

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

Généralités

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. ci-dessous).

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 page 86).

Vous pouvez voir ci-dessous un extrait du fichier recette de ce manuel (qui a bien sûr été produit par **Prawn-For-Book**).

Extrait de fichier recette

```
---
#<book_data>
book_data:
  title:      "Manuel de Prawn-for-book"
```

```

author:    "Philippe PERRET"

version:   1.3
isbn:      null
# ...
#</book_data>
#<book_format>
book_format:
  book:
    width:  '210cm'
    height: '297cm'
    orientation: portrait
    format: :pdf
# ...

```

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 86).

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>
# ...

```

Le Texte du livre

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.

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 .

Les Pages spéciales

Les Types de pages spéciales

Les pages considérées comme *spéciales* dans **Prawn-For-Book** sont les suivantes :

- la **page de garde**. C'est la page vierge, après la couverture,
- la **page de faux-titre**. Page contenant juste le titre, l'auteur,
- la **page de titre**. Page contenant toutes les informations de la couverture, le titre, le sous-titre éventuel, les auteurs, l'éditeur et son logo,
- la **page d'informations**. Page, à la fin du livre, contenant une sorte de *générique* du livre, avec la liste de tous les intervenants (metteur en page, graphiste, concepteur de la couverture, etc.), les informations sur la maison d'édition, la date de parution, l'ISBN et tout autre renseignement utile.

Nous allons aborder chaque page séparément. Ce que l'on peut retenir pour le moment, c'est que l'on détermine facilement dans la recette la présence ou non de ces pages dans la rubrique `inserted_pages` : (ci-dessous, ce sont les valeurs par défaut) :

Dans recipe.yaml

