





**Manuel de Prawn-for-book**  
Philippe Perret



italique	* ... *	<i>texte en italique</i>
gras	** ... **	<b>texte en gras</b>
souligné	___...___	<u>texte souligné</u>
exposants	1^er	1 <sup>er</sup>
	1^re	1 <sup>re</sup>
	2^e	2 <sup>e</sup>

## Description

Ce manuel présente toutes les fonctionnalités, à jour, de l'application « » (« »), 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 produit par « ». Si vous êtes intéressé(e) pour voir comment, et retrouver les deux fichiers `texte.pfb.md` et `recipe.yaml` qui le définissent, ouvrez le dossier `Manuel/manuel_building` de l'application.

{{Ajouter la possibilité d'ouvrir ce dossier dans `pfb open`}}

## Les deux fichiers de base

Un livre *Prawn-for-book* se compose à la base de deux fichiers :

- le fichier *texte* **texte.pfb.md**, contenant le contenu textuel (mais aussi images) du livre,
- le fichier *recette* **recipe.yaml** qui définit l'aspect du livre, à commencer par sa taille, ses marges, sa police par défaut, etc.







# Fonctionnalités

## Les Forces de *Prawn-for-book*

**Prawn-for-book** possède de nombreuses forces et de nombreux avantages dont nous ne pouvons que donner un aperçu non exhaustif ici.

- sans configuration ou définitions, l'application produit un document déjà valide, avec par exemple toutes les lignes de texte alignées sur une grille de référence, aucune veuve, aucune orpheline et aucune ligne de voleur, gestion correcte des références croisées, des index et des bibliographies.
- mais les outils permettent de définir tous les éléments de façon très précise et très fine pour obtenir le rendu exact demandé.

Le reste de ce manuel vous permettra de découvrir l'ensemble des fonctionnalités.

## 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 .

---

## 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 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 remarquez 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.

## Le format YAML de la recette

Comme nous l'avons dit, pour définir très précisément le livre « à la virgule près" nous utilisons une *recette* contenu dans un fichier **recipe.yaml** ("recipe" signifie « recette" en anglais) ou dans un fichier **collection\_recipe.yaml** si le livre appartient à une collection qui partage la même charte graphique.

Vous pourrez trouver des informations sur le format `YAML` en tapant ce mot dans un moteur de recherche. C'est de toute façon un format très simple, *imbriqué*, qui sépare une donnée (par exemple la hauteur de ligne) de sa valeur par deux points : `line_height: 15`.

Ci-dessus, la hauteur de la ligne ("line height" en anglais) sera de 15 points.

Les données sont *imbriquées*, comme nous l'avons dit, pour s'y retrouver plus facilement, entendu que la recette peut contenir des dizaines et des dizaines de choses (nous vous avons prévenu : on peut utiliser **Prawn-For-Book** de façon très simple, en utilisant ses données par défaut, mais on peut l'utiliser aussi de façon très précise pour régler n'importe quel aspect du livre)

Par exemple, tout ce qui relève des informations sur le livre est *rangé* dans un ensemble appelé `book_data` ("données sur le livre" en anglais). Tout ce qui relève du format du livre (pages, texte) est *rangé* dans un ensemble appelé `book_format` ("format du livre" en anglais), comme on peut le voir dans l'exemple de recette donnée ci-dessous :

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

---

```
---
book_data:
  title: "Le Titre du livre"
  author: "Auteur DU LIVRE"
book_format:
  book:
    width: 127mm
    height: 203.2mm
    orientation: portrait
  page:
    numerotation: hybrid
    skip_page_creation: true
fonts:
  Garamond:
```

```
normal: "mesFontes/Garamond.ttf"
italic: "mesFontes/Garamond-italic.ttf"
bold: "mesFontes/Garamond-bold.ttf"
bold_italic: "mesFontes/Garamond-bolditalic.ttf"
```

---

## Pagination

Par défaut, la *pagination* du livre est automatique et *intelligente*. C'est-à-dire qu'elle ne numérote que les pages qui doivent l'être :

- ne numérote pas les pages vierges (blanches),
- ne numérote pas les pages ne contenant qu'un titre,
- ne numérote pas certaines pages communes comme la table des matières, la pages des informations de fin de livre ou la page de faux titre.

Comme pour tous les comportements par défaut de **Prawn-For-Book**, il est possible de les modifier en jouant sur les paramètres de la recette.

### Numérotation des pages vierges

Les pages vierges peuvent être numérotée en modifiant la donnée `book_format: page: :no_num_if_empty`: (« no num if empty" signifie » pas de numéro si la page est vide").

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

---

```
---
book_format:
  page:
    no_num_if_empty: false
```

---

## Aspect du numéro

L'aspect du numéro (sa police, son style, sa taille, sa couleur) peut être défini très précisément dans la recette. On utilise pour cela les paramètres suivant.

La couleur est exprimée de façon hexadécimale (voir l'explication page (( 38 )) ).

