











# Manuel de Prawn-for-book

*(le manuel autoproduit de l'application  
de mise en forme professionnelle)*

Philippe Perret





AVANT-PROPOS .....	11
FONCTIONNALITÉS .....	15
Généralités .....	17
Les Forces de <i>Prawn-For-Book</i> .....	17
Les deux fichiers de base .....	17
Les Recettes .....	19
Recette du livre .....	19
Recette de la collection .....	19
Définition des marges .....	21
Définition de la fonte par défaut .....	23
Éditeur / Maison d'édition .....	23
Les Images .....	24
Insérer une image .....	25
Les Images flottantes .....	39
Bibliographies .....	51
Introduction .....	51
Liste des costumes .....	57
Le Mode Expert .....	59
Description .....	59
Évaluation du code ruby .....	59
Bibliographies en mode expert .....	61
Le format YAML de la recette .....	61
Films cités .....	62
Articles de presse .....	62







# Avant-propos

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).

Ce manuel de 68 pages est auto-produit par « **Prawn-for-book** », c'est-à-dire qu'il est construit de façon *programmative* par l'application elle-même. Il en utilise toutes les fonctionnalités puisqu'il génère de façon automatisé les exemples et notamment les *helpers* de mise en forme, les références croisées ou les bibliographies. En ce sens, ce manuel sert donc aussi de test complet de l'application puisqu'une fonctionnalité qui ne fonctionnerait pas ici ne fonctionnerait pas non plus dans le livre produit.

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 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).

## Signe distinctif

Le *signe distinctif* de **Prawn-For-Book**, utilisé pour tous les codes, est la double parenthèses :

```
(( . . . ))
```

Dès qu'un code doit être ajouté au texte, c'est souvent ce signe qui est utilisé. Vous pouvez le voir pour ce propre texte, où des `(( line ))` ont été utilisés pour sauter des lignes et un `(( {align: :center} ))` a été employé pour mettre le paragraphe suivant à la ligne.

Noter qu'il est impératif de laisser une espace à l'intérieur des parenthèses, de chaque côté. Dans le cas contraire, le code ne serait pas vu.

Si vous utilisez `### REF: annexe_package_sublime_text ###`, vous verrez le code se mettre en couleur, ce qui vous assurera que vous avez la bonne expression.

---

*Description précédente obtenue à l'aide de :*

Le *\*signe distinctif\** de **\*\*\*Prawn-For-Book\*\*\***, utilisé pour tous les codes, est la double parenthèses :

```
(( line ))
```

```
(( . . . ))
```

```
(( line ))
```

Dès qu'un code doit être ajouté au texte, c'est souvent ce signe qui est utilisé. Vous pouvez le voir pour ce propre texte, où des `(( line ))` ont été utilisés pour sauter des lignes et un `(( {align: :center} ))` a été employé pour mettre le paragraphe suivant à la ligne.

Noter qu'il est impératif de laisser une espace à l'intérieur des parenthèses, de chaque côté. Dans le cas contraire, le code ne serait pas vu.

Si vous utilisez `### REF: annexe_package_sublime_text ###`, vous verrez le code se mettre en couleur, ce qui vous assurera que vous avez la bonne expression.

---

# 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 61).

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 page 61).

Vous pouvez trouver ci-dessous les données propres à une collection.

### *Si le fichier `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 des marges

Des marges par défaut sont proposées, mais vous pouvez tout à fait définir celles que vous voulez très précisément, dans la recette du livre ou de la collection.

La seule chose à comprendre, par rapport à aux documents dont vous avez l'habitude, c'est qu'ici les pages sont doubles, en vis-à-vis, et définissent donc :

- une marge haute et une marge basse (traditionnelle),
- une marge intérieure, qui comme son nom l'indique est à l'intérieur du livre, contre le dos,
- une marge extérieure, qui comme son nom l'indique est tournée vers l'extérieure du livre, vers la tranche (attention, la tranche du livre, ça n'est pas la partie qui reprend le titre du livre et son auteur, qu'on appelle le « dos »).

Ces marges sont donc définies par les propriétés `top` (haut) `bot` (pour « bottom », bas), `ext` (pour « extérieur ») et `int` (pour « intérieur »).

Traditionnellement, la marge intérieure est plus large que la marge extérieure, car une bonne partie de cette marge est prise dans la reliure.

De la même manière, la marge basse est plus large que la marge haute car elle contient le numéro de page. Il peut cependant arriver que la marge haute contienne un entête.

Pour régler parfaitement les marges, vous pouvez soit ajouter l'option `-display_margins` à la commande `pfb build` qui construit le livre, soit mettre le `show_margins` de la recette à `true`, comme nous l'avons fait ci-dessous.

Dans l'exemple ci-dessous nous avons volontaire *pousser* les valeurs pour rendre bien visibles les changements.

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

---

```
book_format:
  page:
    margins:
      top: 60mm
      bot: 70mm
      ext: 2mm
      int: 30mm
    show_margins: true
```

*Le livre final (document PDF) contiendra :*

---

Pour cette page, où les marges sont visibles, on illustre des marges de page à 60mm en haut, 70mm en bas, 30mm à l'intérieur et 2mm à l'extérieur.

Vous remarquerez donc une immense marge en bas, une grande marge en haut, une marge externe toute petite (*ce qui ne serait pas du tout bon pour une impression de livre*) et une marge intérieure moyenne.

Remarquez aussi que le texte est automatiquement justifié, il s'aligne parfaitement sur la marge gauche et la marge droite, ce qui donne un rendu impeccable.

---

## 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 le fichier `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 .

