





**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 46).

*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 46). 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.

- 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.
-

## Puce losange noir

*Si recipe.yaml ou recipe\_collection.yaml contient...*

---

```
#<book_format>
book_format:
  text:
    puce:
      text: :black_losange
      size: 14
      vadjust: -2
#</book_format>
```

*Si texte.pfb.md contient...*

---

- \* Ceci est une puce :black\_losange de 14 points, descendue de 2 points avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

*Le livre final (document PDF) contiendra :*

---

- ◆ Ceci est une puce :black\_losange de 14 points, descendue de 2 points avec les autres paramètres laissés intacts,
  - ◆ Le second item identique,
  - ◆ Le troisième.
- 

## Puce carrée

*Si recipe.yaml ou recipe\_collection.yaml contient...*

---

```
#<book_format>
book_format:
  text:
    puce:
      text: :square
      size: 20
      vadjust: 2
```



#</book\_format>

*Si texte.pfb.md contient...*

---

- \* Ceci est une puce :square de 20 points, remontée de 2 points, avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

*Le livre final (document PDF) contiendra :*

---

- Ceci est une puce :square de 20 points, remontée de 2 points, avec les autres paramètres laissés intacts,
  - Le second item identique,
  - Le troisième.
-

## Insérer une image

Pour insérer une image dans le flux du livre, on utilise le code pseudo-markdown **![chemin/vers/image]**. Si des données sont à passer, on utilise **![vers/image](<data>)** où <data> est une **### REF: table\_ruby ###**.

Noter que ce code doit obligatoirement se trouver sur une ligne seule.

### Chemin d'accès à l'image

Le chemin d'accès à l'image peut se donner...

- relativement au dossier du livre,
- relativement au dossier 'images' du livre (s'il existe),
- relativement au dossier de la collection s'il y a collection,
- relativement au dossier 'images' de la collection s'il y a collection.

Les formats (jpg, tiff, png, svg)

### Positionnement de l'image

Par défaut, l'image est placée au centre. Pour la mettre à droite ou à gauche, utiliser respectivement `align: right`, `align: left` et/ou la définition de `left` (décalage avec la marge gauche) et `right` (décalage avec la marge droite), comme dans les exemples ci-dessous.

Par défaut, l'image s'insère de façon fluide et naturelle avec le texte qui la précède et qui la suit. Mais on peut définir l'espacement de façon très précise avec les propriétés `space_before` (« espace avant ») et `space_after` (« espace après »), comme dans les exemples suivant (voir l'exemple 18, par exemple, à la page 30).

Pour un placement optimum, on peut ajuster la position de l'image verticalement à l'aide de `vadjust` (« ajustement vertical »). L'ajustement horizontal, bien entendu, peut se faire avec `right` et `left`. Bien sûr, il est raisonnable de ne faire ces ajustements qu'à la dernière minute, lorsque le livre est presque finalisé.

### Taille de l'image

La taille de l'image peut être définie avec `width:` (largeur), `height:` (hauteur) et/ou `scale:` (échelle). Notez que la largeur sera toujours ramenée à la largeur du livre si elle est supérieure, et la hauteur sera toujours ramenée à sa hauteur de la page si elle excède. Si vraiment on sait ce qu'on veut faire et qu'on désire tout contrôler soi-même, on peut ajouter l'option `no_resize: true` qui empêchera tout redimensionnement (à vos risques et péril [] comme dans l'exemple 19 à la page 31).

### Légende de l'image

Une légende peut être définie en donnant la donnée `legend:` "La légende de l'image". Par défaut, cette légende sera inscrite avec la même police que la police courante, avec une taille de une unité inférieure et en italique.

{{TODO: Modification, par la recette, de a) la taille, b) le style, c) la police, d) la largeur de la légende}}

## Image SVG

Les images SVG sont incontestablement les meilleures images pour l'impression dans le sens où elles conservent leurs qualités quelle que soit la taille adoptée. On peut le voir sur les exemples ci-dessous. Elles sont donc à privilégier.

## Rogner une image SVG

Rogner une image svg ### REF: image\_rogner\_svg ###.

{{TODO: les data}}

## Exemples

Les exemples ci-dessous sont tous construits à partir du code exact donné en exemple. Vous pouvez donc lui faire pleinement confiance.

EXEMPLE 1 • Une image trop large sera réduite à la taille du livre.

! [images/exemples/plus\_large.jpg]



EXEMPLE 2 • La même image, mais avec une légende, insérée (notez qu'ici on n'a pas indiqué images dans le chemin d'accès, puisque ce dossier est implicite).

