

DOCUMENT EN FRANÇAIS AVEC L^AT_EX (BABEL)

Certains conseils donnés dans cette page ne sont plus optimaux avec les dernières évolutions des moteurs et des packages L^AT_EX.

- [Un premier test pour vérifier votre installation](#)
- [Les packages à inclure](#)
- [Les packages non-spécifiques nécessitant une adaptation](#)
- [Les commandes de french qui n'existent pas dans babel](#)
- [Quelques conseils pour écrire vos sources L^AT_EX utilisant babel](#)
- [Inclusion de fichiers graphiques](#)
- [Exemple d'en-tête complet](#)
- [Informations erronées ou manquantes](#)

Rédiger un document L^AT_EX en français nécessite l'utilisation de quelques packages spécifiques afin que ce document adopte les conventions habituelles de la typographie française. Si l'on veut en plus pouvoir créer un document PDF (grâce à pdf_latex), il y a quelques astuces à connaître (NdA : les informations spécifiques à l'utilisation de pdf_latex ne sont pas encore toutes présentes).

Il existe en gros trois méthodes pour franciser le fonctionnement de L^AT_EX :


1. Faire tous les réglages à la main en utilisant les commandes L^AT_EX de base.

C'est la solution la plus difficile à mettre en oeuvre. Cela nécessite de bien connaître L^AT_EX, TeX et les règles typographiques françaises... D'autres l'ont déjà fait pour vous. Donc inutile de réinventer la roue !

2. Utiliser le package french de B.Gaulle.

Ce fut longtemps la meilleure solution étant donné la très grande qualité du code proposé par B.Gaulle. L'installation de ce package n'était pas toujours facile mais les efforts consentis étaient récompensés.

Puis, le package french devint payant et nous ne pouvions donc plus le recommander ni le proposer en version pré-installée. Il existait bien une version `light` de ce package mais elle ne proposait pas les commandes indispensables permettant, à l'occasion, de supprimer l'un ou l'autre des réglages choisis par le mode french.

Après la disparition de B.Gaulle et avec l'accord de sa femme, ce package est devenu libre sous le nom eFrench et un groupe de volontaires le fait évoluer. Pour en savoir plus, visitez le [site officiel](#) .

3. Utiliser le package babel.

À l'inverse babel fut longtemps considéré comme une solution de dépannage. Mais, heureusement, le package s'est largement amélioré au cours du temps et il est maintenant réellement utilisable. De plus, il est intégré dans toutes les distributions L^AT_EX (TeXLive, MikTeX, MacTeX, etc.) fournies sur Windows, MacOS ou Linux.

C'est l'utilisation de ce dernier package que nous décrivons dans la suite de ce document.

Un premier test pour vérifier votre installation

Voici le source d'un petit document de test :

```
\documentclass{article}
\usepackage[latin1]{inputenc}
```

```
\usepackage[cyr]{aeguill}
\usepackage[francais]{babel}
\begin{document}
\showhyphens{signal container \text{événement algèbre}}
\showhyphens{signal container événement algèbre}
\end{document}
```

Compilez ce document avec la commande `latex` puis examinez le contenu du fichier `.log` produit. Vous devriez y trouver quelque chose ressemblant l'une de ces deux lignes :

```
si-gnal contai-ner évé-ne-ment al-gèbre
si-gnal contai-ner ^^Sev^^Se-ne-ment al-g^^Rebre
```

Si ce n'est pas le cas, votre installation n'est pas bien configurée pour gérer la langue française. Par exemple, si vous obtenez `sig-nał` et `con-tainer`, cela signifie que vous utilisez les motifs de césure américain au lieu des français... Si vous obtenez aucune césure pour `événement`, cela signifie que vous utilisez le codage OT1 qui ne permet pas (par défaut) de gérer correctement les césures.

Si la première ligne est correcte mais pas la seconde alors vous devrez choisir un autre encodage que `latin1` (voir plus bas pour les codages utilisables avec `inputenc`)

Si le résultat est celui attendu alors vous pouvez raisonnablement espérer utiliser `babel` pour rédiger vos documents français.

