

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") qui permet 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 avantages dont nous ne pouvons que donner un aperçu non exhaustif ici.

- sans configuration ou définitions, l'application produit un document déjà valide, avec par exemple toutes les lignes de texte alignées sur une grille de référence, aucune veuve, aucune orpheline et aucune ligne de voleur, gestion correcte des références croisées, des index et des bibliographies.
- mais les outils permettent de définir tous les éléments de façon très précise et très fine pour obtenir le rendu exact demandé.

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

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 que ça n'est pas du pur format Markdown et que certaines fonctionnalités ne sont pas prises en compte.

Alignement sur la grille de référence

De façon naturelle et sans aucune intervention de l'auteur ou du metteur en page, les lignes de texte sont alignées sur une `tt`(grille de référence) ce qui permet un affichage professionnel.

Cette grille de référence ne dépend que du paramètre `:line_height` (hauteur de ligne) dans la recette.

Note : pour l'exemple, nous avons demandé l'[affichage de la `tt`(grille de référence)](#afficher_grille_reference_et_marges) afin de voir que les lignes de texte sont effectivement parfaitement alignées, malgré le changement de taille de la police.

Produira dans le livre :

Un premier paragraphe dans une police de taille 20
alors que la hauteur de ligne est réglée ici à 30.

Un autre paragraphe dans une police de taille. Lorem ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum

lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum

ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum lorem ipsum

ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum

lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum ipsum lorem ipsum

ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum lorem ipsum

ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum ipsum lorem ipsum ipsum

lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

Le format pseudo-markdown

Le formatage pseudo-markdown permet de mettre le texte en forme de façon très simple, sans avoir à se soucier des raccourcis clavier ou des menus, par de simples caractères, comme montré ci-dessous.

Dans le fichier texte.pfb.md

Un **paragraphe en italiques**.

Un ****paragraphe en gras****.

Un __texte__ en souligné.

* Un item de liste court.

* Un item de liste plus long pour voir comment la ligne sera enroulée pour laisser la puce bien visible.

Produira dans le livre :

Un *paragraphe en italiques*.

Un **paragraphe en gras**.

Un texte en souligné.

– Un item de liste court.

– Un item de liste plus long pour voir comment la ligne sera enroulée pour laisser la puce bien visible.

Les puces

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

Dans le fichier texte.pfb.md

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

Produira dans le livre :

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

On peut définir une autre puce dans la recette du livre, dans la partie “text” et “book_format”.

Dans le fichier recipe.yaml ou recipe_collection.yaml

```
#<book_format>
book_format:
  ...
  text:
    ...
    puce:
      text: :losange
      vadjust: 2
      hadjust: 4
      left: 20mm
      size: 18
#</book_format>
```

Dans le fichier texte.pfb.md

* 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.

Produira dans le livre :

- ◇ 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.
-

Évaluation du code ruby

Tous les codes qui se trouveront entre `# { . . . }` seront évalués en tant que code ruby, dans le cadre du livre (c'est-à-dire qu'ils pourront faire appel à des méthodes personnalisées) Typiquement, on peut par exemple obtenir la date courante.

Dans le fichier `texte.pfb.md`

Une opération simple permet de savoir que $2 + 2$ est égal à 4 et que le jour courant (au moment de l'impression de ce livre) est le 27 10 2023.

Produira dans le livre :

Une opération simple permet de savoir que $2 + 2$ est égal à 4 et que le jour courant (au moment de l'impression de ce livre) est le 27 10 2023.

Liste exhaustive des fonctionnalités

- suppression des veuves et des orphelines
- suppression 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.)
- 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)
 - changement de fonte (police) pour le paragraphe suivant
[[change_fonte_for_next_paragraph]]
-