---

## Éditeur / Maison d'édition

L'éditeur du livre (« publisher » en anglais) ou la maison d'édition se définissent dans la propriété `publisher` de la recette du livre ou de la collection.

On peut définir toutes les données utiles

{{TODO: Poursuivre cette fonctionnalités}}

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

---

```
---
publisher:
  name: "<Éditeur / Maison d'édition>"
  adresse: "<Numéro rueCode Ville>"
  contact: "<contact@chez.lui>"
  url: "https://<url/site>"
  logo_path: "chemin/vers/logo"
```

---

# Les Images



## 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 37).

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 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 38).

## 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 48.
- 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 `### REF: images_rogner_svg ###`.

## 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<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 placée sur la page suivante")
```

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

*cresc.* *dim.*

*poco a poco cresc.*

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.

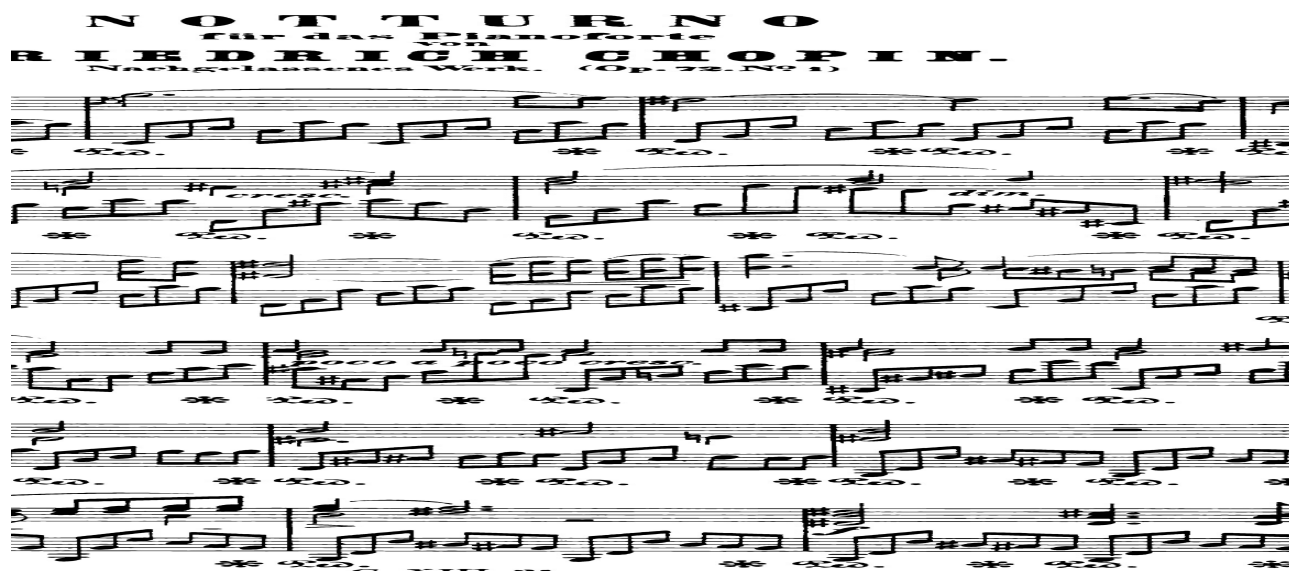
![exemples/plus\_haute.png](height:"30%", legend:"Trop haute avec height à 30%")



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`).

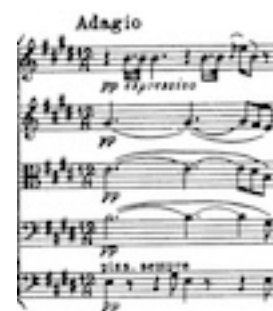
`![exemples/moins_large.jpg](left: "2cm", vadjust: 1, legend: "Image à gauche décalée vers la droite", vadjust_legend: 10)`



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.")

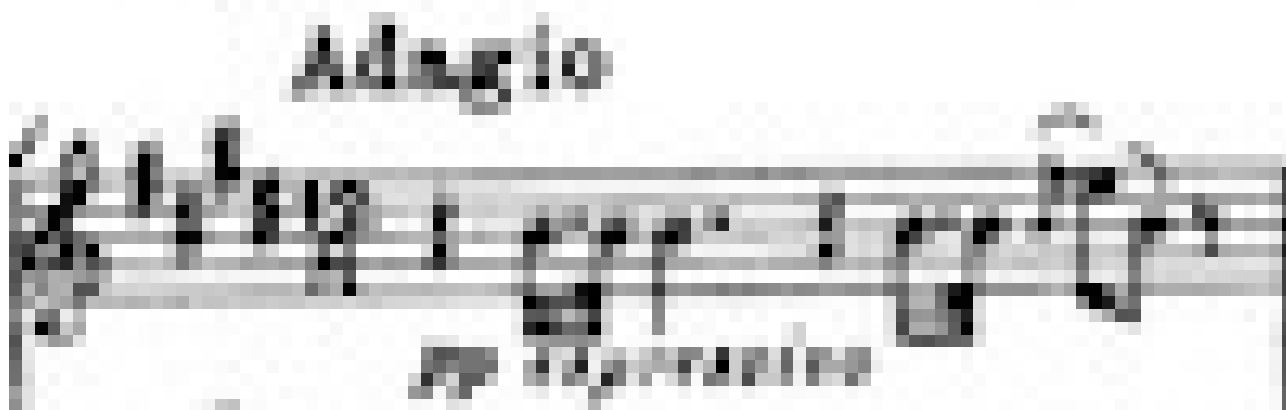


Image agrandie à 100 % avec une grande légende qui doit tenir sur plusieurs lignes car une légende, par défaut, couvre la moitié de la page, mais on peut changer ce comportement.

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

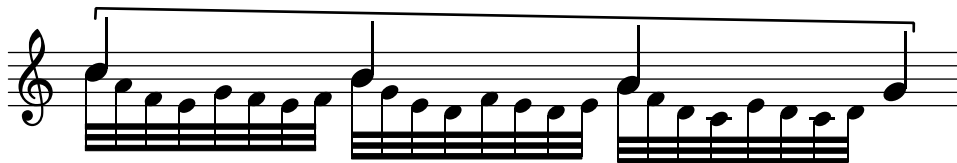
![exemples/moins\_large.jpg](scale: 2.5, legend:"Image avec scale de 2.5")



Image avec scale de 2.5

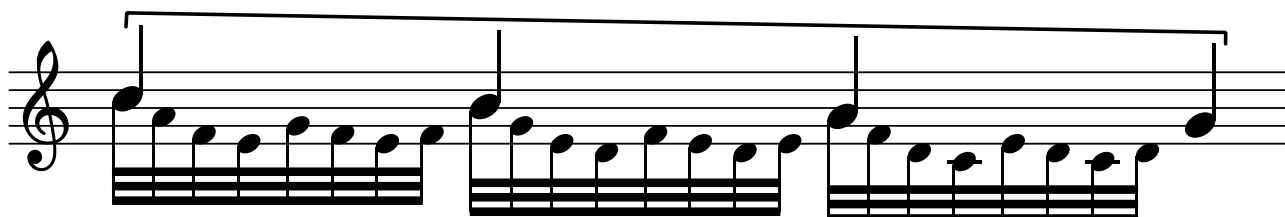
**EXEMPLE 14** • Image SVG affichée sans aucune indication.

`![exemples/image.svg]`



**EXEMPLE 15** • Image SVG grossie.

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



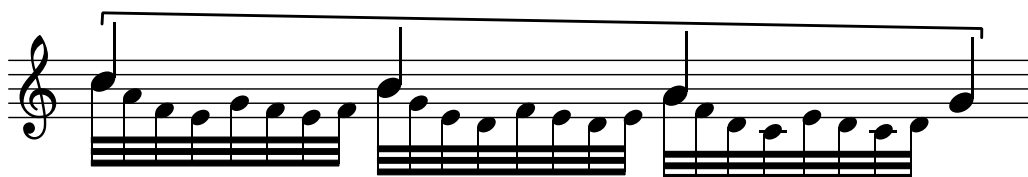
**EXEMPLE 16** • On ne peut pas (à ma connaissance) modifier disproportionnellement une image SVG dans Prawn même avec des valeurs explicites. Ci-dessous, la hauteur, bien que mise à 1000, reste proportionnée à la largeur.

`![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.

## Les Images flottantes

Cette partie décrit comment obtenir une image flottante dans le texte. Pour définir une image flottante, il faut définir deux choses indispensables :

- Une image définissant la propriété `float` (cf. ci-dessous),
- Un texte, *avant cette image*, commençant par « ! ».

Les paramètres pour gérer l'image flottante sont les suivants :

