

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 auto-produit par « **Prawn-for-book** ». Il en utilise toutes les fonctionnalités puisqu'il génère de façon automatique les exemples — donc sert aussi de test complet de l'application — et notamment les *helpers* de mise en forme, les références croisées ou les bibliographies.

Si vous êtes intéressé(e) de voir comment il est généré, vous pouvez consulter principalement le fichier markdown `texte.pfb.md`, le fichier ruby `prawn4book.rb` et le fichier recette yaml `recipe.yaml` qui le définissent, dans le dossier `Manuel/manuel_building` de l'application.

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 Recettes

Recette du livre

La *recette du livre* du livre est un fichier de nom `recipe.yaml` qui se trouve à la racine du dossier du livre. Son nom vient de « recipe » qui signifie *recette* en anglais et de `.yaml`, extension des fichiers au format simple `YAML` (cf. page 15).

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

#<book_data>

```
# ...

#</book_data>
#<book_format>
# ...
#</book_format>
```

Recette de la collection

La *recette de la collection* est un fichier de nom `recipe_collection.yaml` qui se trouve à la racine du dossier d'une collection de livres. Son nom vient de « recipe » qui signifie *recette* en anglais et de `.yaml`, extension des fichiers au format simple YAML (cf. page 15). Vous pouvez trouver ci-dessous les données propres à une collection.

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

```
---
#<collection_data>
# (données de la collection)
collection_data:
  name: "Nom de la collection"
  short_name: "Nom raccourci (pour les messages)"
#</collection_data>
# ...
```

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 ### REF: all_markdown ###, l'intégralité des marqueurs utilisables.

Les puces

On peut utiliser de nombreuses puces et les régler parfaitement à sa convenance en modifiant la recette du livre ou non.

Si texte.pfb.md contient...

* Ceci est la puce de base qui permet un affichage neutre des items de liste.

Le livre final (document PDF) contiendra :

- Ceci est la puce de base qui permet un affichage neutre des items de liste.
-

Puce losange

On peut définir une autre puce dans la 12 ou la 13, dans la partie `text:` de `book_format:`.

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

```
---
book_format:
  text:
    puce:
      text: :losange
      vadjust: 2
      hadjust: 4
      left: 20mm
      size: 18
```

Si `texte.pfb.md` contient...

- * Ceci est une puce losange de 18 points, remontée de 2 points, déplacée à droite de 4, et écartée du texte de 20 millimètres,
- * Le second item de liste est identique,
- * Ainsi que le troisième.

Le livre final (document PDF) contiendra :

- ◇ Ceci est une puce losange de 18 points, remontée de 2 points, déplacée à droite de 4, et écartée du texte de 20 millimètres,
 - ◇ Le second item de liste est identique,
 - ◇ Ainsi que le troisième.
-

Le format YAML de la recette

Le format *YAML* est un format très simple de présentation et de consignation des données. Il est utilisé dans **Prawn-For-Book** pour définir 12, que ce soit pour un livre unique (cf. page 12) ou pour une collection (cf. page 13).

Les données sont *imbriquées*, comme nous l'avons dit, pour s'y retrouver plus facilement, entendu que les recettes peuvent contenir de nombreuses informations. Voyez l'imbrication donnée en exemple ci-dessous.

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

```
---
table_de_donnees:
  sous_ensemble_liste:
    - premier item
    - deuxième item
  sous_ensemble_texte: "Ma donnée texte"
  un_nombre: 12
  une_date: "2023-11-22"
  quelquun:
    prenom: "Marion"
    nom: "MICHEL"
```
