

Manuel de Prawn-for-book
Philippe Perret

Avant-propos

Description

Ce manuel présente toutes les fonctionnalités, à jour, de l'application « Prawn-for-book » (« Prawn pour les livres »), application dont la principale vocation est d'**obtenir un document PDF professionnel prêt pour l'imprimerie** à partir d'un simple fichier de texte (contenant le contenu du livre, le roman par exemple).

Noter que ce manuel est lui-même produit par « Prawn-for-book ». Si vous êtes intéressé(e) pour voir comment, et retrouver les deux fichiers `texte.pfb.md` et `recipe.yaml` qui le définissent, ouvrez le dossier `Manuel/manuel_building` de l'application.

{{Ajouter la possibilité d'ouvrir ce dossier dans `pfb open`.}}

Les deux fichiers de base

Un livre *Prawn-for-book* se compose à la base de deux fichiers :

- le fichier `texte` **`texte.pfb.md`**, contenant le contenu textuel (mais aussi images) du livre,
- le fichier `recette` **`recipe.yaml`** qui définit l'aspect du livre, à commencer par sa taille, ses marges, sa police par défaut, etc.

Fonctionnalités

Les Forces de *Prawn-for-book*

Prawn-for-book possède de nombreuses forces et de nombreux atouts dont nous ne pouvons que donner un aperçu non exhaustif dans cette introduction.

- Sans configuration ou définitions, l'application produit un document PDF valide pour l'imprimerie professionnelle respectant toutes les normes et les habitudes en vigueur, avec par exemple toutes les lignes de texte alignées sur une grille de référence, avec une mise en page en contenant aucune veuve, aucune orpheline et aucune ligne de voleur, avec un contenu gérant les références croisées, les index et les bibliographies, avec un texte respectant toutes les consignes typographiques.
- Bien que **Prawn-For-Book** soit capable de produire de façon automatique ce document, l'application offre la possibilité de définir tous les éléments de façon très précise et très fine pour obtenir le rendu exact souhaité, grâce à une *recette* qui accompagne le texte et où peuvent être paramétrés tous les aspects du livre.
- Plus loin encore, **Prawn-For-Book** permet de travailler comme aucun autre logiciel sur **une collection entière**, grâce à un *fichier recette* partagé par tous les livres — chacun pouvant définir sa propre recette pour rectifier des aspects particuliers. De cette manière, on peut très simplement obtenir une collection cohérente partageant la même charte. On peut même obtenir des références croisées dynamiques entre les différents livres.

Le reste de ce manuel vous permettra de découvrir l'ensemble des fonctionnalités à votre disposition.

Les deux fichiers de base

Un livre **Prawn-For-Book** peut se définir, au minimum, par deux fichiers fonctionnant de paire :

- un fichier définissant le *texte du livre*, donc le **contenu**,
- un fichier définissant l'*aspect du livre*, appelé « recette ».

Le fichier contenu

Le fichier du texte porte impérativement le nom `texte.pfb.md` et contient l'ensemble du contenu du livre. Il peut aussi faire appel à des *helpers* qui produiront du contenu dynamique.

Par exemple, si c'est un dictionnaire, on peut imaginer que le livre travaille en parallèle avec une base de données contenant la définition des mots.

Les bibliographies sont aussi des exemples de contenus qui peuvent être gérés automatiquement, pour un confort plus grand, une agilité incroyable et une cohérence à toute épreuve.

Le fichier porte l'extension `pfb.md` qui est composée de `pfb` pour « Prawn For Book » et de `md`, extension naturelle du *markdown* qui est le langage qui a inspiré l'établissement du contenu. Comme nous le verrons, c'est cependant un *pseudo-markdown* qui est utilisé.

Le fichier recette

Il porte impérativement le nom `recipe.yaml` quand il concerne un livre et `recipe_collection.yaml` quand il concerne une collection (nous reviendrons en temps voulu sur cette distinction).

Comme l'indique son extension, ce fichier est au format `YAML`, un format très simple permettant de définir des données (cf. plus loin).