*(remarquez que ci-dessous nous pouvons utiliser des variables\* au lieu de répéter plusieurs fois la même information. Ici, c'est la fonte par défaut que nous avons mise dans une variable)\**

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

---

```
---
# Définition des variables
define: &default_fonte "Times-Roman"
# Recette du livre
book_format:
  page:
    num_font_name: *default_fonte      # Police
    num_font_style: ''                 # style
    num_font_size: 13                  # taille (points)
    num_font_color: '555555'          # couleur
```

---



## Alignement sur la grille de référence

De façon naturelle et sans aucune intervention de l'auteur ou du metteur en page, les lignes de texte sont alignées sur une `tt`(grille de référence) ce qui permet un affichage professionnel.

Cette grille de référence ne dépend que du paramètre `:line_height` (hauteur de ligne) dans la recette.

Note : pour l'exemple, nous avons demandé l'[affichage de la `tt`(grille de référence)](`#afficher_grille_reference_et_marges`) afin de voir que les lignes de texte sont effectivement parfaitement alignées, malgré le changement de taille de la police.

Le livre final (document PDF) contiendra :

Un premier paragraphe dans une police de taille 20  
alors que la hauteur de ligne est réglée ici à 30.

Un autre paragraphe dans une police de taille. Lorem ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum

[illegible]

ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum lorem ipsum ipsum ipsum lorem ipsum ipsum

lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum

ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum lorem ipsum ipsum ipsum

lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum ipsum lorem ipsum ipsum ipsum lorem ipsum ipsum

lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

## Le format pseudo-markdown

Le formatage pseudo-markdown permet de mettre le texte en forme de façon très simple, sans avoir à se soucier des raccourcis clavier ou des menus, par de simples caractères, comme montré ci-dessous.

Si texte.pfb.md contient...

Un *\*paragraphe en italiques\**.

Un **\*\*paragraphe en gras\*\***.

Un \_\_texte\_\_ en souligné.

\* Un item de liste court.

\* Un item de liste plus long pour voir comment la ligne sera enroulée pour laisser la puce bien visible.

*Le livre final (document PDF) contiendra :*

---

Un *paragraphe en italiques*.

Un **paragraphe en gras**.

Un texte en souligné.

– Un item de liste court.

– Un item de liste plus long pour voir comment la ligne sera enroulée pour laisser la puce bien visible.

---

## Correction automatique des guillemets et apostrophes

En belle typographie, les guillemets et les apostrophes ne s'utilisent pas en dépit du bon sens, ils répondent, tout comme les ponctuations, à des certaines règles. Certaines sont relatives, d'autres sont absolues.

Tous les apostrophes droits (« ' ») seront remplacés par des apostrophes courbes (« ' »).

Tous les guillemets droits (« " ») — réservés à la langue anglaise — sont transformés en guillemets courbes (« ») ou en chevrons (« ... ») en fonction de la recette du livre ou de la collection.

Les espaces insécables oubliées^^ avant les ponctuations doubles sont ajoutées et les points de suspension en trois points sont remplacés par leur signe unique.

^^ « espace » est un mot féminin en typographie.

De la même manière, pour les tirets longs ou demi-longs pour mettre du texte en exergue — comme ici —, il est nécessaire de mettre à l'intérieur des espaces insécables qui empêcheront le tiret de se retrouver seul à la ligne.

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

---

```
book_format:
  text:
    guillemets: ['« ', ' »']
```

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

---

Remarquez ici le \"guillemets droits\" et il y a plein d'apostrophes droits mais laissés dans du #{"code"} reconnu comme tel.

Un point d'exclamation doit être toujours précédé d'une espace insécable! Comme le point d'interrogation? Oui... (après ce \"oui\", trois points seront remplacés par un seul signe que nous appelons \"LE point de suspension\" —essayez de les attraper avec votre souris—)

Remarquez ci-dessus le texte entre tirets longs. Des espaces ont été ajoutées.

*Le livre final (document PDF) contiendra :*

---

Remarquez ici le \"guillemets droits\" et il y a plein d'apostrophes droits mais laissés dans du code reconnu comme tel.

Un point d'exclamation doit être toujours précédé d'une espace insécable ! Comme le point d'interrogation ? Oui... (après ce « oui\", trois points seront remplacés par un seul signe que nous appelons » LE point de suspension\" — essayez de l'attraper avec votre souris —)

Remarquez ci-dessus le texte entre tirets longs. Des espaces ont été ajoutées.

---

## 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 [[recette du livre]], dans la partie `text:` de `book_format:`.

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

---

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

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

---

- \* Ceci est une puce losange de 18 points, remontée de 2 points, déplacée à droite de 4, et écartée du texte de 20 millimètres,
- \* Le second item de liste est identique,
- \* Ainsi que le troisième.

*Le livre final (document PDF) contiendra :*

---

- Ceci est une puce losange de 18 points, remontée de 2 points, déplacée à droite de 4, et écartée du texte de 20 millimètres,
  - Le second item de liste est identique,
  - Ainsi que le troisième.
-

## Puce losange noir

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

---

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

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

---

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

*Le livre final (document PDF) contiendra :*

---

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

## Puce carrée

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

---

```
#<book_format>
book_format:
  text:
    puce:
```

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

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

---

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

*Le livre final (document PDF) contiendra :*

---

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

## Puce carrée noire

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

---

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

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

---

- \* Ceci est une puce :black\_square de 20 points, remontée de 2 points, avec les autres paramètres laissés intacts,

- \* Le second item identique,
- \* Le troisième.