- **float** | Définit que l'image est flottante. Ce paramètre peut avoir la valeur `:left` (« gauche » en anglais) pour « flottante à gauche » (l'image sera donc à gauche du texte) ou `:right` (« droite » en anglais) pour « flottante à droite » (l'image sera donc à droite du texte).
- **lines\_before** | Définit le nombre de lignes qui vont passer au-dessus de l'image. Par défaut à 0, la première ligne de texte est alignée au haut de l'image (en fonction de ligne de référence du texte).
- **margin\_top** | Définit la hauteur à laisser au-dessus de l'image, qu'il y ait ou non des lignes au-dessus. En ne jouant que sur ce paramètre, on ne peut pas faire passer de lignes au-dessus de l'image (utiliser `lines_before` pour ça).
- **vadjust** | Lorsque le `margin_top` ne permet pas de positionner l'image verticalement, on peut se servir de cette propriété *traditionnelle*.
- **margin\_left** | Marge à gauche. Quand l'image est flottante à gauche, détermine la distance de l'image avec la marge gauche. Quand l'image est flottante à droite, détermine la distance de l'image avec le texte à gauche. Cette valeur peut être déterminée par défaut dans la recette comme indiqué ci-dessous.
- **margin\_right** | Marge à droite. Quand l'image est flottante à gauche, détermine la distance de l'image avec le texte à droite. Quand l'image est flottante à droite, détermine la distance entre l'image et la marge droite. Cette valeur peut être déterminée par défaut dans la recette comme indiqué ci-dessous.
- **margin\_bottom** | Marge en bas entre le bas de l'image et la première ligne du texte associé à l'image. Attention : il s'agit bien du *texte associé à l'image* et non pas d'un paragraphe suivant qu'il faut éloigner de l'image soit avec `space_after` soit avec des `(( line ))`.
- **text\_width** | Largeur que doit occuper le texte à côté de l'image. Si non défini, tout l'espace possible.

Note : on ne permet pas l'utilisation d'une image flottante au milieu, ce qui se fait rarement dans un livre qui n'est pas purement graphique.

### POSITIONNEMENT EXACT DE L'IMAGE

On peut déterminer le positionnement exact de l'image de façon générale, dans le recette (pour que toutes les images disposent de la même présentation) ou pour chaque image en particulier.

### PASSAGE NATUREL À LA PAGE SUIVANTE

Pour le moment, **Prawn-For-Book** ne gère pas le passage naturel à la page suivante lorsque l'image et son texte enroulé ne tiennent pas dans la page. Il faut donc le gérer *manuellement*, à l'aide de `(( line ))` et de `(( new_page ))` par exemple.

## ROTATION DE L'IMAGE

Pour le moment, en mode simple, **Prawn-For-Book** ne permet pas d'effectuer une rotation de l'image. Mais cette fonctionnalité sera implémentée plus tard.

Pour le moment, la solution consiste à faire une image déjà tournée et à jouer sur des valeurs négatives pour bien placer le texte. Ou, en mode expert, à l'imprimer explicitement avec du code ruby.

### *Définition des valeurs par défaut dans la recette*

---

