









# Manuel de Prawn-for-book

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

Philippe Perret

Icare Éditions









AVANT-PROPOS .....	15
FONCTIONNALITÉS .....	19
Format du livre et des pages .....	21
Définition des marges .....	21
Section en multi-colonnes .....	24











# 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 XXX pages est auto-produit par « **Prawn-for-book** », c'est-à-dire qu'il est construit de façon *programmatische* 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.





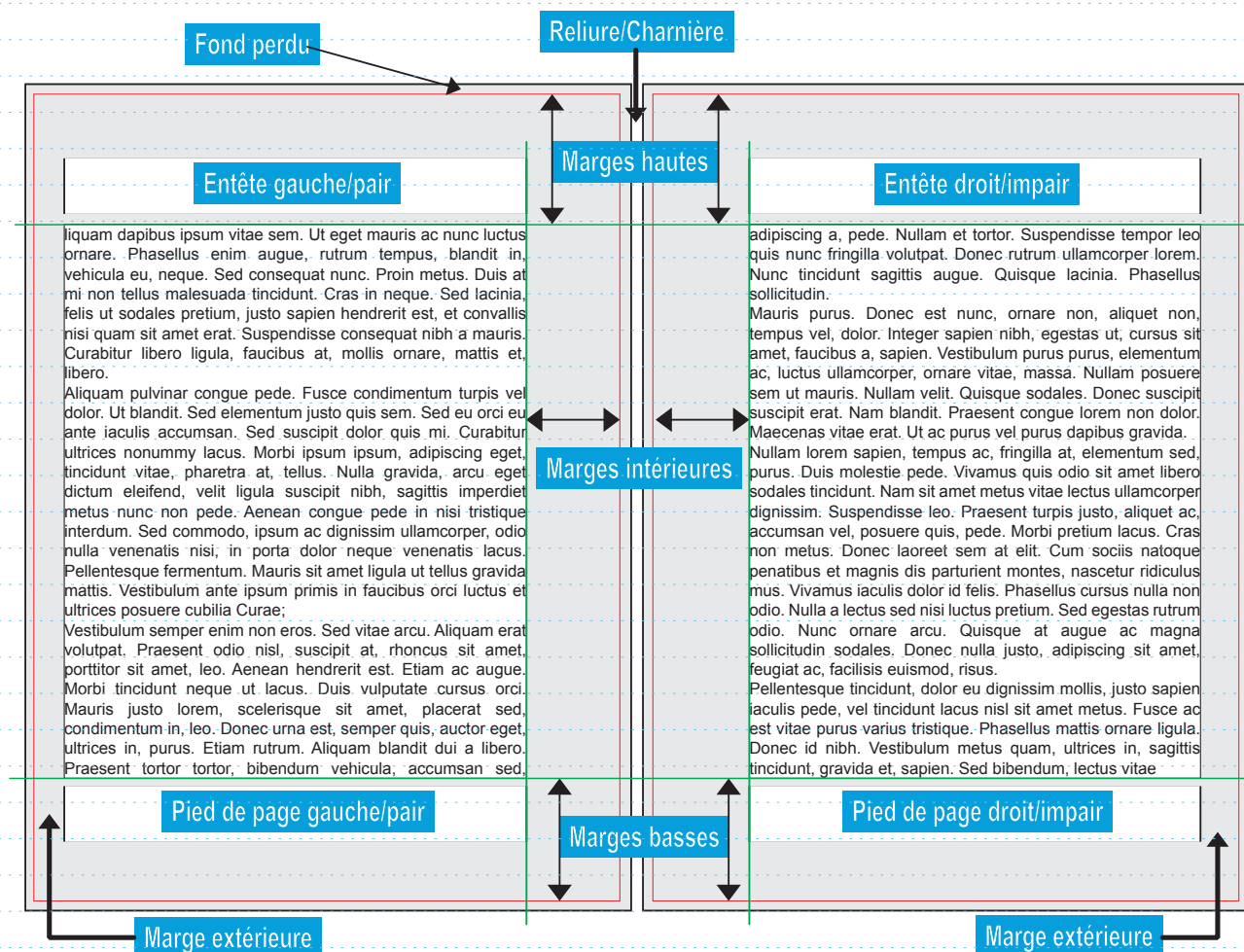


# Fonctionnalités



# Format du livre et des pages

## 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 aux documents dont vous avez l'habitude, c'est qu'ici les pages sont paires ou impaires, 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, près de la reliure, de la *charnière*, du *dos du livre* (souvent confondu avec la *tranche du livre*),
- une marge *extérieure*, qui comme son nom l'indique est tournée vers l'extérieur du livre, vers la tranche (la vraie cette fois).