```
---
...
inserted_pages:
  page_de_garde:  true
  faux_titre:     false
  page_de_titre:  true
  page_infos:     false
```

la Table des matières

La description

la Page des informations

La description

Entêtes et Pieds de page

Description

Comme pour les autres éléments, on pourra laisser les **entêtes** et **pieds de page** par défaut, ce qui signifiera n'afficher que les numéros de pages — sur les pages adéquates —, ou au contraire on pourra définir des entêtes et pieds de page complexes et adaptés au contenu pour une navigation optimum.

Comme pour les autres éléments de **Prawn-For-Book**, les entêtes et pieds de page par défaut sont conçus pour être directement « professionnels ». C'est-à-dire que la numérotation est intelligente, elle ne numérote pas bêtement toutes les pages de la première à la dernière. Seules sont numérotées les pages qui le sont dans un livre imprimé. Sont soigneusement évitées les pages vides, les pages de titre ou les pages spéciales comme la Table des matières (page 30) ou la Page des informations (page 30).

Les pages suivantes vont définir les différents entête et pieds de page que l'on peut définir, en présentant dans la page le code utilisé et le résultat dans les entêtes et/ou les pieds de page.

Lexique

Comme pour les autres parties, nous définissons ici le lexique des termes qui seront rigoureusement utilisés dans cette partie.

- **Entête**. C'est la partie, en haut de page, au-dessus du texte principal de la page, qui contient le plus souvent le titre courant, qui permet de naviguer plus facilement dans les chapitres du livre.
- **Header**. *Entête*, en anglais.
- **Pied de page**. C'est la partie, en bas de page, sous le texte principal de la page, qui contient le plus souvent le *numéro de page*.
- **Footer**. *Pied de page*, anglais.
- **Portion**. Nous appelons « portion » l'un des 6 tiers de page qui constituent une *entête* ou un *pied de page*. Il y en a 6 parce qu'on travaille ici en double page. On trouve la *portion gauche*, la *portion droite* et la *portion centrale* de la page gauche et les mêmes portions sur la page droite.
- **Disposition**. Une *disposition* décrit le contenu des pieds et page et entête d'une ou plusieurs pages. Cela revient à définir les 6 portions de l'entête et les 6 portions du pied de page.

Trois portions de page

Un *entête* ou un *pied de page* est un espace de page qui contient trois portions au maximum, trois tiers de page, une portion gauche, une portion droite et une portion centrale. On répartit les éléments dans ces trois portions. Les trois portions peuvent être différentes entre page gauche et page droite.

On peut ne définir qu'une seule de ces trois portions, ou deux seulement.

Définition des entêtes et pieds de page

Les *entêtes* et *pieds de page*, comme pour le reste, se définissent dans les Recettes (page 21) du livre ou de la collection, dans une section qui s'appelle, on ne s'en étonnera pas : `headers_footers`.

On peut définir autant de *dispositions* que l'on veut, même s'il est conseillé, toujours, de rester le plus sobre possible. On peut se contenter, pour un résultat optimum, d'une *disposition* pour le corps du livre, son contenu principal, et une *disposition* pour les annexes si elles existent.

La propriété principale de la disposition est la valeur de son header (son « entête ») et son footer (« pied de page »). Par défaut, les valeurs sont :

- header : "| x | x | x || x | x | x |"
- footer : "| -NUM | x | x || x | x | NUM- |"

On voit clairement la reliure représentée par | | et de chaque côté les trois portions.

Les x signifient qu'on ne met rien dans ces portions.

Le NUM indique l'endroit où l'on va marquer le numéro de la page. Voir ci-dessous les *éléments* qui peuvent composer une disposition.

Le tiret (moins) qui précède ou suit NUM indique l'alignement dans la portion. Le trait avant (-NUM) signifie que le numéro sera aligné à gauche, le trait après (NUM-) signifie que le numéro sera aligné à droite. Pour le centrer, on aurait rien mis car l'élément est aligné au centre par défaut.

Mais en fait, la vraie disposition est plus simple, car il est inutile de définir toutes les portions si elles sont vides. La vraie disposition par défaut dans **Prawn-For-Book** est :

- header : "| x || x |"
- footer : "| -NUM || NUM- |"

Les éléments

Les éléments qui peuvent être utilisés dans les entêtes et les pieds de page sont illimités en mode expert (### REF: expert_header_footer ###). Pour le commun des mortels, ils se limitent — ce qui est déjà largement suffisant — aux titres courants jusqu'au niveau 5 (en majuscules, en minuscules, ou format titre), au numéro de la page ainsi qu'au nombre total de page.

- **NUM** pour le numéro de page,
- **TIT1** pour le titre de niveau 1 courant, en majuscules,
- **tit1** pour le titre de niveau 1 courant, en minuscules,
- **Tit1** pour le titre de niveau 1 courant, en format titre,
- **TIT2** (et **tit2**, **Tit2**) pour le titre de niveau 2 courant,
- **TIT3** (et **tit3**, **Tit3**) pour le titre de niveau 3 courant,
- **TIT4** (et **tit4**, **Tit4**) pour le titre de niveau 4 courant,
- **TIT5** (et **tit5**, **Tit5**) pour le titre de niveau 5 courant,
- **TOT** pour le nombre total de pages,

ENTÊTES ET PIEDS DE PAGE

- `#{code}` pour un code ruby quelconque à évaluer à la volée, par exemple un numéro de version ou des auteurs,

Ajustement

Noter bien que l'*ajustement vertical* effectué avec `header_vadjust` et `footer_vadjust` fonctionne différemment. Dans les deux cas, il définit l'éloignement *par rapport au texte principal de la page*. Ainsi, pour `header_vadjust` (l'entête), une valeur positive fera monter les portions alors que pour `footer_vadjust` (le pied de page), une valeur positive les fera descendre.

Fontes

Bien qu'on puisse définir les polices très précisément pour chaque élément, il est conseillé de ne pas trop les différencier. La sobriété est toujours bonne conseillère, en matière de mise en page.

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