! [exemples/plus\_large.jpg] (legend: "Image plus grande réduite en largeur")



Image plus grande réduite en largeur

**EXEMPLE 3 •** Une image PNG trop haute, *sans légende*, qui sera réduite à la taille d'une page du livre et mise sur la page suivante.  
![exemples/plus\_haute.png]

# NOTTURNO

für das Pianoforte  
von  
**RIEDRICH CHOPIN.**  
Nachgelassenes Werk. (Op. 72. N<sup>o</sup> 1)



C. XIII. 31.

**EXEMPLE 4 •** Une image PNG trop haute, *avec légende*, qui sera réduite à la taille d'une page du livre et mise sur la page suivante.

```
![exemples/plus_haute.png](legend:"Image plus haute réduite  
et<br>placée sur la page suivante")
```

# N O T T U R N O

für das Pianoforte  
von  
**RIEDRICH CHOPIN.**  
Nachgelassenes Werk. (Op. 72. N<sup>o</sup> 1)

C. XIII. 31.

*Image plus haute réduite et  
placée sur la page suivante*



**EXEMPLE 5 •** Une image PNG trop haute, dont on réduit explicitement la hauteur.

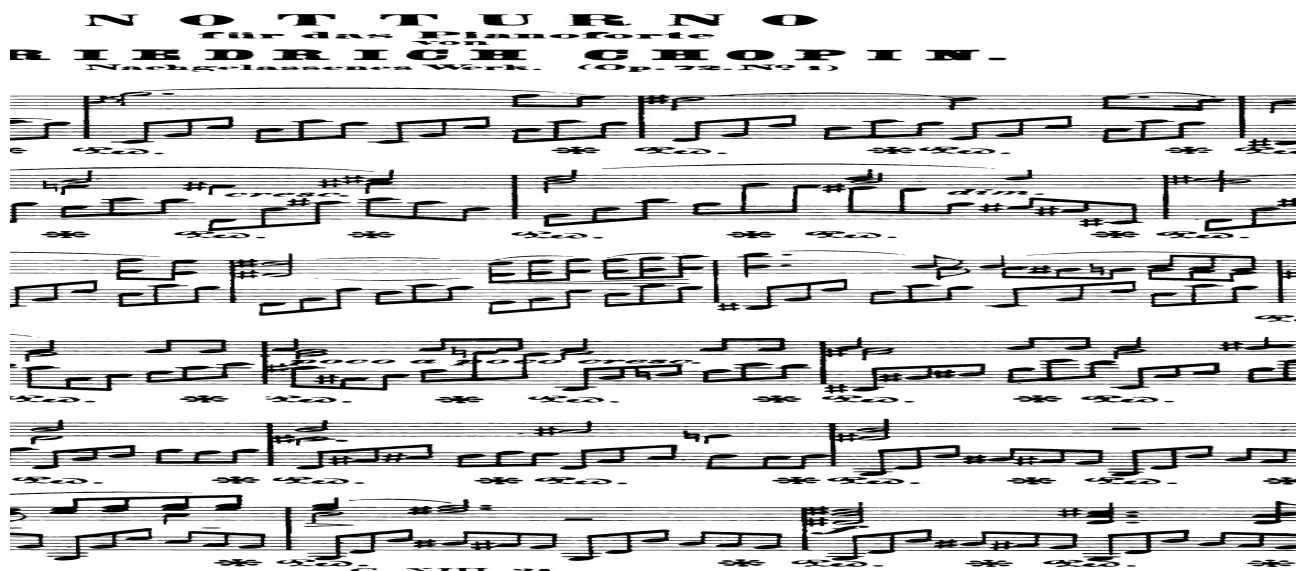
`![exemples/plus_haute.png](height:"30%", legend:"Trop haute avec height à 30%")`



*Trop haute  
avec height à  
30%*

**EXEMPLE 6 •** Une image PNG trop haute, dont on réduit explicitement la hauteur mais qui doit faire la largeur de la page. Elle va bien entendu être complètement déformée...

