





**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 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 30).

*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 30). 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>
# ...
```

## 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 12, dans la partie `text:` de `book_format:`.

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

---

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

*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.
- 

## Puce carrée noire

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

---

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

#</book\_format>

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

---

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

## Puce ronde

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

---

```
#<book_format>
book_format:
  text:
    puce:
      text: :bullet
      size: 20
      vadjust: 1
#</book_format>
```

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

---

- \* Ceci est une puce :bullet de 20 points, remontée de 1 point, avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

*Le livre final (document PDF) contiendra :*

---



- Ceci est une puce `:bullet` de 20 points, remontée de 1 point, avec les autres paramètres laissés intacts,
  - Le second item identique,
  - Le troisième.
- 

## Puce ronde noire

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

---

```
#<book_format>
book_format:
  text:
    puce:
      text: :black_bullet
      size: 30
      vadjust: 6
      left: 6.5mm
#</book_format>
```

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

---

- \* Ceci est une puce `:black_bullet` de 30 points, remontée de 6 points, avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

*Le livre final (document PDF) contiendra :*

---

- Ceci est une puce `:black_bullet` de 30 points, remontée de 6 points, avec les autres paramètres laissés intacts,
  - Le second item identique,
  - Le troisième.
- 

## Puce doigt tendu

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

---

```
#<book_format>
book_format:
  text:
    puce:
      text: :finger
      size: 20
      vadjust: 1
      left: 6.5mm
#</book_format>
```

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

---

- \* Ceci est une puce :finger de 20 points, remontée de 1 point, avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

*Le livre final (document PDF) contiendra :*

---

- ☞ Ceci est une puce :finger de 20 points, remontée de 1 point, avec les autres paramètres laissés intacts,
  - ☞ Le second item identique,
  - ☞ Le troisième.
- 

## Puce doigt tendu noir

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

---

```
#<book_format>
book_format:
  text:
    puce:
      text: :black_finger
      size: 20
      vadjust: 1
      left: 7mm
```

#</book\_format>

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

---

- \* Ceci est une puce :black\_finger de 20 points, remontée de 1 point, avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

*Le livre final (document PDF) contiendra :*

---

- ☛ Ceci est une puce :black\_finger de 20 points, remontée de 1 point, avec les autres paramètres laissés intacts,
  - ☛ Le second item identique,
  - ☛ Le troisième.
- 

## Puce gros tiret

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

---

```
#<book_format>
book_format:
  text:
    puce:
      text: :hyphen
      size: 100
      vadjust: 65
      left: 20mm
#</book_format>
```

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

---

- \* Ceci est une puce :hyphen de 100 points, remontée de 65 points, texte décalé de 20mm, avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

- Ceci est une puce :hyphen de 100 points, remontée de 65 points, texte décalé de 20mm, avec les autres paramètres laissés intacts,
  - Le second item identique,
  - Le troisième.
- 

## Puce personnalisées

Enfin, on peut utiliser des puces personnalisées partant d'une image que vous fournissez. Il suffit de donner son chemin d'accès soit relatif, par rapport au dossier du livre ou de la collection (recommandé), soit absolu. Dans l'exemple, la puce se trouve dans le dossier 'images' du livre qui, comme son nom l'indique, contient toutes les images que nous utilisons dans le manuel.

On peut aussi définir la propriété :height pour la hauteur de l'image. Si :width est défini (et non proportionnel), l'image sera déformée, sinon, la largeur sera proportionnellement adaptée à la hauteur.

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

---

```
#<book_format>
book_format:
  text:
    puce:
      text: 'images/custom_bullet.png'
      size: 14
      vadjust: -2
      left: 8mm
#</book_format>
```

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

---

- \* Ceci est une puce personnalisée de 14 points, remontée de -2 point, avec les autres paramètres laissés intacts,
- \* Le second item identique,
- \* Le troisième.

- ❏ Ceci est une puce personnalisée de 14 points, remontée de -2 point, 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 **### table\_ruby ###**.

Les formats (jpg, tiff, png, svg)

Rogner une image svg **### image\_rogner\_svg ###**.

{{TODO: Développer, montrer des exemples}}

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

---

