	monpackage est pour...
<code>\bdocmacro{unemacro}</code> permet de...	<code>\unemacro</code> permet de...
<code>\bdocenv{env}</code> sert à...	<code>\begin{env} ... \end{env}</code> sert à...

Remarque. L'intérêt des macros précédentes vis à vis de l'usage de `\bdocinlatex` est l'absence de coloration. De plus, la macro `\bdocenv` demande juste de taper le nom de l'environnement³.

V. 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 retenus. Ceci se fait facilement comme suit.

<code>\bdocpre{sup}</code> est relatif à...	sup est relatif à...
<code>\bdocprewhy{sup.erbe}</code> signifie...	sup·erbe signifie...
<code>\emph{\bdocprewhy{sup.er} pour...}</code>	<i>sup·er pour...</i>

Remarque. Le choix du point pour scinder un mot permet d'utiliser des mots avec un tiret comme dans `\bdocprewhy{ca.sse-brique}` qui donne *ca·sse-brique*.

VI. Cela veut dire quoi en « *angliche* »

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

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

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

La macro `\bdocinEN` et sa version étoilée s'appuient sur `\bdocquote` : par exemple, le code `\bdocquote{sémantique}` produit « *sémantique* ».

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

VII. Un rendu en situation réelle

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. Ceci se tape comme suit.*

```
\begin{bdocshowcase}
  \bfseries Un peu de code \LaTeX.

  \bigskip

  \emph{\large Fin de l'affreuse démo.}
\end{bdocshowcase}
```

On obtient alors le rendu suivant⁴.

————— Début du rendu réel —————

Un peu de code \LaTeX .

Fin de l'affreuse démo.

————— Fin du rendu réel —————

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

Note. Il faut savoir qu'en coulisse la macro `\bdocruler` est utilisée.

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

————— Un pseudo-titre décoré —————

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

```
\begin{bdocshowcase}[before = Mon début,
                      after  = Ma fin à moi,
                      color   = red]
  Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...
\end{bdocshowcase}
```

Ceci produira ce qui suit.

————— Mon début —————

Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...

————— Ma fin à moi —————

Note.

- Les textes explicatifs s'adaptent à la langue choisie lors du chargement de `bdoc`.

4. La bande est créée sans effort grâce au package `clrstrip`.

- Vous avez sûrement noté que l'on n'obtient pas un rouge pur : en coulisse les macros développables `\bdocbackcolor` et `\bdocdarkcolor` sont utilisées pour créer, à partir de la couleur proposée à `\begin{bdocshowcase} ... \end{bdocshowcase}`, celle du fond et celle des titres respectivement. Ces macros à un seul argument, la couleur choisie, admettent les codes suivants.

```
\NewExpandableDocumentCommand{\bdocbackcolor}{m}{#1!5}

\NewExpandableDocumentCommand{\bdocdarkcolor}{m}{#1!50!black}
```

Avertissement. Avec la mise en forme par défaut, si le code $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ commence par un crochet ouvrant, il faudra indiquer explicitement une option vide comme dans l'exemple suivant.

```
\begin{bdocshowcase}[]
  [Cela fonctionne...]
\end{bdocshowcase}
```

Ceci produira ce qui suit.

————— Début du rendu réel —————

[Cela fonctionne...]

————— Fin du rendu réel —————

2. Sans bande colorée

Le rendu de `\begin{bdocshowcase} ... \end{bdocshowcase}` avec une bande colorée peut parfois ne pas être acceptable malgré le travail fait par `clrstrip`. Voyons comment contourner ce type de limitation.

Exemple 1. L'option booléenne `nostripe` sert à ne pas employer `clrstrip`. Voici un exemple d'utilisation.

```
\begin{bdocshowcase}[nostripe]
  Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...
\end{bdocshowcase}
```

Ceci produira ce qui suit.

————— Début du rendu réel —————

Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...

————— Fin du rendu réel —————

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

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

```
\end{bdocshowcase}
```

Ceci produira ce qui suit.

Mon début

Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...

Ma fin à moi

3. En important le code L^AT_EX

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

Ce qui suit a été obtenu via `\bdocshowcaseinput{examples/external.tex}`.