```
---
# ...
# Définit le début de la définition des entêtes et pieds de
# page
headers_footers:
  # Fonte générale pour tous les pieds et page et entêtes
  # Si non défini, c'est la fonte par défaut qui sera utilisée
  font: "<name>/<style>/<size>/<color>"
  # Définition des dispositions
  dispositions:
    # Pour supprimer la disposition par défaut, il suffit
    # de faire :
    default: null
    # Définition d'une disposition
    ma_disposition:
      name: "Nom humain juste pour mémoire"
      # Fonte particulière pour cette disposition en
      # particulier :
      font: "<font name>/<font style>/<font size>/<font color>"
      pages: <première page>-<dernière page>
      header: | x | x | x || x | x | x |
      footer: | x | x | x || x | x | x |
      header_vadjust: <ajustement vertical entête>
      header_hadjust: <ajustement horizontal entete>
```

v4.452

footer_vadjust: <ajustement vertical pied>

footer_vadjust: <ajustement vertical pied>

- name: "Autre disposition"

font: ...

etc.

ENTÊTES ET PIEDS DE PAGE

Exemple complexe

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

```
---
headers_footers:
  dispositions:
    exemple1:
      name: "Exemple headers/footers complexe pour manuel"
      font: "Courier/regular/12/FF0000"
      pages: "36-41"
      header: "| -TIT3      || -v4.452 | TIT4- |"
      footer: "| x | NUM || NUM/TOT | PhP |"
      header_font: "Reenie/normal/20/009900"
```

Si texte.pfb.md contient...

((new_page))

Une première page pour voir l'entête et le pied de page définis (ça se poursuit sur l'autre page).

((new_page))

Une seconde page qui doit permettre aussi de voir l'entête et le pied de page sur les deux pages, car on ne sait pas, **a priori**, où va se retrouver cette page puisque le manuel est autoproduit par *****Prawn-For-Book***** lui-même !

Donc cette page que vous lisez peut se trouver en "fausse page", c'est-à-dire à gauche du livre, en page paire, aussi bien qu'en "belle page", c'est-à-dire à droite du livre, en page impaire.

Ainsi, dans tous les cas vous devriez pouvoir obtenir une nouvelle page contenant l'entête et le pied de page voulu.

Donc, à la différence des pages par défaut, on devrait trouver ici, à gauche, le numéro de page au milieu, alors qu'il est à gauche dans le pied de page droit, avec le nombre total de pages dans le manuel.

On doit trouver en à gauche de la page gauche le titre principal courant, donc le chapitre, qui s'intitule "Entêtes et Pieds de pages".

Et on doit trouver sur la page droite, en entête, le numéro de version "4.452" à gauche et au milieu le sous-titre du chapitre donc "Exemple complexe".

Le livre final (document PDF) contiendra :

v4.452

Une première page pour voir l'entête et le pied de page définis (ça se poursuit sur l'autre page).

ENTÊTES ET PIEDS DE PAGE

Une seconde page qui doit permettre aussi de voir l'entête et le pied de page sur les deux pages, car on ne sait pas, *a priori*, où va se retrouver cette page puisque le manuel est autoproduit par **Prawn-For-Book** lui-même !

Donc cette page que vous lisez peut se trouver en "fausse page", c'est-à-dire à gauche du livre, en page paire, aussi bien qu'en "belle page", c'est-à-dire à droite du livre, en page impaire.

Ainsi, dans tous les cas vous devriez pouvoir obtenir une nouvelle page contenant l'entête et le pied de page voulu.

Donc, à la différence des pages par défaut, on devrait trouver ici, à gauche, le numéro de page au milieu, alors qu'il est à gauche dans le pied de page droit, avec le nombre total de pages dans le manuel.

On doit trouver en à gauche de la page gauche le titre principal courant, donc le chapitre, qui s'intitule « Entêtes et Pieds de pages ».

Et on doit trouver sur la page droite, en entête, le numéro de version « 4.452 » à gauche et au milieu le sous-titre du chapitre donc « Exemple complexe ».

Le Texte en détail

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 page 21 ou la page 22, 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_loange
      size: 14
      vadjust: -2
#</book_format>
```

Si texte.pfb.md contient...

