

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
Aperçu rapide des commandes	17
Les Forces de <i>Prawn-For-Book</i>	17
Définitions minimales	19
Les Recettes	22
Recette du livre	22
Recette de la collection	22
Définition des fontes (embarquées)	23
Définition de la fonte par défaut	23
Éditeur / Maison d'édition	24
Le Texte du livre	25
Texte au format pseudo markdown	25
Définition des titres	25
Comportement (par défaut)	28
Alignement sur la grille de référence	29
Les pages du livres	31
Introduction	31
Les pages spéciales	31
La page de Faux-titre	31
Page de titre	33
La page de mention légale	35
La page de dédicace	35
Table des matières	38
La page des remerciements	41
Page des crédits (Colophon)	42
Page d'index	45
Aides & Assistants	47
Manuel et autres aides	47
Les « Snippets »	48
Messages d'erreurs et notices	49
Afficher grille de référence et marges	49
Exportation du texte	50

Pagination	51
Les types de pagination	51
Suspension de la pagination	52
Format du livre et des pages	55
Définition des marges	56
Page en multi-colonnes	59
Le Mode Expert	62
Description	63
Indication du contexte d'erreur	63
Injection	63
Évaluation du code ruby	64
Bibliographies en mode expert	66
Mode Multi-colonnes	66
Formateurs	68
Modification de la hauteur de ligne	68
ANNEXE	71
Reconstruction du manuel	72
Marques markdown	72
Le format YAML de la recette	72
Les "Fontes-String"	73
Définition de la couleur	73
Rogner une image SVG	75
Synopsis de création	78
Installation de l'application	78
Constantes	78
Les pages d'un livre	79
Liste exhaustive des fonctionnalités	79

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 88 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

Généralités

Aperçu rapide des commandes

Si vous avez déjà consulté l'aide et produit votre premier livre, vous aurez peut-être simplement besoin d'un rafraichissement de mémoire sur les commandes de base. Ce chapitre les expose.

(note : toutes ces commandes s'effectuent dans une console/un terminal ouvert dans le dossier du livre à considérer)

pfb init	Instanciation d'un nouveau livre/une nouvelle collection.
pfb build	Construction du livre.
pfb build -open	Construction et ouverture du livre.
pfb build -t	Exportation du livre
pfb open	Ouvrir un élément quelconque.
pfb install	Installer le livre (les snippets).
pfb calc	Pour effectuer des calculs (marges, couverture...).

Options de la commande **build**

La commande de construction du livre (**build**) supporte plusieurs options qui affectent son comportement :

-open	Ouvrir le livre une fois construit.
-margins	Construit le livre avec les marges visibles.
-grid	Construit le livre avec les lignes de référence.
-debug	Affiche des messages d'erreur plus précis.

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 ne contenant aucune veuve, aucune orpheline et aucune ligne de voleur,
- permet d'utiliser de façon simple les feuilles de styles,
- respecte toutes les règles de typographie en vigueur,
- produit une page de titre valide par défaut,
- permet de travailler sur une collection entière,
- permet de gérer automatiquement une infinité d'index,
- gère facilement les bibliographies,
- gère les références, même à d'autres livres,

- gère les images flottantes,
- produit bien sûr une table des matières valide par défaut,
- extension infinie pour les experts qui connaissent le langage Ruby et la gem Prawn.

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.

Définitions minimales

Cette section se concentre sur les définitions minimales que vous aurez à faire si les définitions par défaut ne conviennent pas au livre que vous êtes en train d'écrire. Typiquement, il est fort possible que la taille du livre ainsi que les marges ne soient pas définies par défaut comme votre livre le demande.

C'est ici que vous apprendrez à régler ces valeurs.

Cette partie suppose que vous avez déjà parcouru cette aide et que vous connaissez, notamment, l'usage des recettes utilisées par **Prawn-For-Book** (voir « recette du livre », page 24 et « recette de la collection », page 24). Dans le cas contraire, reportez-vous à ces sections avant de lire celle-ci (qui se trouve avant juste pour pouvoir l'atteindre plus rapidement lorsque l'on connaît déjà l'application et que l'on a juste besoin de se rafraîchir la mémoire.

Définition des titre et auteur

La première définition concernant le livre sera certainement son titre et son auteur(e) que nous ne pouvons pas définir par défaut. Pour ce faire, nous définissons les propriétés **title** (« titre » en anglais) et **author** (« auteur » en anglais) dans la section **book_data** (« données du livre » en anglais) de la recette du livre `recipe.yaml`.

```
#./recipe.yaml
book_data:
  title: "Le titre du livre"
  author: "Prénom NOM, Autre AUTEUR"
```

Notez que les noms se mettent avec capitales, pour que **Prawn-For-Book** puisse les reconnaître. On les sépare également par des virgules (signe qui, normalement, ne se trouve dans aucun patronyme).