```
---
book_format:
  text:
    # Distance avec l'image flottant à gauche
    left_margin_with_floating_image: 10
    # Distance avec l'image flottant à droite
    right_margin_with_floating_image: 10
```



## Exemples divers

## Texte enroulé autour de l'image

Un texte avec deux lignes au-dessus de l'image et qui va s'enrouler ensuite par la droite jusqu'au milieu de l'image ensuite. Il est obtenu avec un *float* à *:left*, un *lines\_before* à 2 et ensuite toutes les marges à 0. Le *margin\_top* à 0 fait que l'image « colle » aux deux lignes supérieures. Le *margin\_left* fait que l'image « colle » à la marge gauche et le *margin\_right* fait que le texte « colle » à l'image. Et enfin, le *margin\_bottom* fait que le texte qui passe sous l'image est collé à l'image comme le texte au-dessus. Noter cependant que le texte, se plaçant toujours sur des lignes de référence, se trouve plus éloigné de l'image en bas car la ligne de référence se trouve plus loin. En fonction du texte, on joue sur les propriétés *margin\_top* et *margin\_bottom* pour obtenir la meilleure harmonie. Il faut toujours s'arranger pour que l'air au-dessus et au-dessous de l'image soit toujours le même. C'est ce que l'on fait avec l'image suivante, en partant sur les mêmes bases que cette image-ci.



Même chose que ci-dessus, mais la position de l'image a été ajustée avec un *margin\_top*: 4 pour être bien au milieu verticalement. Obtenu avec un *float*: *:left*, un *lines\_before*: 2 et les deux marges *margin\_left* et *margin\_right* à 0. Le *margin\_left* fait que l'image « colle » à la marge gauche et le *margin\_right* fait que le texte « colle » à l'image. Et enfin, le *margin\_bottom* fait que le texte qui passe sous l'image est collé à l'image comme le texte au-dessus. Noter cependant que le texte, se plaçant toujours sur des lignes de référence, se trouve plus éloigné de l'image en bas car la ligne de référence se trouve plus loin. En fonction du texte, on joue sur les propriétés *margin\_top* et *margin\_bottom* pour obtenir la meilleure harmonie. Il faut toujours s'arranger pour que l'air au-dessus et au-dessous de l'image soit toujours le même. C'est ce que l'on fait avec cette image, en partant sur les mêmes bases que l'image précédente.



Même chose que ci-dessus, mais avec les valeurs de marge droite et gauche (*margin\_left* et *margin\_right*) laissées à leur valeur par défaut, c'est-à-dire respectivement 0 et 10. Donc : *float*: *:left*, position de l'image a été ajustée avec un *margin\_top*: 4 pour être bien au milieu verticalement. *lines\_before*: 2 pour laisser deux lignes au-dessus, *margin\_left* à 0 et *margin\_right* à 10. Et enfin, le *margin\_bottom* fait que le texte qui passe sous l'image est collé à l'image comme le texte au-dessus. Pour rappel, ce texte s'enroule autour de l'image parce qu'il est précédé d'un point d'interrogation (!Même chose que ci-dessus, [etc.]). En fonction du texte, on joue sur les propriétés *margin\_top* et *margin\_bottom* pour obtenir la meilleure harmonie. Il faut toujours s'arranger pour que l'air au-dessus et au-dessous de l'image soit toujours le même.



Ce texte commence avec deux lignes au-dessus de l'image car `lines_before` est à 2, puis s'enroule à droite de l'image pour repasser ensuite ses dernières lignes sous l'image.



Aucune marge n'est ajoutée à gauche (`margin_left: 0`) et un espace de 40 (donc plus grand que l'espace par défaut qui est de 10) est défini entre l'image et le texte grâce à la propriété `margin_right`).

De la même manière que pour le haut, on a mis un `margin_bottom` à une grosse valeur pour montrer comment on peut avoir beaucoup d'espace entre le texte et l'image en jouant sur toutes ces propriétés.

On voit aussi grâce à cet exemple comment plusieurs paragraphes peuvent être « enroulés » autour d'une image simplement en les précédant d'un point d'exclamation (voir le code du texte et le rendu). Noter que le nombre de lignes sous l'image n'a pas besoin d'être précisé, contrairement au nombre

de lignes au-dessus, puisque c'est simplement tout le texte restant qui sera écrit sous l'image, au besoin. Ici, on n'aura que deux lignes, mais ça pourra dépendre encore des changements de taille du présent manuel.

## Texte à côté de l'image



Un texte qui va simplement se placer à droite de l'image, sans dépasser ni en haut ni en bas, avec float: :left et width: 100. Code:

```
!Un texte qui va simplement [etc.].  
![exemples/image.jpg](float: :left, width: 100)
```

Le paragraphe qui suit l'image se place bien en dessous et, normalement, ne doit pas être mangé par le bas de l'image.

Texte placé à gauche d'une image qui porte le float à :right (flottante à droite) et le :width à 100. Code :