- * Ceci est une puce :black_loange 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_loange 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.

Le livre final (document PDF) contiendra :

■ 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.

Le livre final (document PDF) contiendra :

-  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 `### REF: table_ruby ###`.

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

La suite va présenter toutes les propriétés utilisables avec les images, mais les pages suivantes reprendront toutes ces propriétés pour en donner des exemples concrets et illustrés.

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 page 65).

Pour un placement optimum, on peut ajuster la position de l'image verticalement à l'aide de `valign` (« 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 ramenée par défaut à la largeur de la page du livre si elle est supérieure, et la hauteur de la page si sa hauteur l'excède. Mais si l'on sait ce que l'on veut obtenir et qu'on désire tout contrôler soi-même, on peut ajouter l'option `no_resize: true` qui empêchera le redimensionnement naturel (à vos risques et péril comme dans l'exemple 19 à la page page 66).

Légende de l'image

Une légende peut être définie en renseignant 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, en noir et en italique.

Mais on peut régler tous ces aspects :

- soit pour tout le livre (ou toute la collection) dans le fichier recette, dans `book_format > images > legend` (avec les propriétés `font`, `size`, `style` et `color`) — comme ce sera illustré dans la partie page 66.
- soit de façon ponctuelle pour une seule image/légende en définissant dans les données de l'image les propriétés `legend_font` (nom de la police), `legend_style` (style de la police, par exemple `:regular`), `legend_size` (taille de la police) et `legend_color` (couleur 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

Pour savoir comment rogner une image svg, voir page 86.