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

NB : Quels que soient les réglages, il y aura toujours un *fond perdu* (« bleeding » en anglais) de 10 pps (points-postscript) autour de la page. C'est la « marge » que s'accorde l'imprimeur pour découper le livre.

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 `-margins` à la commande `pdfbuild` 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 la recette du livre contient...

```
book_format:
  page:
    margins:
      top: 40mm
      bot: 20mm
      ext: 2mm
      int: 35mm
    show_margins: true
```

#### ET que le texte du livre contient...

Pour cette page, où les marges sont visibles, on illustre des marges de page à 40 mm en haut, 20 mm en bas, 35 mm à l'intérieur et 2 mm à l'extérieur.

Vous remarquez donc une immense marge en haut, une grande marge en bas, 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 le bord de ces marges gauche et droite, ce qui donne un rendu impeccable.

#### ALORS le livre contiendra...

Les marges sont visibles grâce au `show_margins` mis à `true` dans la recette. On aurait pu aussi jouer la commande avec l'option `-margins`.

Pour cette page, où les marges sont visibles, on illustre des marges de page à 40 mm en haut, 20 mm en bas, 35 mm à l'intérieur et 2 mm à l'extérieur.

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

1

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

2

## Section en multi-colonnes

On peut provisoirement passer en double colonnes ou plus grâce à la marque :

```
(( colonnes(nombre de colonnes) ))
```

C'est ce qu'on appelle le *mode multi-colonnes*.

Pour quitter le mode multi colonnes, il suffit de revenir à une seule colonne :

```
(( colonnes(1) ))
```

Vous l'aurez déjà compris, grâce à cette marque, on peut avoir autant de colonnes que l'on désire, comme ci-dessous.

Un paragraphe au-dessus de la section multi-colonnes.

Ce texte se trouve en mode multi-colonnes, avec trois colonnes, grâce à la marque

```
(( colonnes(3) ))
```

qui se trouve au-dessus et la marque

```
(( colonnes(1) ))
```

qui se

trouve en dessous. On peut forcer le nombre de lignes.

Un paragraphe au-dessous de la section multi-colonnes.

### Définition plus précise des colonnes

Comme pour tout élément **Prawn-For-Book**, le comportement par défaut est harmonieux et devrait apporter satisfaction à tout utilisateur. Mais comme pour tout élément, on peut néanmoins redéfinir de façon précise de nombreux éléments, simplement en ajoutant un deuxième paramètre à la méthode `colonnes` après le nombre de colonnes.

Ce paramètre est une table ruby, donc entre accolades, avec des paires « propriété : valeur ». Par exemple :

```
(( colonnes\(<nombre cols>, {\<prop>: \<valeur>, \<prop>: \<valeur>, etc.}) ))
```

### Largeur de gouttière entre les colonnes

On appelle *gouttière* l'espace vertical laissé entre deux colonnes. On peut le redéfinir avec la propriété `gutter` (« gouttière » en anglais).

```
(( colonnes(2, gutter: 40) ))
```

### Largeur totale occupée par les colonnes

On peut définir aussi sur quelle largeur les colonnes devront tenir, par exemple la moitié de la page :

```
(( colonnes(2, width: PAGE_WIDTH/2) )) 1
```



<sup>1</sup> Vous remarquez ci-dessus l'utilisation d'une constante (cf. `annexe_constantsannexe/constants`).

## Lignes avant et après la section multicolonnes

Grâce à `lines_before` et `lines_after`, on peut définir le nombre de lignes à laisser entre le texte et la section en multi-colonnes. Il s'agit très précisément du *nombre de lignes vides* avant (`lines_before` — "lignes avant" en anglais) et après (`lines_after` — "lignes après" en anglais) la section multi-colonnes.

Par exemple, avec :

```
(( colonnes(2, {lines_before: 2, lines_after:3}) ))
```

... on laissera 2 lignes vides entre le paragraphe précédent et le début de la section à 2 colonnes, et 3 lignes vides entre la fin de la section multi-colonnes et le paragraphe suivant.

## Styles du texte dans les multi-colonnes

On peut modifier les paragraphes de façon générale à l'intérieur d'une section multi-colonne grâce au deuxième paramètre.