```
!Text placé à gauche d'une [etc.]  
![exemples/image.jpg](float: :right, width: 100)
```



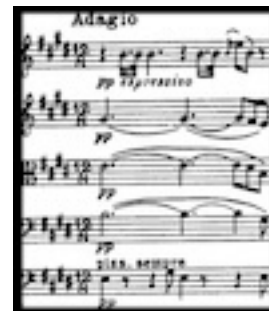
Le paragraphe qui suit l'image se place bien en dessous et, normalement, ne doit pas être mangé par le bas de l'image.

## Image flottante avec légende

Un texte qui s'enroule autour d'une image flottante à droite qui possède une légende normale et toutes les valeurs par défaut.

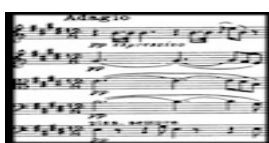
Code utilisé :

```
!Un texte qui s'enroule autour d'une [etc.]
![exemples/image.jpg](float: :right, legend: « La
légende de l'image »)
```



La légende de  
l'image

Dans cette exemple d'image avec une légende, le texte s'enroule vraiment autour de l'image c'est-à-dire qu'un `lines_before` à 2 permet de laisser passer deux lignes au-dessus et la longueur du texte fait passer des lignes en dessous de la légende. Un `margin_left` à 20 permet de décoller l'image de la marge gauche tandis qu'un `margin_right` à 20 éloigne un peu le texte horizontalement. Un `margin_bottom` à 8 fait descendre d'une ligne les lignes sous l'image et pour bien positionner verticalement l'image et sa légende, un `vadjust` à 12 fait descendre l'image (ce qui est nécessaire puisque le texte se place toujours sur les lignes de référence — il est donc nécessaire, en fonction de l'image, d'ajuster la position verticale comme on le ferait à la main).



La légende de  
l'image

Pour obtenir l'effet ci-dessus, on a utilisé le code :

```
!Dans cette exemple d'image avec une légende, le texte [etc.]
![exemples/image.jpg](float: :left, width: 100, height: 50,
legend: « La légende de l'image », lines_before: 2, vadjust: 12,
margin_bottom: 8, margin_left: 20, margin_right: 20)
```

## Autre cas d'utilisation



Pour une image qui flotte à gauche (`float: :left`), éloignement de la marge gauche avec un `margin_left` à 40. Le texte est éloigné de l'image avec un `margin_right` à 20 au lieu de la valeur par défaut 10.

Pour une image qui flotte à droite (`float: :right`), éloignement de la marge droite avec un `margin_right` à 40. Le texte est éloigné de l'image avec un `margin_left` à 20 au lieu de la valeur par défaut 10.



On peut définir la largeur que devra prendre le texte avec la propriété `text_width` qui est mise ici à 200 pour produire un effet de légende sur le côté. Le `margin_left` est à 40, ainsi que le `margin_right`.

## Adaptations naturelles du texte

Montrons quelques cas d'adaptation du texte à l'image en fonction de quelques propriétés comme les marges et la largeur du texte.



Un largeur de texte normale, un simple `float: left` avec dimensionnement de l'image (pour avoir moins de lignes)

On place une ligne au-dessus (`lines_before: 1`) et une largeur de texte plus réduite

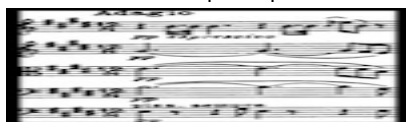


(`text_width: 200`) mais qui ne sera donc appliquée que lorsque le texte passera à côté de l'image.

Lorsqu'il repassera en dessous, il

trouvera sa largeur normale. On a ajouté un petit espace sous l'image (`margin_bottom: 8`) pour que le texte en soit pas trop collé à l'image en dessous (pour qu'il passe sur la ligne de référence suivante).

La même chose que précédemment, mais en éloignant l'image de la marge avec



`margin_left: 30` et en éloignant le texte de l'image avec `margin_right: 45`. Toutes les autres valeurs restent similaires et

notamment le `margin_bottom: 8` qui permet de passer cette ligne sur la bonne ligne de référence.

Et enfin, toujours en s'inspirant de l'image ci-dessus, on se sert de la propriété `vadjust:`

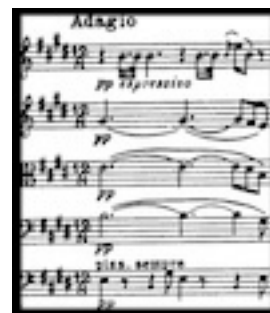


`5` (ajustement vertical) pour équilibrer l'espace vertical entre le texte et l'image, pour qu'il y ait le même espace au-dessus et au-dessous et

qu'ainsi l'image soit affichée de façon plus équilibrée.

## Cas spéciaux

Un texte qui va être placé plus bas, presque au milieu de l'image, grâce à un `margin_top` négatif.



Pour obtenir l'effet ci-dessus, nous avons eu recours à :

```
(( line ))
(( line ))
(( line ))
```

```
(( line ))
```

!Un texte qui va être placé plus bas, presque au milieu de l'image, grâce à un `margin\_top` négatif.