Les packages à inclure

Voici la liste des packages à inclure et des fonctionnalités qu'ils proposent :

`\usepackage[latin1]{inputenc}`

L'inclusion de ce package permet d'utiliser des sources LaTeX contenant des caractères accentués. L'option `[latin1]` précise que l'encodage des fichiers est `iso-latin1` aussi connu sous le nom `iso-8859-1`. Si vous êtes sur une plate-forme Un*x (Solaris, Linux, *BSD...), c'est certainement l'encodage que vous utilisez. Si vous travaillez dans un environnement Windows ou MacOS, vous devrez choisir une autre option (`[ansinew]` et `[applemac]` respectivement).

`\usepackage[cyr]{aeguill}`

Permet d'inclure la package `ae` en le complétant afin de pouvoir utiliser les guillemets français («»). Le package `ae` utilise des fontes virtuelles afin d'utiliser en interne la codage T1 (qui permet la césure correcte des mots accentués) et d'utiliser tout de même en externe les polices Computer Modern (OT1) qui sont les seules à exister en version PostScript Type1 gratuite. Ceci permet, par exemple, de faire des documents PDF de qualité sans être obligé d'utiliser les polices Adobe de base (Times, Courier...) ou de payer les polices EC en PostScript Type1.

`\usepackage{xspace}`

Permet à `babel` d'utiliser la macro `xspace` partout où c'est nécessaire. (Voir la doc de `babel` pour de plus amples explications.)

`\usepackage[francais]{babel}`

Ceci permet de charger le package `babel` en lui indiquant que l'on veut travailler en français. Attention : `[francais]` signifie 'french babel'. Si vous mettez `[french]` à la place, `babel` essaiera de trouver le package `french` de B.Gaule.

Depuis la version 3.6 de `babel`, on peut utiliser `[frenchb]` à la place de `[francais]`.

On peut aussi utiliser `babel` pour écrire un document en plusieurs langues. Pour cela, il faut lui donner en option la liste de toutes les langues qui seront utilisées en plaçant en dernier la langue principale du document.

Les packages non-spécifiques nécessitant une adaptation

Voici quelques packages utiles qui, du fait de l'utilisation de `babel` en français, nécessitent quelques adaptations :

`\usepackage{url}`

Ce package permet d'afficher correctement des URL dans un document LaTeX. Un URL contient toujours le caractère : (deux points). Or ce caractère devient actif en français. Il faut donc momentanément désactiver ce caractère lors de l'utilisation de la commande `\url` :

```
\let\urlorig\url
\renewcommand{\url}[1]{%
  \begin{otherlanguage}{english}\urlorig{#1}\end{otherlanguage}%
}
```

Cette astuce nécessite l'utilisation momentanée de la langue anglaise lors de l'affichage de l'URL. Il faut donc demander à `babel` de charger aussi cette langue :

```
\usepackage[english,français]{babel}
```

(Merci à Daniel FLIPO (flipo@jacta.univ-lille1.fr) pour l'amélioration de cette astuce qui ne sera plus nécessaire dès que `babel` v3.7 sera sorti.)

Les commandes de french qui n'existent pas dans babel

Le package `french` propose toute une série de commandes supplémentaires permettant de faciliter la conception de documents français. Ces commandes ne sont bien évidemment pas disponibles dans `babel`. Pour la plupart, il existe une manière de faire compatible avec `babel` (le fonctionnement ne sera pas toujours exactement similaire à la commande équivalente de `french`).

`\sommaire`

Dans un document français, le sommaire est au début du document et la table des matières est à la fin du document (en général, on utilise l'un ou l'autre mais pas les deux ;-). Or `babel` traduit le titre de la commande `\tableofcontents` par « Table des matières ». Pour utiliser cette commande pour réaliser un sommaire, il suffit de changer sa traduction dans votre document :

```
\renewcommand{\contentsname}{Sommaire}
```

`\Numero` et `\numero`

Ces deux commandes ont leur équivalent dans `babel` :

```
\newcommand{\Numero}{\No}
\newcommand{\numero}{\no}
```

`\fup`

Cette commande a un équivalent dans `babel` :

```
\newcommand{\fup}[1]{\up{#1}}
```

Quelques conseils pour écrire vos sources LaTeX utilisant babel

Les caractères accentués

Dans la mesure du possible, utilisez les caractères accentués directement dans votre source. L'utilisation du package `inputenc` vous le permet. Ne vous en privez pas.