Taille du livre

La taille du livre doit être définie dans la recette du livre si elle est unique même dans une collection ou dans la recette de la collection si elle est commune à tous les livres de cette collection (ce qui est généralement le cas).

On doit simplement définir les propriétés **width** (« largeur » en anglais) et **height** (« hauteur » en anglais) dans la section **book_format** (« format du livre » en anglais) :

```
#./recipe.yaml ou ../recipe_collection.yaml
book_format:
  width: "<quantité><unité>"    # p.e. "210mm"
  height: "<quantité><unité>"    # p.e. "29.7cm"
```

Marges du livre

Les marges du livre relèvent aussi du format du livre (`book_format`) et plus précisément du format des pages. On les indique dans la partie **margins** (« marges » en anglais) de cette partie.

```
#./recipe.yaml ou ../recipe_collection.yaml
book_format:
  page:
    margins:
      top: "<quant><unité>"          # marge en haut
      bottom: "<quant><unité>"        # marge en bas
      ext: "<quant><unité>"          # marge extérieure
      int: "<quant><unité>"          # marge intérieure
```

À titre de rappel, dans un livre, on ne définit pas les marges gauche et droite, entendu qu'elles sont différentes suivant qu'on se trouve sur une page gauche ou une page droite. On définit à la place les marges intérieures (int) et les marges extérieures (ext).

Vous pouvez trouver tout ce qui concerne les marges dans la section « Définition des marges » (p. 58).

Polices embarquées dans le livre

Il y a fort à penser que la police par défaut de **Prawn-For-Book**, pour les textes et les titres, ne vous conviennent pas tout à fait. Il convient de la définir.

Cette définition se fait en deux étapes : d'abord, il faut **informer Prawn-For-Book des polices que vous voulez utiliser**, si elles ne font pas partie des polices par défaut, et ensuite indiquer **quelle police sera utilisée** pour tel ou tel élément du texte (texte général, titre de niveau 1, de niveau 2, etc.).

Pour la définition des polices, nous vous renvoyons à la section « Définition des fontes (embarquées) » (p. 25) qui est parfaitement détaillée.

Brièvement, voilà ce qu'il faut définir quand on veut changer un minimum de choses :

```
#./recipe.yaml ou ../recipe_collection.yaml
fonts:
  # Définition d'une fonte à embarquer
  <nom utilisé pour la fonte>:
    <style un>: "path/to/fichier/ttf"
    <style deux>: "path/to/autrefichier/ttf"
    etc.
```

Par exemple :

```
#./recipe.yaml ou ../recipe_collection.yaml
fonts:
  TexteDefault:
    normal: "fontes/MaFonte/regular.ttf"
    italic: "fontes/MaFonte/italic.ttf"
    bold: "fontes/MaFonte/gras.ttf"
    bold_italic: "fontes/MaFonte/gras-italic.ttf"
```

Noter, ci-dessus, les chemins relatifs. Cela signifie que les fontes seront recherchées dans un dossier fontes dans le dossier du livre (ou le dossier de la collection si c'est une collection).

Le nom défini (ci-dessus « TexteDefault ») sera le nom repris soit dans la définition des éléments (cf. ci-dessous) soit dans le texte, lorsqu'on voudra appliquer une fonte à du texte.

Police appliquée au texte normal

Le premier élément à définir, au niveau de la police à utiliser est bien évidemment le texte par défaut. Ça relève du texte (`text`) dans le format du livre (`book_format`). On définit simplement :

```
book_format:
  text:
    font: "TexteDefault/normal/11/222222"
```

C'est une définition qu'on appelle « fonte string » (prononcer « strine'gue »). Voir « Les "Fontes-String" » (p. 75). Noter qu'on utilise la police embarquée plus haut.

Fontes appliquées aux titres

On peut définir de la même manière les polices/styles/taille/couleur des niveaux de titre dont on a besoin. On le fait toujours dans `book_format`, puisque ça relève du format du livre, dans la section `titles` (« titres » en anglais).

```
book_format:
  titles:
    # Fonte générale qui servira pour tous les titres
    font: "Numito/regular//555555"
    level1:
      font: "/regular/20"
    level2:
      font: "//16"
    level3:
      font: "/italic/14"
    etc.
```

Noter que les « fontes-string » n'ont jamais besoin de définir les quatre données fonte, style, taille et couleur. Une donnée non définie prendra toujours la valeur par défaut. Pour les titres, ci-dessus, par exemple, la police sera toujours *Numito*, police définie de façon générale pour les titres. De même que la couleur sera toujours 555555, donc un noir un peu clair.