```
![exemples/moins_large_border.jpg](float: :right, margin_top: -40)
```



L'effet ci-dessus est obtenu à partir du code :

```
!**<font size="16"><color rgb="FF0000">Un texte qui va « manger »  
sur [...] négatif.</color></font>**
```

```
![exemples/image.jpg](float: :left, margin_right: -40)
```

---

## 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 19, 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 le fichier `recipe.yaml` ou `recipe_collection.yaml` contient...*

---

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

*Codes suivis de leur interprétation*

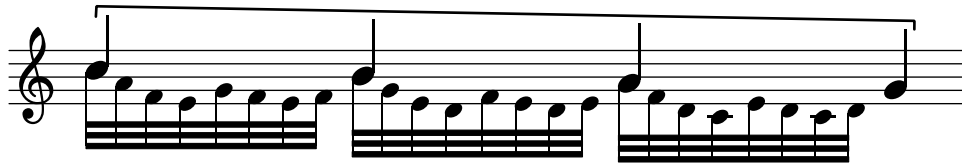
---

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



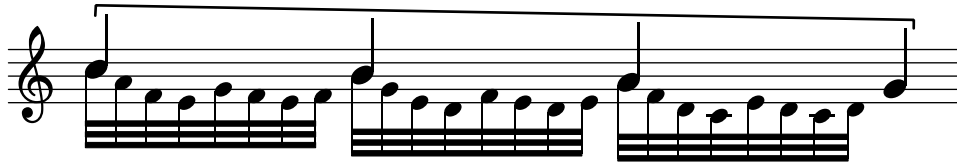


```
![exemples/image.svg](legend: "Légende avec couleur et  
ponctuelle", legend_color: [0,110,20,23], legend_size: 22)
```



*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.

---

# Bibliographies

## Introduction

**Prawn-For-Book** possède un système de gestion des bibliographies très puissant. Elle peut en gérer autant que l'on veut, donc une infinité, et les mettre en forme de la façon exacte que l'on veut, avec une connaissance minimum du langage ruby (et encore, comme d'habitude, on peut se contenter des comportements par défaut).

Une bibliographie se définit dans **Prawn-For-Book** par une donnée dans la section principale de recette `bibliographies` et une balise qui servira d'identifiant pour la bibliographie et de balise pour repérer le texte. Par exemple les balises `article` et `film` qui serviront d'illustration ci-dessous et seront utilisés dans le texte de cette manière :

« C'est une référence à l'article `article(Les chaussures|0001)` et une référence au film `film(Titanic)`. »

## LEXIQUE

*(comme pour les autres parties, commençons par un lexique qui permet de fixer les termes que nous employons pour parler de bibliographie)*

- **Bibliographie** | Ensemble de sources quelconques, de toutes natures, dont il est fait référence au cours du livre, et qu'ils faut rassembler à la fin du livre, avec ses informations, pour que le lecteur puisse s'y référer. Par exemple une liste de livres, de films ou de mots techniques. Désigne aussi l'ensemble de ses *items*.
- **item** ou **item bibliographique** | C'est un élément en particulier de la bibliographie, un film en particulier s'il s'agit d'une liste de films, un livre en particulier s'il s'agit d'une liste de livres, un mot en particulier si c'est un index de mots.
- **Marque de référence** | Marque spéciale, dans le texte, qui permet de consigner un *item bibliographique*. Elle se construit à l'aide de l'identifiant de la bibliographie, suivi de parenthèses à l'intérieur desquelles on place l'identifiant de l'*item bibliographique*. Par exemple `livre(Les Misérables)` dans la phrase Avez-vous lu `livre(Les Misérables)`, un livre de `person(Victor Hugo)`.
- **Données bibliographiques** ou **Banque de données** | Ensemble complet des *fiches* qui comprennent l'intégralité des *items* d'un bibliographie, même ceux non cités, dans laquelle on va puiser les sources pour y faire référence.
- **Fiche** ou **carte** | Fichier informatique ou enregistrement (dans un fichier CSV) qui contient toutes les données de l'*item bibliographique* (par exemple toutes les données du film ou du livre).
- **Liste des sources** | Désigne la liste, à la fin du livre, qui liste l'ensemble des *items* cités (et seulement les items cités) et, dans **Prawn-For-Book**, indique les pages où il y est fait référence.

## DONNÉES MINIMALES

Une bibliographie est définie par les données minimales et indispensables suivantes (pour la compréhension, on choisit d'illustrer une bibliographie pour des articles de journaux) :

- Le **tag** (ou **identifiant**) | C'est un mot simple en minuscule (donc seulement les lettres de « a » à « z », au singulier (p.e. « article ») qui sert de *clé* pour la bibliographie. C'est ce tag qui sera utilisé au cours du livre pour construire un *marquer de référence* pour faire référence à un *item bibliographique* en particulier (cf. plus bas).
- **title** (« titre » en anglais) | Le titre de la bibliographie, qui sera imprimé avant d'afficher la bibliographie.
- **path** (« chemin d'accès » en anglais) | qui indique le chemin jusqu'à la *banque des données bibliographiques* qui contient, comme son nom l'indique, toutes les données bibliographiques. C'est au mieux un dossier (contenant les *cartes* ou les *fiches* de données, une par *item*) ou un fichier (contenant toutes les données).
- **main\_key** (« clé principale » en anglais) | Si ce n'est pas la propriété `title` de l'*item* (à ne pas confondre avec le `title` dont nous venons de parler) qui doit être utilisée pour remplacer la marque de référence, c'est la valeur de cette clé principale qu'il faudra prendre (et qui doit donc impérativement être définie pour chaque *item bibliographique*)

Ce sont les données minimales à définir pour qu'une bibliographie soit utilisable. Dans la recette (cf. page 19), elles sont définies de cette manière :

bibliographies:

```
<tag>:
# Propriétés indispensables
title: "<titre>"
path: "<path/to/data>"
main_key: « <clé item> » # si autre que title
# Propriétés optionnelles
format: <format>
picto: "<picto>"
```

À partir de là, dans le texte, il suffit de mettre un mot ou un grand de mot entre parenthèses en le précédant de l'identifiant de la bibliographie pour que cet élément soit pris en repère. Par exemple :

```
<id biblio>(<id item>)
```

Ce mot sera remplacé par la propriété `title` de l'*item* (cf. plus bas) et cette référence sera enregistrée pour l'ajouter à la liste des sources bibliographiques à la fin du livre.

Si le `title` de l'*item* ne convient pas dans une utilisation particulière, on peut définir un texte quelconque avant l'identifiant, en le séparant avec un trait droit (« | ») de l'identifiant, comme ci-dessous.