Les propriétés modifiables sont :

- **align** | Alignement des paragraphes. Justifiés (`:justify`) par défaut, on peut les aligner à gauche (`:left` ou `LEFT`) ou à droite (`:right` ou `RIGHT`). La section ci-dessous est obtenue avec le code

```
(( colonnes(2, {align: RIGHT}) ))
```

Un paragraphe normal situé au-dessus de la section à double colonnes.

Ce texte est aligné à droite dans la section double colonne grâce à la propriété `align` mise

à `RIGHT`. Rappel : par défaut, le texte est justifié.

Un paragraphe normal situé sous la section à double colonnes.

- **font** | Fonte utilisée pour le texte. C'est une `annexe_font_stringannexe/font_string` classique. La section ci-dessous est engendrée par le code

```
(( colonnes(2, {font:"Arial/bold/8.5/008800"}) ))
```

Un paragraphe normal situé au-dessus de la section à double colonnes.

Une section en double colonnes avec la police « Arial », le style « bold », une taille de 8.5 et une couleur vert

foncé (008800). Par défaut, c'est la police du livre qui est utilisée.

Un paragraphe normal situé sous la section à double colonnes.

## Très long texte en multi-colonnes

Si vous avez un extrêmement long texte en multi-colonnes (par exemple tout votre livre), il est préférable de le diviser en plusieurs sections plutôt que de le faire tenir dans une seule section (ce qui entrainerait fatalement des problèmes).



Par exemple, renouvelez la marque `(( colonnes(2) ))` à chaque nouveau chapitre (sans oublier de terminer la section précédente par `(( colonnes(1) ))`).

## Espace avant et après la section multi-colonnes

On peut utiliser de la même manière `space_before` et `space_after`, en leur donnant comme valeur une distance (en points-postscript, en millimètre, etc.).

Mais les propriétés `lines_before` et `lines_after` doivent être préférées, sauf dans le cas où vous connaissez les distances précisément, en pouce, millimètre ou autre, à avoir entre le texte et la section multi-colonnes.

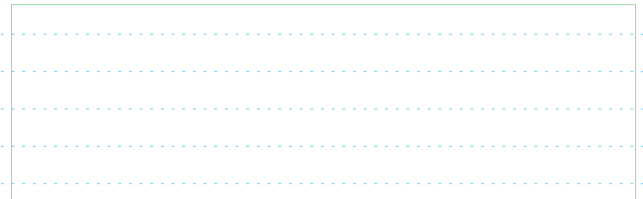
## Ajouter ou retrancher des lignes en multi-colonnes

Malgré tous nos efforts, ou pour des besoins propres, il est possible que les colonnes ne correspondent pas à ce que l'on attend au niveau de leur hauteur.

Dans ce cas, grâce à la propriété `add_lines`, on peut ajouter ou retrancher un certain nombre de lignes. Par exemple, ci-dessous, on force l'affichage dans une seule colonne à l'aide de `(( colonnes(2, {add_lines: 3}) ))`.

Un paragraphe normal situé au-dessus de la section à double colonnes.

Un texte qui devrait tenir sur les deux colonnes mais qu'on a allongé en hauteur pour avoir 3 lignes de plus. Donc ce texte tient intégralement sur la première colonne comme voulu par le nombre.



Un paragraphe normal situé sous la section à double colonnes.

## Nombre fixe de lignes en multi-colonnes

De la même manière et malgré tous nos efforts, il est possible de fixer de façon précise le nombre de lignes.

On peut utiliser soit la propriété `lines_count` pour définir le nombre de lignes (c'est la valeur) que doit avoir la section multi-colonnes, soit la propriété `height` pour définir la hauteur avec une unité de mesure (pouces, millimètre, etc.).

```
(( colonnes(3, {lines_count: 10}) ))
# ...
(( colonnes(3, {height: "4cm"}) ))
```

## Précaution pour les sections multi-colonnes

Attention à toujours terminer par `(( colonnes(1) ))`, même lorsque c'est la dernière page. Dans le cas contraire, en cas d'oubli, la section multi-colonnes — et tout son texte — ne sera tout simplement pas gravée.

Le livre final (document PDF) contiendra :

Un premier paragraphe en haut de page qui commence en mode normal (sans colonne). Juste sous ce paragraphe, on a inscrit le code (invisible ici) : `(( colonnes(3) ))` qui permet de passer la suite en triple colonnes.