`![exemples/plus_haute.png](height:"30%", width:"100%", legend:"Trop haute avec height à 30% et width à 100%")`



*Trop haute avec height à 30% et width à 100%*

**EXEMPLE 7 •** Une image plus petite que la largeur sera laissée telle quelle.

`![exemples/moins_large.jpg](legend:"Image centrée")`



*Image centrée*

**EXEMPLE 8 •** Image plus petite en largeur, alignée à gauche et décalée vers la droite, sans légende.

`![exemples/moins_large.jpg](left: "2cm")`



**EXEMPLE 9 •** Image plus petite en largeur, avec légende, alignée à gauche et décalée vers la droite. Noter comment la légende a été découpée pour passer à la ligne. Noter également l'utilisation d'un ajustement vertical pour l'image (`vadjust`) et pour la légende (`vadjust_legend`).

`![exemples/moins_large.jpg](left: "2cm", vadjust: 1, legend: "Image à gauche décalée vers la droite", vadjust_legend: 10)`



*Image à gauche  
décalée  
vers  
la droite*

**EXEMPLE 10 •** Image plus petite en largeur alignée à droite.

`![exemples/moins_large.jpg](align: :right, legend: "Alignée à droite")`



*Alignée à droite*

**EXEMPLE 11 •** Image plus petite en largeur alignée à droite et décalée vers la gauche.

`![exemples/moins_large.jpg](right: 2.cm, legend:"À 2 cm de la droite")`



*À 2 cm de la droite*

**EXEMPLE 12 •** Image plus petite que la largeur, agrandie avec « `width: "100%"` » pour tenir dans la largeur (déconseillé car cela « abîme » l'image).

`![exemples/moins_large_et_bas.jpg](width:"100%", legend:"Image agrandie à 100 % avec une grande légende qui doit tenir sur plusieurs lignes car une légende, par défaut, couvre la moitié de la page, mais on peut changer ce comportement.")`



*Image agrandie à 100 % avec une grande légende qui doit tenir sur plusieurs lignes car une légende, par défaut, couvre la moitié de la page, mais on peut changer ce comportement.*

**EXEMPLE 13 •** Image plus grande grâce à « `scale: 2.5` ».

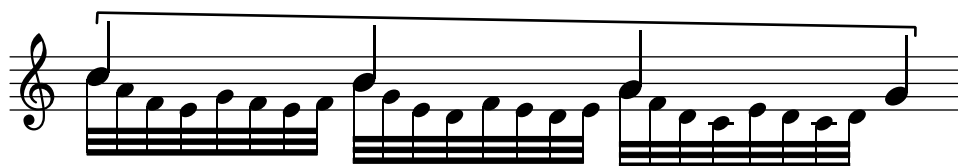
`![exemples/moins_large.jpg](scale: 2.5, legend:"Image avec scale de 2.5")`



Image avec scale de 2.5

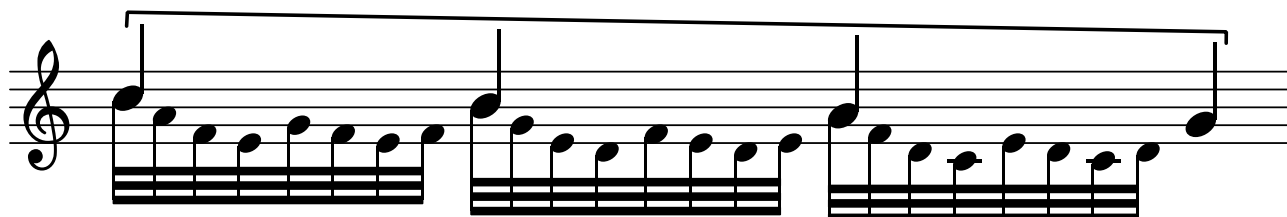
**EXEMPLE 14 •** Image SVG affichée sans aucune indication.

![exemples/image.svg]



**EXEMPLE 15 •** Image SVG grossie.

![exemples/image.svg](width: "100%")



**EXEMPLE 16 •** On ne peut pas (à ma connaissance) modifier disproportionnellement une image SVG dans Prawn même avec des valeurs explicites. Ci-dessous, la hauteur, bien que mise à 1000, reste proportionnée à la largeur.