```
C'est un film(exemple de film|the titanic) dans le texte.
```

Avec le code ci-dessus, le texte « exemple de film » sera utilisé à la place du titre du film, mais c'est une référence au film en question qui sera enregistrée en liste des sources en fin de volume.

## DONNÉES OPTIONNELLES

Pour commencer à personnaliser l'affichage de la bibliographie (ou des *items* dans le texte), on peut utiliser ces propriétés optionnelles :

- **picto** | Pictogramme à utiliser avant le texte qui remplacera la *marque de référence* dans le texte du livre.
- **title\_level** (« niveau de titre » en anglais) | Le niveau de titre pour la section qui liste les sources citées. Par défaut, ce niveau est 1, le plus grand titre.
- **format** | Format à utiliser pour l'affichage de la *liste des sources*,
- **key\_sort** (« clé de classement » en anglais) | Clé de classement des *items* dans la *liste des sources*

## DONNÉES BIBLIOGRAPHIQUES

Dans l'idéal, les données sont consignées dans un dossier qui contient chaque donnée sous forme de fiche (un fiche par item) au forme YAML (le format des recettes). Ce format permet une édition facile des données. Cependant, vous pouvez aussi utiliser le format JSON ou même TXT (simple texte).

## ITEMS DE LA BIBLIOGRAPHIE

Au minimum (mais ce serait un peu idiot de n'avoir que ça...), un item de bibliographie doit définir sa propriété `title` qui correspond à son titre. `title` est sa clé principale. Mais puisque **Prawn-For-Book** est hautement configurable, même cette clé principale peut avoir un autre nom, il suffit de la définir dans la propriété `main_key` des données de la bibliographie, dans la recette.

Donc un item bibliographique peut ressembler à :

```
# Dans "the titanic.yaml"
---
title: "The Titanic"
title_fr: "Titanic"
director: "James CAMERON"
year: 1999
```

## LISTE DES SOURCES

Dans le livre, il suffit de placer le code `(( id bibliographie> ))` pour insérer la bibliographie à l'endroit voulu. Par défaut, toutes les informations seront affichées, mais il sera possible de tout formater à sa convenance (cf. plus loin).

Par exemple :

```
(( biblio(film) ))
```

... pour afficher la liste des films cités (et seulement ceux cités) et/ou :

```
(( biblio(article) ))
```

... pour afficher la liste des articles (seulement ceux cités), avec les pages où ils sont cités.

## FORMATAGE DE LA LISTE DES SOURCES

Le formatage de l'affichage de la *liste des sources* en fin d'ouvrage se définit, comme nous l'avons vu, par la propriété `format` de la bibliographie, dans la recette.

Cette propriété n'est pas obligatoire et en son absence, ne sera affiché à la fin du livre que le titre de l'*item* (propriété `title`) ainsi que la liste des pages (ou autre indication en fonction de la `_pagination_` choisie) où cet *item bibliographique* est cité. C'est le cas, dans l'exemple, pour la liste des articles par exemple (cf. ci-dessous).

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

---

```
---
bibliographies:
  # Ici la définition de toutes les bibliographies

  # Définition de la première bibliographie

  article: # id bibliographie
    title: "Articles de presse"
    title_level: 2
    path: "chemin/vers/données/articles"

  # Définition de la deuxième bibliographie

  film:
    title: "Liste des films"
    path: "chemin/vers/données/films"

  # Définition d'une autre bibliographie

  autrebib:
    title: "Pour propose autre chose"
    path: "..."
    picto: :fiche
    main_key: "truc" # autre clé que 'title'
```

Si le fichier « `texte.pfb.md` » contient...

---

Un texte exemple qui utilise un article(premier article) en référence bibliographique. Ce paragraphe contient aussi une référence au film film(The Titanic) qui se déroule sur un bateau.

Le livre final (document PDF) contiendra :

---

Un texte exemple qui utilise un Mon Premier Article en référence bibliographique. Ce paragraphe contient aussi une référence au film The Titanic qui se déroule sur un bateau.

---

## Personnalisation des bibliographiques

Comme pour tout élément de **Prawn-For-Book**, on peut le garder simple ou au contraire le formater tel qu'on le désire. C'est pratiquement incontournable pour les bibliographies, au moins au niveau de la *liste des sources*.

## DÉFINIR LA FONTE

Il n'est pas très heureux de modifier la fonte à l'intérieur d'un texte, ça le rend beaucoup plus compliqué à lire et même à afficher. En revanche, on peut profitablement introduire des pictogrammes discrets qui permettent de caractériser les éléments.

## FORMATAGE DE LA LISTE DES SOURCES

Si la propriété `format` de la bibliographie est fournie dans la recette, elle se sert de marques de type `%{<propriété>}` pour définir le format à utiliser, où `<propriété>` est le nom de la propriété dans les données de l'*item bibliographique* (à commencer par `title`, son titre).

Lorsque la valeur doit être modifiée, par exemple mise en capitales, on ajoute la méthode de transformation après un trait droit « `|` » :

```
%{propriété|méthode transformation}
```

(voir ci-dessous les méthodes existantes).

Par exemple :