Début du texte. In mollit anim veniam est ut officia sit mollit est dolor consequat cillum. In mollit anim veniam est ut	officia sit mollit est dolor consequat cillum. In mollit anim veniam est ut officia sit mollit est dolor consequat	cillum. In mollit anim veniam est ut officia sit mollit est dolor consequat cillum. (il faudra remettre 20 fois) Fin du texte.
--	--	--

## Problèmes avec les multi colonnes

Cette section décrit les problèmes qui peuvent survenir avec les doubles ou multi colonnes.

*Note : dans l'idéal, ce fichier devrait être supprimé car il sert surtout à reproduire les erreurs pour pouvoir les corriger dans l'application.*

*Si la recette du livre contient..*

---

```
---
book_format:
  page:
    margins:
      top: 2mm
      bot: 8mm
```

*ET que le texte du livre contient...*

---

Une section en double colonnes, avec seulement deux items courts qui ne dépassent pas les colonnes.

`(( colonnes(2) ))`

Premier item

Deuxième item

`(( colonnes(1) ))`

Texte sous la section multi-colonnes.

`(( new_page ))`

Section en double colonnes avec seulement deux items mais plus longs qu'une colonne (donc un passage à la ligne).

`(( colonnes(2) ))`

Premier item assez long pour passer à la ligne

Deuxième item assez long pour passer à la ligne

`(( colonnes(1) ))`

Texte sous la section multi-colonnes.

`(( new_page ))`

Section en double colonnes avec trois items courts les un au-dessus des autres.

`(( colonnes(2) ))`

Un premier item

Un deuxième item

Un troisième item

`(( colonnes(1) ))`

Texte sous la section multi-colonnes.

`(( new_page ))`

Section en double colonnes avec un long paragraphe unique qui passe de la première colonne à la seconde.

(( colonnes(2) ))

Le paragraphe de cette section doit aller de la 1<sup>er</sup> colonne à la deuxième et surtout il doit générer deux colonnes qui sont de la bonne hauteur pour s'ajuster parfaitement au contenu sur plusieurs lignes.

(( colonnes(1) ))

Texte sous la section multi-colonnes.

(( new\_page ))

Section en double colonnes avec plusieurs paragraphes qui passent de l'une à l'autre.

(( colonnes(2) ))

Le paragraphe de cette section doit aller de la 1<sup>er</sup> colonne à la deuxième.

Il doit surtout générer deux colonnes qui sont de la bonne hauteur.

Les colonnes se doivent de s'ajuster parfaitement au contenu sur plusieurs lignes.

(( colonnes(1) ))

Texte sous la section multi-colonnes.

(( new\_page ))

Section en triple colonnes avec un long paragraphe unique qui passe de la première colonne à la seconde puis à la troisième.

(( colonnes(3) ))

Le paragraphe de cette section doit aller de la 1<sup>er</sup> colonne à la deuxième et ensuite à la troisième de la même façon et surtout il doit générer 3 colonnes de la bonne hauteur pour bien s'ajuster au contenu sur x lignes.

(( colonnes(1) ))

Texte sous la section multi-colonnes.

(( new\_page ))

Section en triple colonnes avec plusieurs paragraphes qui passent de l'une à l'autre.

(( colonnes(3) ))

Le paragraphe de cette section doit aller de la 1<sup>er</sup> colonne à la deuxième.

Nouveau paragraphe pour aller à la troisième de la même façon.

Il doit surtout générer 3 colonnes de la bonne height pour bien s'ajuster au contenu sur x lignes.

(( colonnes(1) ))

Texte sous la section multi-colonnes.

(( new\_page ))

Section en triple colonnes avec plusieurs paragraphes \_\_alignés à gauche\_\_ et \_\_indentés\_\_ qui passent de l'une à l'autre.

(( colonnes(3) ))

(( {align: :left, indent: "5mm" } ))

Un texte très très long pour qu'il tienne sur plusieurs colonnes.

Pour voir comment le texte se répartira.

Il ne devrait pas être justifié, mais aligné à gauche.

(( colonnes(1) ))

Texte sous la section multi-colonnes.

(( new\_page ))

Un texte au-dessus de la section à deux colonnes avec \_\_changement d'alignement\_\_ des paragraphes.

(( colonnes(2) ))

Un premier paragraphe avec l'alignement par défaut, c'est-à-dire un texte justifié.

(( {align: :right} ))

Le paragraphe suivant (celui-ci) est aligné à droite et il est de moyenne longueur. Il évalue  $4 + 4 = \# \{4 + 4\}$ .