Définition de la fonte par défaut

Bien que l'application propose, clé en main, une fonte qui peut être utilisée pour imprimer un livre, on peut définir n'importe quelle fonte comme fonte par défaut, et même une fonte personnelle dont on aura au préalable acheté la licence (c'est presque toujours nécessaire si le livre doit être vendu).

Si `recipe.yaml` ou `recipe_collection.yaml` contient...

```
book_format:
  text:
    default_font_and_style: "Times-Roman/"
    default_font_style: ""
    default_font_size: 20
```

Le livre final (document PDF) contiendra :

Pour écrire ce texte, nous avons ponctuellement modifié la

police par défaut en utilisant la fonte Times-Roman avec une taille de 20 et le style .

Texte au format pseudo markdown

Le fichier texte du livre porte le nom « **texte.pfb.md** ».

Ce nom obligatoire permet d'utiliser l'application *Sublime Text* (<https://www.sublimetext.com>) pour rédiger son texte avec de nombreuses aides qui simplifient considérablement la tâche et permettent de se concentrer sur le contenu du livre. Notez cependant, si vous connaissez le format Markdown (ou *Multimarkdown*), que ça n'est pas du pur format Markdown et que certaines fonctionnalités ne sont pas prises en compte. Vous trouverez en annexe, page 18, l'intégralité des marqueurs utilisables.

Les Tables

Bien que les tables soient très pratiques pour mettre en forme des données tabulaires, elles sont à utiliser prudemment dans **Prawn-For-Book** car elles ont le tort de ne pas aligner leur contenu sur la grille de référence et perturbent donc l'harmonie des pages.

Le Mode Expert

Le *mode Expert* permet d'ouvrir de façon exponentielle les possibilités de **Prawn-For-Book** afin de produire les contenus les plus variés et les plus originaux.

Il demande une compétence particulière dans le langage Ruby ainsi qu'une bonne connaissance de la gem `Prawn` qui permet de produire les livres avec **Prawn-For-Book**.

Annexe

Marques markdown

Vous trouverez ci-dessous toutes les marques Markdown utilisables.

Si texte.pfb.md contient...

La table ci-dessous a été construite aussi en markdown, en utilisant le format :

```
| CA1 | CA2 | CA3 | CA4 |  
| CB1 | CB2 | CB3 | CB4 |  
|/|
```

Vous pouvez trouver toutes les informations sur l'utilisation des tables à la page 14.

Le livre final (document PDF) contiendra :

italique	* ... *	<i>texte en italique</i>
gras	** ... **	texte en gras
souligné	___...___	<u>texte souligné</u>
exposants	1^er	1 ^{er}
	1^re	1 ^{re}
	2^e	2 ^e
	X^e	X ^e

Liste exhaustive des fonctionnalités

- définition de la taille du livre
- pagination automatique et personnalisable
- définition de la fonte par défaut
- justification par défaut des textes
- colorisation des textes
- suppression des veuves et des orphelines
- suppression automatique des lignes de voleur
- nombreuses sortes de puces et puces personnalisées [[puces]]

- évaluation à la volée du code ruby (pour des opérations, des constantes, etc.)
- traitement des références croisées
 - traitement dynamique des références à d'autres livres
 - génération dynamique de contenu
 - corrige l'erreur typographique de l'apostrophe droit
 - corrige l'erreur typographique de l'absence d'espace avant et après les chevrons
 - corrige l'erreur typographique de l'espace avant et après les guillemets droits et courbes
 - corrige l'oubli de l'espace avant les ponctuations doubles
 - corrige l'erreur d'espace avant les ponctuations doubles (pose d'une insécable)
 - corrige l'absence d'espace insécable à l'intérieur des tirets d'exergue
 - changement de fonte (police) pour le paragraphe suivant
[[change_fonte_for_next_paragraph]]
 - placement sur n'importe quelle ligne de la page
 - exportation seulement du texte produit
 - exportation comme livre numérique (pur PDF)