Exemples

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

Aspect des images en fonction du code

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^o 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 placée sur la page suivante")
```

NOTTURNO
für das Pianoforte
von
RIEDRICH CHOPIN.
Nachgelassenes Werk. (Op. 72. N^o 1)

The image displays a musical score for Frédéric Chopin's Nocturne Op. 72, No. 1. It consists of six systems of music, each with a treble and bass staff. The key signature is one sharp (F#), and the time signature is common time (C). The score includes various musical notations such as slurs, ornaments (marked with a stylized 'w' and an asterisk), and dynamic markings: *cresc.*, *dim.*, and *poco a poco cresc.*. The bottom of the page is marked with the number 'C. XIII. 31.'

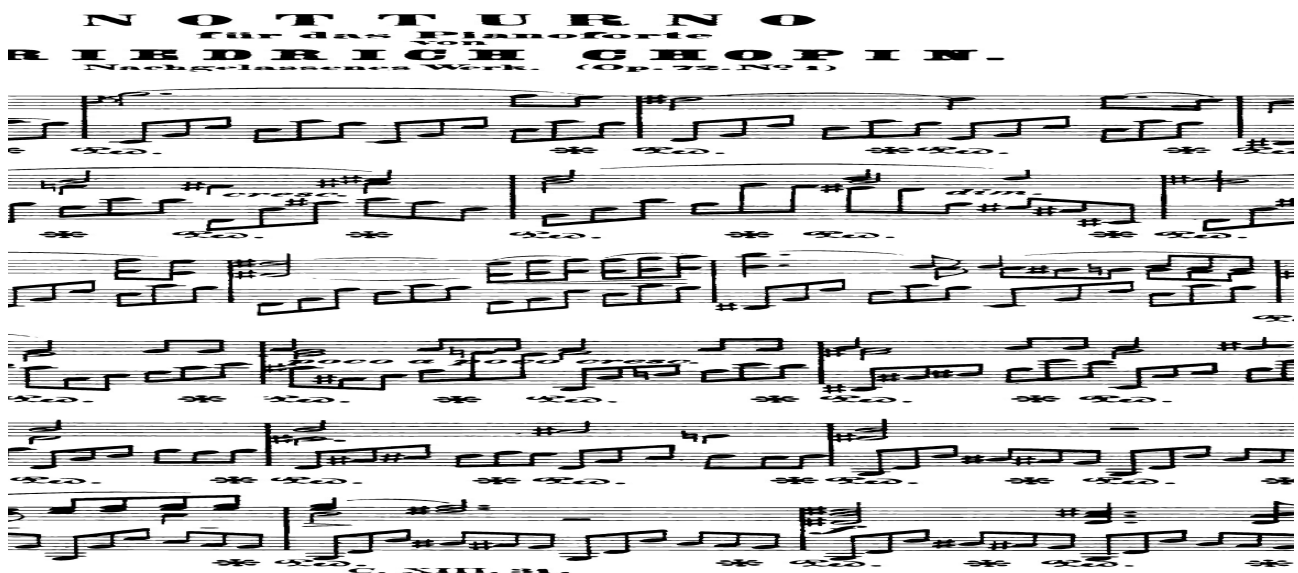
Image plus haute réduite et
placée sur la page suivante

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` ».

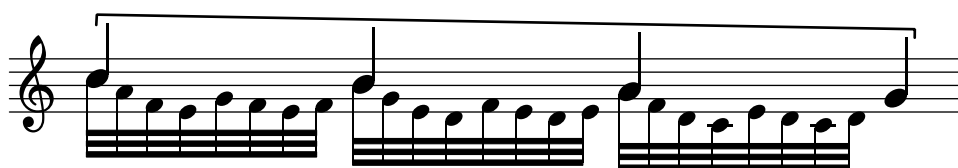
`![exemples/moins_large.jpg](scale: 2.5, legend:"Image avec scale de 2.5")`



Image avec scale de 2.5

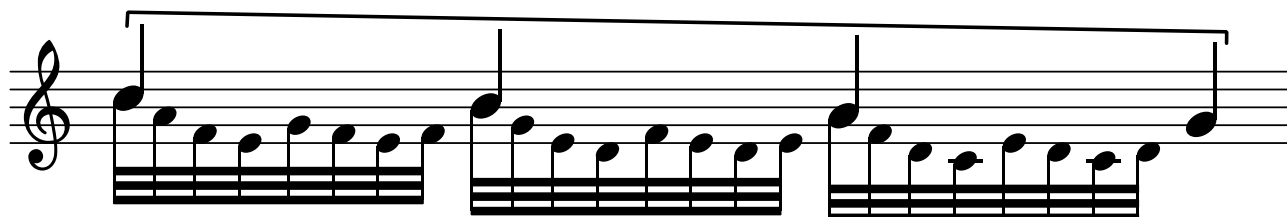
EXEMPLE 14 • Image SVG affichée sans aucune indication.

`![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)
```