Par contre, il est impossible d'utiliser les accents directement dans une formule mathématique. Pour obtenir quand même des accents, utilisez la commande `\text`. Exemple :

```
\begin{equation}
  A = E_{\text{é}}
\end{equation}
```

La commande `\text` n'est disponible que si vous utilisez le package `amsmath` :

```
\usepackage{amsmath}
```

Les guillemets

Utilisez les caractères « et » plutôt que `<<` et `>>` sinon (avec le package `aeguill`) vos guillemets ne s'afficheront pas correctement. Rappel : on place toujours une espace après l'ouverture des guillemets et avant la fermeture des guillemets. Si vous utilisez « et », il vous faudra taper `«~texte~»` pour que les espacements soient corrects.

Mais le mieux est d'utiliser les deux commandes `\og` (pour «) et `\fg` (pour ») qui ont l'avantage de mieux régler l'espacement entre les guillemets et ce qu'ils entourent puisqu'elles utilisent un espacement variable à la place de l'espace insécable créée par `~`.

Les acronymes, siglaisons et nom propres

`babel` propose la commande `\bsc` pour placer un bout de texte en petites capitales. Cela peut être pratique pour les acronymes, siglaisons et les noms propres. Exemple :

```
La \bsc{sncf} embauche Jacques \bsc{Durand}.
```

Les ponctuations double (:, ?, !, ;, etc.)

Placez toujours un caractère espace avant une double ponctuation ! `babel` le transformera tout seul en une espace insécable. Par défaut, `babel` ajoutera même cette espace si vous l'oubliez mais cela peut poser un problème dans les rares cas où il ne serait pas nécessaire. Il vaut donc mieux y penser soi-même et demander à `babel` de ne pas l'ajouter tout seul :

```
\NoAutoSpaceBeforeFDP
```

Les symboles divers

`babel` propose deux commandes pour afficher les degrés : `\degre` et `\degres`. La première sert pour les angles et la seconde pour les températures.

Quelques commandes sont disponibles pour des caractères spéciaux : `\at` (@), `\boi` (\), `\tilde` (~), `\circonflexe` (^).

1^{er}, 2^e, etc.

`babel` fournit les commandes suivantes pour les abréviations de premier, deuxième, etc. : `\ier`, `\iers`, `\iere`, `\ieres`, `\ieme` et `\iemes`. Ces commandes sont pratiques car elles utilisent les bonnes règles d'affichage des ces abréviations.

On peut aussi utiliser les commandes `\primo`, `\secundo`, `\tertio` et `\quarto`.

Inclusion de fichiers graphiques

Il existe de nombreuses méthodes pour créer des images en LaTeX. Quelques-unes sont internes à LaTeX. D'autres font appel à des packages externes (ou extensions). D'autres encore utilisent des fichiers externes. Les plus complexes combinent différentes techniques. Voici quelques conseils pour faciliter la production de documents LaTeX en français soit en PostScript soit en PDF.

L'environnement LaTeX `picture` et son extension `epic`

Cet environnement permet de créer des images directement dans le source LaTeX. L'extension `epic` y ajoute quelques améliorations. Mais cela reste d'une utilisation peu pratique sauf pour de très petites figures. Par contre, un dessin conçu via ces méthodes ne posera aucun problème pour produire un fichier PostScript (via `dvips`) ou PDF (via `pdflatex`).

Exemple d'en-tête complet

Voici un exemple d'en-tête pour un fichier source LaTeX utilisant les conseils précédents :

```
\usepackage[latin1]{inputenc}
\usepackage[cyr]{aequill}
\usepackage{xspace}
\usepackage{amsmath}
\usepackage[english,français]{babel}
\usepackage{url}
\let\urlorig\url
\renewcommand{\url}[1]{%
  \begin{otherlanguage}{english}\urlorig{#1}\end{otherlanguage}%
}
\renewcommand{\contentsname}{Sommaire} % si tableofcontents au début
\newcommand{\Numero}{\No}
\newcommand{\numero}{\no}
\newcommand{\fup}[1]{\up{#1}}
%%% N'oubliez pas les espaces devant les doubles ponctuations
\NoAutoSpaceBeforeFDP
```

Informations erronées ou manquantes

Ce document n'est absolument pas exhaustif. Je l'enrichirai au fur et à mesure de mes propres expérimentations et des retours des utilisateurs. N'hésitez donc pas à me poser des questions à ce sujet.