Début du rendu réel

Blablobli, blablobli, blablobli, blablobli, blablobli, blablobli...

Fin du rendu réel

Via `\bdocshowcaseinput[color = orange]{examples/external.tex}`, on change la coloration comme suit.

Début du rendu réel

Blablobli, blablobli, blablobli, blablobli, blablobli, blablobli...

Fin du rendu réel

VIII. Cas d'utilisation en L^AT_EX

1. Codes « en ligne »

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

1. `\bdocinlatex|$a^b = c$|` produit $a^b = c$.
2. `\bdocinlatex|$a^b = c$|` vient de `\bdocinlatex+\bdocinlatex|$a^b = c$|+` obtenu via `\bdocinlatex#\bdocinlatex+\bdocinlatex|$a^b = c$|+#` ... etc.

Important. La macro `\bdocinlatex` est utilisable dans une note de pied de page. En voici la preuve⁵.

Remarque. Le nom de la macro `\bdocinlatex` vient de « in·line L^AT_EX » soit « L^AT_EX en ligne » en anglais.

2. Codes tapés directement

Exemple 1 (Face à face). Dans le code suivant, l'option `sbs` est pour « s·ide b·y s·ide » soit « côte à côte » en anglais.

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

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

Ceci donne :

————— Début du rendu réel —————

$A = B + C$

$A = B + C$

————— Fin du rendu réel —————

Exemple 2 (À la suite). L’environnement `\begin{bdoclateg} ... \end{bdoclateg}` choisit par défaut l’option `std`⁶ et produit le résultat suivant.

$A = B + C$

$A = B + C$

Exemple 3 (Juste le code). Afficher le code seul comme ci-après s’obtient via l’option `code` de l’environnement `\begin{bdoclateg} ... \end{bdoclateg}`.

$A = B + C$

Avertissement. Avec la mise en forme par défaut, si le code commence par un crochet ouvrant, il faudra indiquer explicitement l’option par défaut. Voici un cas d’usage.

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

Ceci permet d’obtenir ce qui est attendu :

————— Début du rendu réel —————

[Étrange... Ou pas !]

[Étrange... Ou pas !]

————— Fin du rendu réel —————

3. Codes importés

Pour les codes suivants, on considère un fichier `examples/listing/xyz.tex` dont le chemin est donné relativement au document présent. Le contenu de ce fichier se réduit à l’unique ligne $x\ y\ z = 1$.

6. `std` fait référence au comportement « *standard* » de `tcolorbox` vis à vis de la librairie `minted`.

Noter ci-après que la macro `\bdoclaxinput` s'utilise de façon semblable à l'environnement `\begin{bdoclax} ... \end{bdoclax}` excepté que l'on fournit le chemin d'un fichier.

Exemple 1 (Face à face).

```
\bdoclaxinput[sbs]{examples/listing/xyz.tex}
```

Ceci produit la mise en forme suivante.

$x y z = 1$	$xyz = 1$
-------------	-----------

Exemple 2 (À la suite).

```
\bdoclaxinput{examples/listing/xyz.tex}
```

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

$x y z = 1$
$xyz = 1$

Exemple 3 (Juste le code).

```
\bdoclaxinput[code]{examples/listing/xyz.tex}
```

Ceci produit la mise en forme suivante.

$x y z = 1$

4. Codes importés et mis en situation

Ce qui suit est utile pour des exemples plus sophistiqués.

Exemple 1 (Showcase). Ce qui suit a été obtenu via la macro `\bdoclaxshow` en tapant `\bdoclaxshow{examples/listing/xyz.tex}`.

————— Début du rendu dans cette doc. —————

$x y z = 1$

Ceci donne :

————— Début du rendu réel —————

$xyz = 1$

————— Fin du rendu réel —————

————— Fin du rendu dans cette doc. —————

Exemple 2 (Changer le texte explicatif). Pour obtenir ce qui suit avec un texte personnalisé, il suffit d'employer `\bdoclaxshow[explain = Voici le rendu réel.]{...}`.

————— Début du rendu dans cette doc. —————

$\$x\ y\ z = 1\$$

Voici le rendu réel.