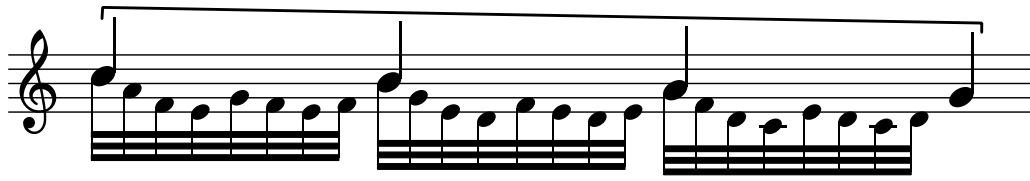


Image SVG avec légende

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.")
```



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 page 21, ou de façon ponctuelle¹ grâce aux données avant le paragraphe définissant l'image.

¹ 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...

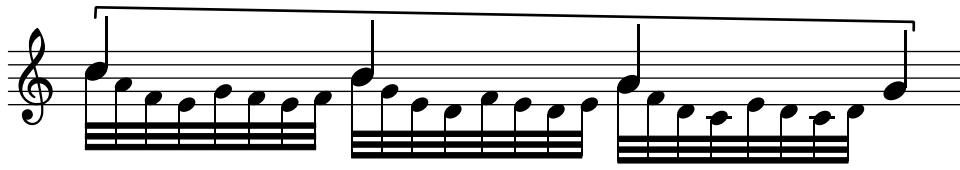
```
---
book_format:
  images:
    legend:
      font: Courier
      size: 8
      color: "0FF0F0"
```

Codes suivis de leur interprétation

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

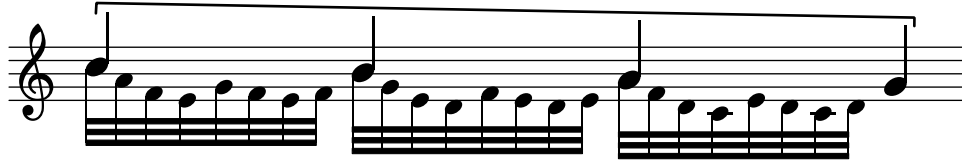


```
![exemples/image.svg](legend: "Légende avec couleur et taille  
ponctuelle", legend_color: [0,110,20,23], legend_size: 22)
```



*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 références croisées

Lexique

Pour bien comprendre de quoi nous parlons, nous nommons les choses de façon stricte :

- la **cible** (ou **cible de la référence**), c'est l'endroit du livre auquel nous faisons référence (il se marque par `<-(id_ref)`),
- l'**appel** (ou **appel de la référence**), c'est la marque qui permet, dans le code-texte du livre, de faire référence à un autre endroit du livre (il ressemble à `\->(id_ref)`),
- la **marque** (ou **marque de la référence**), c'est le texte précis qui remplacera l'*appel* et qui sera donc lu dans le livre imprimé,
- l'**endroit** (ou **endroit de la référence**), c'est précisément la page, le paragraphe ou la page et le paragraphe où se trouve la *cible*.

Définition

Les *références croisées* permettent de faire référence, dans un livre, à une autre partie du livre ou même, avec **Prawn-For-Book**, à une partie d'un autre livre. Typiquement, c'est le « cf. page 12 » qu'on trouve dans un ouvrage.

Les *références croisées* fonctionnent à partir d'une *cible*, la partie du livre à rejoindre, et d'un *lien* vers cette cible.

Format de la marque

Dans **Prawn-For-Book**, la *marque de la référence* peut s'imprimer de différentes manières. Par défaut, elle pourra avoir trois aspects qui dépendront du type de numérotation défini, à savoir :

- le numéro de page (par exemple page 12,
- le numéro de paragraphe (par exemple § 123),
- le format *hybride* avec numéro de page et de paragraphe (par exemple p 12 § 123).

Il est cependant possible de définir de façon précise et différente la *marque* en utilisant un *appel* de la forme :