```
format: "%{title|all_caps} (%{year} – %{year|age}, réalisation :
%{director|person}, acteurs : %{actors|person}, durée :
%{duration|horloge})"
```

Ce *format* va afficher le titre, tout en majuscules (`all_caps`), puis, entre parenthèses, l'année de sortie (`year` qui signifie « année » en anglais), le réalisateur (précédé du terme « réalisation : » et traité comme un individu) et enfin la liste des acteurs et actrices, traités comme des individus, précédée de la marque « acteurs : ».

Optionnellement, si la liste des références ne doit pas s'afficher à la suite de l'*item* après un double-point, ajouter `%{pages}` à l'endroit voulu.

Vous pouvez voir ci-dessous le rendu de ce formatage.

## MÉTHODE DE FORMATAGE COMMUNES

- `all_caps` | Met la donnée en majuscule.
- `person` | Transforme une donnée « Prénom NOM » en « Prénom Nom » ou une liste de personnes de la même manière.
- `age` | Transforme une donnée « année » en âge par rapport à maintenant.
- `horloge` | Transforme une donnée secondes en horloge de type `h:mm:ss`.
- `minute_to_horloge` | Transforme une donnée minutes en horloge de type `h:mm:ss`.

En mode expert (cf. page 61) vous pouvez également définir toutes formes de méthodes de transformation.

*Extrait de la recette*

---

---

```
# ...
bibliographies:
  article:
    title: Articles de presse
    title_level: 2
    path: biblios/articles
    picto: :fiche
  film:
    title: Films cités
    title_level: 2
    path: biblios/films
    picto: clap
    format:      "%{title|all_caps}      (%{year}      - %{year|age},
réalisation :  %{director|person},
    acteurs :      %{actors|person},      durée :
%{duration|minute_to_horloge}). Occurrences :
    %{pages}"
  costum:
    title: Liste des costumes
    title_level: 3
    path: biblios/costums
    main_key: nom
    format: "%{nom|transforme_nom}"
```

*Si le fichier « texte.pfb.md » contient...*

---

```
Un costum(cravate) pour voir.
(( new_page ))
(( biblio(costum) ))
(( new_page ))
```

*Le livre final (document PDF) contiendra :*

---

Un Cravate pour voir.



## Liste des costumes

**Cravate** : 56.



# 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**.

## 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.

## Évaluation au second tour

Certaines opérations ou certains calculs ne peuvent s'opérer qu'au second tour de l'application<sup>2</sup> — typiquement, le nombre de pages du livre —. On utilise alors la tournure suivante pour réaliser ces opérations.

```
#{{{ "#{operation}" if Prawn4book.second_turn? } } }3
```

Dans le code ci-dessus, le contenu des guillemets ne sera évalué qu'au second tour de l'application. Mais attention, cela peut occasionner un changement des numéros de page si le texte ajouté au second tour est conséquent. Il est donc plus prudent de mettre au premier tour un texte d'environ la longueur du résultat attendu pour ne pas fausser le suivi. Pour le numéro des pages, que nous estimons au départ à plusieurs centaines mais moins d'un millier nous utilisons :

```
#{{{ Prawn4book.first_turn? ? « XXX » : « #{book.pages.count} » } } }
```

... qui signifie qu'au premier tour, **Prawn-For-Book** marquera simplement « XXX » et aux suivants il inscrira le nombre de pages.

---

<sup>2</sup> C'est le cas par exemple de l'impression du nombre de pages de ce manuel dans l'avant-propos, c'est-à-dire alors que le livre est à peine esquissé.

<sup>3</sup> Noter ici l'utilisation des trois accolades obligatoires lorsque le code lui-même a recours aux accolades.



Une opération simple permet de savoir que  $2 + 2$  est égal à `{2+2}` et que le jour courant (au moment de l'impression de ce livre) est le `{Time.now.strftime('%d %m %Y')}`.

*Le livre final (document PDF) contiendra :*

---

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 10 12 2023.

---

## Bibliographies en mode expert

TODO: Décrire comment faire une méthode de formatage propre dans `Prawn4book::Bibliography` (méthode d'instance) pour l'utiliser dans la liste des sources, pour une propriété particulières. Si, par exemple, la donnée `format` de la bibliographie (dans la recette), définit `{title|mon_transformeur}`, alors il faut définir la méthode `Prawn4book::Bibliography#mon_tranformeur` qui reçoit en argument la valeur de `:title` de l'item.

TODO: Montrer qu'on peut par exemple définir une font et/ou une couleur propre comme pour la bibliographie `costume`

## 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 19) ou pour une collection (cf. page page 19). 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 le fichier `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"
```

---

## Films cités

THE TITANIC (1999 — 24 ans, réalisation : James Cameron, acteurs : Kate Wistlet et Leonardo Dicaprio, durée : 2:56:48). Occurrences : : 54.

## Articles de presse

Mon Premier Article : 54.



AVANT-PROPOS .....	11
FONCTIONNALITÉS .....	15
Généralités .....	17
Les Forces de <i>Prawn-For-Book</i> .....	17
Les deux fichiers de base .....	17
Les Recettes .....	19
Recette du livre .....	19
Recette de la collection .....	19
Définition des marges .....	21
Définition de la fonte par défaut .....	23
Éditeur / Maison d'édition .....	23
Les Images .....	24
Insérer une image .....	25
Les Images flottantes .....	39
Bibliographies .....	51
Introduction .....	51
Liste des costumes .....	57
Le Mode Expert .....	59
Description .....	59
Évaluation du code ruby .....	59
Bibliographies en mode expert .....	61
Le format YAML de la recette .....	61
Films cités .....	62
Articles de presse .....	62









Icare Éditions

Conception  
Philippe Perret  
([philippe.perret@icare-editions.fr](mailto:philippe.perret@icare-editions.fr))

Mise en page  
Prawn-for-book

Couverture  
MM & PP

Correction & relecture  
Marion Michel

Imprimé par Prawn-For-Book