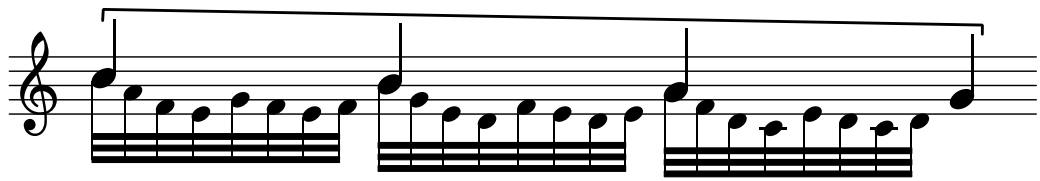
![exemples/image.svg](width: 200, height: 1000)



**EXEMPLE 17 •** Image SVG avec légende. Remarquez l'utilisation de `vadjust_legend` (« ajustement vertical de la légende ») qui permet ici de

l'éloigner de l'image.

```
![exemples/image.svg](width:"80%", legend:"Image SVG avec  
légende", vadjust_legend: 10)
```



*Image SVG avec légende*

**EXEMPLE 18** • Image avec de l'espace avant et de l'espace après définis par les propriétés `space_before` (« espace avant ») et `space_after` (« espace après »).  
![exemples/plus\_large.jpg](space\_before:180, space\_after: 100, legend:"La légende se place toujours bien.")

**Adagio**

The image shows a musical score for a piece titled "Adagio". It consists of five staves. The first staff is in treble clef and contains a melodic line with many slurs and ties, marked with *pp espressivo*. The second staff is also in treble clef and contains a line of notes with slurs, marked with *pp*. The third staff is in alto clef and contains a line of notes with slurs, marked with *pp*. The fourth staff is in bass clef and contains a line of notes with slurs, marked with *pp*. The fifth staff is in bass clef and contains a line of notes with slurs, marked with *pp* and *pizz. sempre*. The score is written in a key with three sharps (F#, C#, G#) and a 12/8 time signature. The tempo is marked "Adagio".

La légende se place toujours bien.

**EXEMPLE 19** • Image sans aucun redimensionnement, grâce à l'option `no_resize`. Cette image « mange » évidemment sur la marge extérieure...

```
![exemples/plus_large.jpg](no_resize:true, width:1000, left:0.1)
```



Fin des exemples d'images.

## Format de la légende

Comme nous l'avons mentionné, par défaut, la légende d'une image se formate avec la police par défaut, la taille diminuée de une unité, le style *italique* et la couleur noire.

Mais cet aspect naturel, qui passera pour tous les livres, peut être modifié pour tout le livre, dans les 12, ou de façon ponctuelle <sup>1</sup> grâce aux données avant le paragraphe définissant l'image.

<sup>1</sup> Ce genre de modification doit être absolument évitée pour garder une cohérence dans l'aspect général du livre.

---

*Si recipe.yaml ou recipe\_collection.yaml contient...*

---

---

```
book_format:
  images:
    legend:
      font: Courier
      size: 8
      color: "0FF0F0"
```

---

*Codes suivis de leur interprétation*

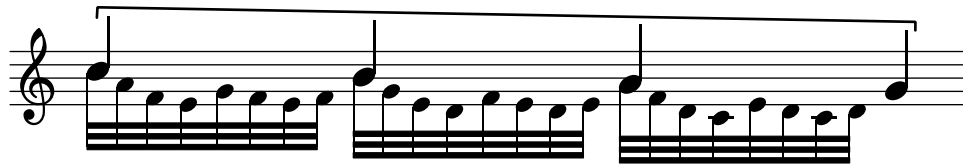
---

```
![exemples/image.svg](legend: "Une légende dans le style défini en recette")
```



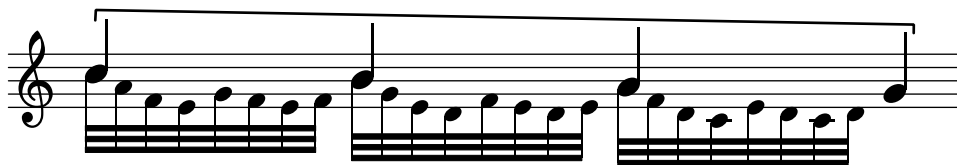


```
![exemples/image.svg](legend: "Légende avec couleur et  
ponctuelle", legend_color: [0,110,20,23], legend_size: 22)
```



*Légende avec couleur et  
taille ponctuelle*

```
![exemples/image.svg](legend: "Autre fonte, style, taille et  
couleur", legend_font: "Helvetica", legend_size: 14, legend_style:  
:regular, legend_color:[10,10,100,10], vadjust_legend: 15,  
space_before: 10)
```



Autre fonte, style, taille et couleur

Enim excepteur non ea officia sunt nostrud exercitation occaecat aliqua nostrud.

---





# Les Pages spéciales









# Le Mode Expert

## Description

Le *mode Expert* permet d'élargir 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

## 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 `### REF: recette_juste_titre ###`, 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"
```

---