————— Début du rendu réel —————

$xyz = 1$

————— Fin du rendu réel —————

————— Fin du rendu dans cette doc. —————

Note. Le texte imprimé par défaut tient compte de la langue choisie lors du chargement du package `bdoc`.

Exemple 3 (Les options disponibles). En plus du texte explicatif, il est aussi possible d'utiliser toutes les options de `\begin{bdocshowcase} ... \end{bdocshowcase}`, voir [VII.](#) Voici un exemple illustrant ceci.

```
\bdoclatexshow[explain = Ce qui vient est coloré...,
               before  = Rendu ci-après.,
               after   = Rendu fini.,
               color    = orange]
{examples/listing/xyz.tex}
```

Ceci va produire ce qui suit.

————— Début du rendu dans cette doc. —————

$\$x\ y\ z = 1\$$

Ce qui vient est coloré...

————— Rendu ci-après. —————

$xyz = 1$

————— Rendu fini. —————

————— Fin du rendu dans cette doc. —————

IX. Mettre en avant du contenu

Note. Les environnements présentés dans cette section⁷ 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 `bdoc`.

1. Des exemples

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

⁷ La mise en forme provient du package `amsthm`.

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

Voici différents emplois possibles.

<pre>Bla, bla, bla...</pre> <pre>\begin{bdocexa} Ble, ble, ble... \end{bdocexa}</pre> <pre>\begin{bdocexa}[Magnifique] Bli, bli, bli... \end{bdocexa}</pre> <pre>\begin{bdocexa}<nonb> Blo, blo, blo... \end{bdocexa}</pre> <pre>\begin{bdocexa}<nonb>[Superbe] Blu, blu, blu... \end{bdocexa}</pre>	<pre>Bla, bla, bla...</pre> <p>Exemple 1. <i>Ble, ble, ble...</i></p> <p>Exemple 2 (Magnifique). <i>Bli, bli, bli...</i></p> <p>Exemple. <i>Blo, blo, blo...</i></p> <p>Exemple (Superbe). <i>Blu, blu, blu...</i></p>
--	--

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. Voici comment faire (ce tour de passe-passe reste valable pour les environnements présentés dans les sous-sections suivantes). Noter au passage que la numérotation suit celle de l'exemple précédent comme souhaité.

<pre>\begin{bdocexa} \leavevmode \begin{enumerate} \item Point 1. \item Point 2. \end{enumerate} \end{bdocexa}</pre>	<p>Exemple 3.</p> <ol style="list-style-type: none"> 1. Point 1. 2. Point 2.
--	---

2. Des remarques

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

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

3. Une astuce

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

<code>\begin{bdoctip}</code> Une astuce. <code>\end{bdoctip}</code>	Astuce. <i>Une astuce.</i>
<code>\begin{bdoctip}[Mini titre]</code> Utile ? <code>\end{bdoctip}</code>	Astuce (Mini titre). <i>Utile ?</i>

4. Note informative

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

<code>\begin{bdocnote}</code> Un truc utile à vous dire... <code>\end{bdocnote}</code>	Note. <i>Un truc utile à vous dire...</i>
<code>\begin{bdocnote}[Mini titre]</code> Utile ? <code>\end{bdocnote}</code>	Note (Mini titre). <i>Utile ?</i>

5. Un truc important

L'environnement `\begin{bdocimportant} ... \end{bdocimportant}` permet d'indiquer quelque chose d'important mais sans danger.

<code>\begin{bdocimportant}</code> Un truc important sans danger. <code>\end{bdocimportant}</code>	Important. <i>Un truc important sans danger.</i>
<code>\begin{bdocimportant}[Mini titre]</code> Utile ? <code>\end{bdocimportant}</code>	Important (Mini titre). <i>Utile ?</i>

6. Avertir d'un point très délicat

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

<pre>\begin{bdoccaution} Prudence, prudence... \end{bdoccaution</pre>	Prudence. <i>Prudence, prudence...</i>
<pre>\begin{bdoccaution}[Mini titre] Utile ? \end{bdoccaution</pre>	Prudence (Mini titre). <i>Utile ?</i>