*Le livre final (document PDF) contiendra :*

---

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

## Puce ronde

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

---

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

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

---

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

*Le livre final (document PDF) contiendra :*

---

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



- Le troisième.
- 

## Puce ronde noire

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

---

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

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

---

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

*Le livre final (document PDF) contiendra :*

---

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

## Puce doigt tendu

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

---

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

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

---

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

*Le livre final (document PDF) contiendra :*

---

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

## Puce doigt tendu noir

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

---

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

#</book\_format>

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

---

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

*Le livre final (document PDF) contiendra :*

---

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

## Puce gros tiret

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

---

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

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

---

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

*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.

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

## Les Tables

Bien que les tables soient très pratiques pour mettre en forme des données tabulaires, elles sont à utiliser prudemment dans **Prawn-For-Book** car elles ont le tort de ne pas aligner leur contenu sur la grille de référence et perturbent donc l'harmonie des pages.

## Notes de pages

Un parti-pris de **Prawn-For-Book**, que certains regretteront sans doute, est de ne jamais utiliser de notes de bas de page. Si elles sont souvent pratiquées, nous pensons qu'elles offrent une mauvaise expérience à la lectrice ou au lecteur, et que souvent il ne prend pas la peine d'interrompre le flux de sa lecture pour consulter une note.

Nous avons donc pris l'option d'utiliser plutôt les "notes de page" qui se placent plutôt, en général, à la fin du paragraphe ou à la fin du bloc cohérent de paragraphes.

Ou de les reporter toutes à la fin de l'ouvrage.

*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 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.

---

## 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.

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

---

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 01 11 2023.

---

## Export comme livre numérique

Bien que **Prawn-For-Book** soit principalement pensé pour produire un PDF pour l'imprimerie, on peut également s'en servir pour produire un fichier PDF numérique (comme ce manuel par exemple).

Les différences sont les suivantes :

- les liens sont interactifs (quand on clique sur un renvoi, on rejoint ce renvoi),
- les adresses internet externes sont activables,
- **la table des matières est interactive**, il suffit de cliquer sur un titre pour rejoindre la partie.

Pour déterminer l'export en livre numérique, il faut agir sur la donnée `book_format` :  
`book:` `format:` en mettant sa valeur à `pdf`.

*(le format normal est `publishing`)*

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

---

---

`book_format:`

`book:`

`format: pdf`

---









# Annexe

## Définition de la couleur

Dans **Prawn-For-Book** comme dans *Prawn*, la couleur s'exprime de façon hexadécimale(( )).

Une couleur hexadécimale — c'est-à-dire "exprimée en base 16" — est composée de trois paires de chiffres/lettres — par exemple 'AA', 'BB' et 'CC' qui forment 'AABBCC' — chaque paire représente une valeur de 0 à 125 — '00' vaut 0 et 'FF' vaut 125 — en RGB — Red/rouge, Green/vert, Blue/bleu — c'est-à-dire que la première paire indique la quantité de rouge, la seconde la quantité de vert et la troisième la quantité de bleu)

Quand tous les signes sont égaux — par exemple '55555' — la couleur est grise, sauf pour '000000' qui est noir et 'FFFFFF' qui est blanc.

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

---

Un texte avec du <color rgb="000077">bleu</color>.

*Le livre final (document PDF) contiendra :*

---

Un texte avec du [bleu](#).

---

## Marques markdown

Vous trouverez ci-dessous toutes les marques Markdown utilisables.

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

---

La table ci-dessous a été construite aussi en markdown, en utilisant le format :

| CA1 | CA2 | CA3 | CA4 |

| CB1 | CB2 | CB3 | CB4 |

Vous pouvez trouver toutes les informations sur l'utilisation des tables à la page 31.

## Liste exhaustive des fonctionnalités

- pagination automatique et personnalisable
- définition de la fonte par défaut
- justification par défaut des textes
- colorisation des textes
- suppression des veuves et des orphelines
- suppression des lignes de voleur
- nombreuses sortes de puces et puces personnalisées `[[puces]]`
- évaluation à la volée du code ruby (pour des opérations, des constantes, etc.)
- traitement des références croisées
- traitement dynamique des références à d'autres livres
- corrige l'erreur typographique de l'apostrophe droit
- corrige l'erreur typographique de l'absence d'espace avant et après les chevrons
- corrige l'erreur typographique de l'espace avant et après les guillemets droits et courbes
- corrige l'oubli de l'espace avant les ponctuations doubles
- corrige l'erreur d'espace avant les ponctuations doubles (pose d'une insécable)
- corrige l'absence d'espace insécable à l'intérieur des tirets d'exergue
- changement de fonte (police) pour le paragraphe suivant `[[change_fonte_for_next_paragraph]]`
- exportation seulement du texte produit
- exportation comme livre numérique (pur PDF)