(( {align: :left} ))

On passe ensuite, ici, à un paragraphe aligné à gauche de la même longueur à peu près.

```
(( {align: :center, size: 20, color: "FF0000"} ))
```

Ce paragraphe est plus différent, avec une taille de police et une couleur différente. Il est également plus long. Il comporte de l'*\*italique\** et du ***\*\*\*gras italique\*\*\****.

```
(( colonnes(1) ))
```

Texte sous la section multi-colonnes.

```
(( new_page ))
```

Ce paragraphe se trouve au-dessus d'une section à 3 colonnes qui est séparée de 1 ligne de ce texte.

```
(( colonnes(3, {lines_before: 1, lines_after:3 }) ))
```

Une ligne

Une autre ligne

Une troisième ligne

```
(( colonnes(1) ))
```

Texte sous la section multi-colonnes qui doit être séparé par 3 lignes.

```
(( new_page ))
```

Les deux pages suivantes (dont celle-ci) présentent un texte très long qui va passer sur \_\_plusieurs pages\_\_. On profite de ce long texte pour essayer beaucoup de choses (lire le texte).

```
(( colonnes(2) ))
```

Ce très long texte va comporter beaucoup de choses pour vérifier que tout s'y passe bien.

Une liste à puce pour voir :

- \* item 1

- \* item 2

- \* item 3

Un paragraphe contenant du style html, de l'*\*italique\**, du ***\*\*\*gras\*\*\****, du \_\_souligné\_\_, et même les ***\*\*\*\_\_trois ensemble\_\_\*\*\****.

```
(( {indent: "1cm"} ))
```

Ce paragraphe est indenté de 1 centimètre, ce qui est pas mal pour le coup et il écrit la date du jour où ce livre a été actualisé. Il a été actualisé le `{Time.now.strftime("%d %m %Y")}` et cette date est incroyable, non ?

Ce paragraphe-ci contient une note<sup>2</sup> numérotée automatiquement, qui devrait trouver son explication en fin de colonnes, si tout va bien. Il contient aussi une autre note<sup>3</sup> numérotée automatiquement aussi mais qui trouvera son explication sous la section à colonnes.

---

<sup>2</sup> Explication de la note initiée dans la colonne.

```
(( colonnes(1) ))
```

Texte sous la section multi-colonnes.

---

<sup>3</sup> Explication de la seconde initiée dans la colonne.

---

## ALORS le livre contiendra...

---

Le texte qui doit apparaitre dans le livre.

Le retour de la construction du livre est :

(Aucun retour, tout s'est bien passé)

Une section en double colonnes, avec seulement deux items courts qui ne dépassent pas les colonnes.

Premier item

Deuxième item

Texte sous la section multi-colonnes.

1

Section en double colonnes avec seulement deux items mais plus longs qu'une colonne (donc un passage à la ligne).

Premier item assez long pour passer à la ligne

Deuxième item assez long pour passer à la ligne

Texte sous la section multi-colonnes.

2



Section en double colonnes avec trois items courts les un au-dessus des autres.

Un premier item

Un troisième item

Un deuxième item

Texte sous la section multi-colonnes.

3

Section en double colonnes avec un long paragraphe unique qui passe de la première colonne à la seconde.

Le paragraphe de cette section de la bonne hauteur pour doit aller de la 1<sup>er</sup> colonne à la s'ajuster parfaitement au deuxième et surtout il doit contenu sur plusieurs lignes. générer deux colonnes qui sont

Texte sous la section multi-colonnes.

4

Section en double colonnes avec plusieurs paragraphes qui passent de l'une à l'autre.

Le paragraphe de cette section hauteur.

doit aller de la 1<sup>er</sup> colonne à la deuxième. Les colonnes se doivent de s'ajuster parfaitement au

Il doit surtout générer deux contenu sur plusieurs lignes.

colonnes qui sont de la bonne

Texte sous la section multi-colonnes.

5

Section en triple colonnes avec un long paragraphe unique qui passe de la première colonne à la seconde puis à la troisième.

Le paragraphe de à la troisième de la pour bien s'ajuster cette section doit même façon et au contenu sur x aller de la 1<sup>er</sup> surtout il doit lignes.

colonne à la générer 3 colonnes

deuxième et ensuite de la bonne hauteur

Texte sous la section multi-colonnes.

6

AVANT-PROPOS	15
FONCTIONNALITÉS	19
Format du livre et des pages	21
Définition des marges	21
Section en multi-colonnes	24















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 et Philippe Perret

Imprimé par Prawn-For-Book