7. Avertir d'un danger

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

<pre>\begin{bdocwarn} Evitez les dangers... \end{bdocwarn</pre>	Avertissement. <i>Evitez les dangers...</i>
<pre>\begin{bdocwarn}[Mini titre] Utile ? \end{bdocwarn</pre>	Avertissement (Mini titre). <i>Utile ?</i>

X. Changements

Exemple 1 (Expliquer dans le journal). *Pour fournir des explications efficaces dans l'historique des changements, bdoc propose de procéder comme suit.*

```
\begin{bdoctopic}[Un court changement accentué]
  ajout de \bdocmacro{unemacro}.
\end{bdoctopic}

\begin{bdoctopic}<Lister plusieurs changements>
% Ici le point s'impose.
  \begin{itemize}
    \item Modification de \bdocmacro{vieillemacro}.
    \item Création de \bdocenv{unenv}.
  \end{itemize}
\end{bdoctopic}
```

Ceci donne :

UN COURT CHANGEMENT ACCENTUÉ : ajout de `\unemacro`.

LISTER PLUSIEURS CHANGEMENTS.

- Modification de `\vieillemacro`.
- Création de `\begin{unenv} ... \end{unenv}`.

Exemple 2 (Dater des nouveautés). Pour indiquer la date d'une nouveauté, il suffit d'employer la macro `\bdocdate` comme dans l'exemple suivant où la date doit être donnée au format anglais AAAA-MM-JJ.

Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...

`\medskip`

`\bdocdate{2023-09-24}`

Ble, ble, ble, ble, ble, ble, ble, ble, ble, ble, ble, ble...

`\medskip`

`\bdocdate[gray]{2020-05-08}`

Bli, bli, bli, bli, bli, bli, bli, bli, bli, bli, bli, bli...

Blo, blo, blo, blo, blo, blo, blo, blo, blo, blo, blo, blo...

Blu, blu, blu, blu, blu, blu, blu, blu, blu, blu, blu, blu...

Ceci donne :

Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...

Ble, ble, ble, ble, ble, ble, ble, ble, ble, ble, ble...

Bli, bli, bli, bli, bli, bli, bli, bli, bli, bli, bli...

Blo, blo, blo, blo, blo, blo, blo, blo, blo, blo, blo...

Blu, blu, blu, blu, blu, blu, blu, blu, blu, blu, blu...

Exemple 3 (Versionner et dater des nouveautés). L'exemple suivant montre une macro plus complète que la précédente, à savoir `\bdocversion`.

`\bdocversion[red]{10.2.0-beta}{2021-07-05}`

Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...

`\bigskip % ATTENTION ! Ceci évite le chevauchement.`

[illegible]

Début du rendu réel

Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...

*Bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla,
bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla, bla,
bla, bla, bla, bla, bla, bla, bla, bla, bla, bla...*

Fin du rendu réel

1. Les macros `\bdocdate` et `\bdocversion` 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 donnée au format anglais `AAAA-MM-JJ`.
3. Le numéro de version sera du type `maj.min.bug` ou `maj.min.bug-extra` avec trois entiers `maj`, `min`, `bug` et un optionnel `extra` égal à `alpha`, `beta` ou `rc` pour « release candidate ».

Avertissement. Seul l'emploi du format YYYY-MM-DD est vérifié⁸, et ceci est un choix ! Pourquoi cela ? Tout simplement car dater et versionner des explications doit se faire de façon semi-automatisée afin d'éviter tout bug humain⁹.

Finissons cette documentation avec de petites outils de mise en forme qui rendent de grands services.

9. L'auteur de **bdoc** est entrain de mettre en place un ensemble d'outils permettant une telle semi-automatisation.

Bla, bla, bla...

`\bdocsep` *% Pratique pour délimiter.*

Ble, ble, ble...

Bli, bli, bli...

`\bdocxspace` *% Espace subtile
% mais utile.*

Blo, blo, blo...

Blu, blu, blu...

Bla, bla, bla...

Ble, ble, ble...

Bli, bli, bli...

Blo, blo, blo...

Blu, blu, blu...