```
\->(id_ref|"le texte de la marque>")
```

Par exemple :

C'est un texte avec une `\->(mon_chapitre|référence croisée)` qui sera titrée.

Produira :

C'est un texte avec une référence croisée (page 12) qui sera titrée.

Dans cette formule, l'**endroit** sera toujours spécifié entre parenthèse après le texte donné. On peut cependant aller plus loin encore et définir très précisément l'aspect d'une référence particulière :

(utiliser la marque « `_ref_` » pour indiquer où doit se marquer « page 12 » ou « p. 123 § 12 »)

`\->(id_ref|(_ref_) le texte>)`

Par exemple :

C'est un texte avec une `\->(mon_chapitre|(_ref_) référence croisée)` qui sera personnalisée.

Produira :

C'est un texte avec une (page 12) référence croisée qui sera personnalisée.

Mais on peut aussi le définir de façon très précise avec `_page_` et `_paragraph_`

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

J'ai dans cette phrase une cible `MA_CIBLE` qui sera appelée à la page suivante<-(exemple_cible).

((`new_page`))

La cible `MA_CIBLE` se trouve à la page `->(exemple_cible)`.

Sur ce paragraphe je fais référence à un passage qui se trouve `->(ex_autre_cible|sur la _ref_ du livre)` avec un appel personnalisé.

Le livre final (document PDF) contiendra :

J'ai dans cette phrase une cible `MA_CIBLE` qui sera appelée à la page suivante.

La cible MA_CIBLE se trouve à la page page 71.

Ici se marque une référence (cible) à AUTRE_CIBLE pour montrer, sur la page suivante, comment sera présentée la marque personnalisée.

Sur ce paragraphe je fais référence à un passage qui se trouve sur la page 72 du livre avec un appel personnalisé.

Marque personnalisée

Dans la marque personnalisée, on peut utiliser `_ref_` pour indiquer où doit se marquer la marque de référence.

_ Une référence à `->(exemple_cible|une _ref_ deux)` pour voir.

=> Une référence à une page 71 deux pour voir.

On peut utiliser `_page_` pour le numéro de page (et seulement le numéro de page).

_ Une référence à `->(exemple_cible| la page bonne _page_)` pour avoir le numéro.

=> Une référence à la page bonne 71 pour avoir le numéro.

Enfin, on peut utiliser `_paragraph_` (sans « e ») pour le numéro du paragraphe ².

_ Une référence `->(exemple_cible|au paragraphe _paragraph_ de la page _page_ \(_ref_\))`. ³

=> Une référence au paragraphe 446 de la page 71 (page 71).

² Mais dans ce cas, notez que suivant le format de pagination, ce numéro pourra être le numéro absolu ou le numéro par rapport à la double-page (cf. `### REF: pagination ###`)

³ Noter que les parenthèses, à l'intérieur de la marque de référence, ont été « échappées » (précédées du signe « \ ») afin de ne pas être prises pour la fin de l'appel de référence.

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^{^^} de façon automatique, bien pratique par exemple pour les notes de fin d'ouvrage.

Ceci est un paragraphe avec une note numérotée^{^112} explicitement.

^{^^} Note de la note numérotée automatiquement.

^{^112} Note de la note numéroté explicitement.

Le livre final (document PDF) contiendra :

Ceci est un paragraphe avec une note numérotée⁴ de façon automatique, bien pratique par exemple pour les notes de fin d'ouvrage.

Ceci est un paragraphe avec une note numérotée¹¹² explicitement.

⁴ Note de la note numérotée automatiquement.

¹¹² 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 ⁵ avec une autre note auto-incrémentée ⁶ et une troisième note ⁷ pour voir.

On en a même une quatrième ⁸ sur un autre paragraphe.

⁵ Explication de la note automatique

⁶ Explication de la note auto-incrémentée (ou « note automatique »)

⁷ 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.

⁸ Bien sûr, les explications des notes auto-incrémentées doivent être dans l'ordre de leur apparition dans les paragraphes.

Id in deserunt deserunt ut incididunt consequat occaecat est.

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**.

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 par `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 ¹.

¹ 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 correspond dans le programme 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))`). Si elle n'est pas fournie, elle sera égale à `"user_metho"`.

{{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 `### REF: recette_juste_titre ###`, que ce soit pour un livre unique (cf. page page 21) ou pour une collection (cf. page page 22). 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"
```

Rogner une image SVG

Il est fort possible qu'en produisant une image SVG, et en l'insérant dans le livre, elle laisse voir trop de blanc autour d'elle, comme dans le premier exemple donné ci-après. Pour palier ce problème, il faut « rogner » cette image SVG. Mais *rogner* une image SVG ne se fait pas aussi facilement qu'avec une image d'un format non vectoriel (JPG, PNG, etc.). Il faut pour ce faire utiliser, après avoir chargé la commande `inkscape` dans votre ordinateur, le code suivant :

```
inkscape -l -D -o image-rogned.svg image.svg
```

Le livre final (document PDF) contiendra :

Image non rognée :



La même image rognée :



Amet exercitation ut dolore in in nulla adipisicing laborum.