```
---  
# ...
```

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

---

Le texte en exemple. Si 'texte' n'est pas défini, sera interprété aussi. Sinon sera mis en illustration et c'est 'texte' qui sera interprété comme texte du livre.

*Le livre final (document PDF) contiendra :*

---

Texte à interpréter, si 'sample\_texte' ne peut pas l'être.

---

## Notes de pages

Un des partis-pris fort de **Prawn-For-Book**, que certains regretteront sans doute, est de ne jamais utiliser de *notes de bas de page*. Malgré leur usage, nous pensons qu'elles offrent une mauvaise expérience à la lectrice ou au lecteur, et que souvent ce lecteur ou cette lectrice n'apprécie pas d'interrompre sa lecture pour consulter ces notes.

Nous avons donc pris l'option d'utiliser plutôt les « notes de page » qui se placent, en général, à la fin du paragraphe ou à la fin du bloc cohérent de paragraphes (vous pouvez les placer à l'endroit que vous voulez, mais elles seront toujours insérées dans le flux du texte, donc sur la page suivante si vous les placez trop loin).

Vous pouvez également décider de les reporter toutes à la fin de l'ouvrage. Mais alors, assurez-vous que ce ne soit pas des notes capitales dont l'ignorance nuierait à la compréhension du texte...

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

---

Ceci est un paragraphe avec une note numérotée<sup>^^</sup> de façon automatique, bien pratique par exemple pour les notes de fin d'ouvrage.

Ceci est un paragraphe avec une note numérotée<sup>^112</sup> explicitement.

<sup>^^</sup> Note de la note numérotée automatiquement.

<sup>^112</sup> Note de la note numéroté explicitement.

*Le livre final (document PDF) contiendra :*

---

Ceci est un paragraphe avec une note numérotée<sup>1</sup> de façon automatique, bien pratique par exemple pour les notes de fin d'ouvrage.

Ceci est un paragraphe avec une note numérotée<sup>112</sup> explicitement.

---

<sup>1</sup> Note de la note numérotée automatiquement.

<sup>112</sup> Note de la note numéroté explicitement.

---

Un paragraphe après les notes.

---

## Formatage des notes de pages

Comme tous les autres aspects d'un livre **Prawn-For-Book**, l'aspect des notes de page peut être réglé de façon très fine grâce à la recette du livre ou de la collection.

On peut définir la fonte, la taille et le style. On peut définir si les traits qui sont mis au-dessus et en dessous doit être appliqués (pour ne pas les imprimer, mettre `note_page_borders` à `false` ou `0`).

Par défaut, les notes sont alignées à gauche comme le texte, mais grâce à `note_page_left`, on peut définir de les mettre en retrait. La valeur est un nombre de points-pdf (points post-script).

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

---

```
---
...
book_format:
  text:
    notes_page:
      font: "Times-Roman"
      size: 15
      style: ""
      color: CCCCCC
      borders: 8
      border_color: 00FF00
      left: 40
```

*Le livre final (document PDF) contiendra :*

---

Ceci est une note automatique<sup>2</sup> avec une autre note auto-incrémentée<sup>3</sup> et une troisième note<sup>4</sup> pour voir.

On en a même une quatrième<sup>5</sup> sur un autre paragraphe.

---

<sup>2</sup> Explication de la note automatique

<sup>3</sup> Explication de la note auto-incrémentée (ou « note automatique »)

<sup>4</sup> Explication de la troisième note, qui tiendra elle sur trois lignes pour bien voir l'aspect de ces notes et voir que la ligne (le *border*) est conforme à ce que l'on a décidé, avec couleur et épaisseur.

<sup>5</sup> Bien sûr, les explications des notes auto-incrémentées doivent être dans l'ordre de leur apparition dans les paragraphes.

---

Un paragraphe à la suite du bloc de notes.

---

## Indication du contexte d'erreur

À titre préventif dans les méthodes personnalisées, *helpers* et autres modules, on peut indiquer le contexte qui devra être affiché en cas d'erreur.

Cela se fait en utilisant le code `PFBError.context = "Le contexte"`.

*Exemple dans un helper\*\**

---

```
# ruby
def monHelper(pdf, book)
  12.times do |i|
    # Indiquer le contexte
    PFBError.context = <<~EOC
      Dans la boucle de calcul et d'écriture du
      chiffre, avec i = #{i}
    EOC
    ecrire_ce_chiffre(i)
  end
  # Penser à "défaire" le contexte
  PFBError.context = nil
end
```

## Injection

**`book.inject(pdf, string, idx, self)`** est vraiment la formule magique pour ajouter du contenu au livre. L'avantage principale de cette méthode est d'analyser précisément le type de contenu —représenté ici pour `string`— et de le traiter conformément à son type. Par exemple :

- si `string` est `![images/mon.svg]`, alors ce sera une image qui sera traitée,
- si `string` est `### Mon beau titre` alors c'est un titre qui sera inséré,
- si `string` est `(( new_page ))` alors on passera à la nouvelle page <sup>1</sup>.

---

<sup>1</sup> Bien sûr, ici, dans le programme, on pourrait utiliser `pdf.start_new_page`, mais l'idée derrière l'utilisation de `book.inject(...)` est de pouvoir utiliser le même code que dans le livre. Inutile d'apprendre une nouvelle langue ou de fouiller dans le code du programme pour savoir comment exécuter telle ou telle action.

---



**idx** correspond à l'index du paragraphe dans la source injectée, il n'a de valeur que pour le débogage. Dans le programme, il correspond par exemple au numéro de ligne dans le fichier. On pourra l'utiliser comme l'on veut.

**self**, dans le programme, correspond à l'instance du fichier de texte (`Prawn4book::InputTextFile`). On peut le définir comme tel si le code injecté vient d'un fichier (même si, dans ce cas, il vaudrait mieux utiliser tout simplement la `string`: ( `include mon/fichier.md` ) ).  
{{TODO: Développer encore}}







# 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 12, que ce soit pour un livre unique (cf. page 12) ou pour une collection (cf. page 12).

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"
```

---