Hauteur de ligne

La *hauteur de ligne* (`line_height` en anglais) est une valeur très importante puisqu'elle va déterminer le positionnement de toutes les lignes de texte, entendu que, comme cela est expliqué dans la section « Alignement sur la grille de référence » (p. 31).

Elle relève du format du livre donc se trouve dans `book_format`, et relève précisément du texte (`text`). On la définit donc ainsi :

```
book_format:
  text:
    line_height: <valeur en point-postscript>
```

Cette valeur dépendra bien entendu de la taille de fonte utilisée par défaut dans le livre (cf. plus haut).

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. « Le format YAML de la recette » (p. 74)). 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 « Le format YAML de la recette » (p. 74)). Vous pouvez trouver ci-dessous les données propres à une collection.

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

```
---
# (données de la collection)
collection:
```

```
name: "Nom de la collection"
short_name: "Nom raccourci (pour les messages)"
#/collection
# ...
```

Définition des fontes (embarquées)

Bien que l'application propose, clé en main, des fontes à utiliser pour imprimer son livre, on peut définir n'importe quelle fonte dont on posséderait la licence commerciale pour être utilisée pour tout élément du livre (texte, titres, etc.). Mais afin que l'imprimeur puisse s'en servir, il faut l'embarquer dans son document PDF à destination de cet imprimeur.

Conseil concernant l'emplacement des fontes

Il vaut mieux faire un dossier `fontes` dans votre dossier de collection ou de livre et y dupliquer les polices que vous voulez utiliser. De cette manière, si vous avez à transmettre le dossier à un collaborateur (ou autre), celui-ci ou celle-ci pourra imprimer correctement le livre, avec les fontes voulues (qu'il ou elle devra charger dans ses fontes personnelles).

Styles obligatoires pour les fontes

Si vous voulez utiliser la mise en forme HTML utilisant des `ou des` pour les italiques, des ***ou des*** pour les gras, ou les deux en même temps, il est obligatoire de définir les styles *italic*, **bold** et ***bold_italic***.

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 rue Code Ville>"
  contact: "<contact@chez.lui>"
  url: "https://<url/site>"
  logo_path: "chemin/vers/logo"
```

Le Texte du livre

Texte au format pseudo markdown

Le fichier texte du livre porte le nom « **texte.pfb.md** ».

Ce nom obligatoire permet d'utiliser l'application *Sublime Text* (<https://www.sublimetext.com>) pour rédiger son texte avec de nombreuses aides qui simplifient considérablement la tâche et permettent de se concentrer sur le contenu du livre. Notez cependant, si vous connaissez le format Markdown (ou *Multimarkdown*), que ça n'est pas du pur format Markdown et que certaines fonctionnalités ne sont pas prises en compte. Vous trouverez en annexe, page all_markdown (page 74), l'intégralité des marqueurs utilisables.

Définition des titres

Le format de base du texte étant le format markdown, on marque un titre dans le texte simplement grâce à des dièses suivi du titre (1 dièse => titre de niveau 1, 2 dièses => titre de niveau 2, etc.)

Le format de ces titres sont définis par défaut pour s'afficher harmonieusement avec le texte, mais on peut en définir très précisément l'aspect dans la recette, grâce à la données `titles` (« titres » en anglais) dans le format du livre (donnée `book_format` — « format du livre » en anglais)

```
# ./recipe.yaml ou ../recipe_collection.yaml
book_format:
  titles:
    # ... définition des titres
```

On peut commencer par y définir une fonte-string¹ qui s'appliquera à tous les niveaux de titre, pour définir par exemple la police spéciale à utiliser (mais chaque niveau pourra ensuite définir précisément ses données).

¹ Cf. « *Les "Fontes-String"* » (p. 75)

Par exemple :

```
book_format:
  titles:
    font: "Helvetica/normal//222222"
```

Noter ci-dessus que la taille (troisième membre de la fonte-string) n'est pas définie.

Ci-dessus, par défaut, tous les titres seront en Helvetica, de style normal, et de couleur noir très légèrement éclaircie (222222).

Niveaux des titres

Dans cette section de la recette (cf. ci-dessus), on définit chaque niveau de titre grâce à la propriété `level` (« niveau » en anglais) suivi du niveau de titre, de 1 à 7 (`level1`, `level2`, `level3`...).

Astuce : si vous n'utilisez que 4 niveaux de titre, vous pouvez utiliser les titres restants (5 à 7) pour formater des titres tout à fait spéciaux.

On trouve donc:

```
book_format:
  titles:
    level1:
      # Données du titre de premier niveau
    level2:
      # Données du titre de second niveau
    level3:
      # Données du titre de troisième niveau
    # etc.
```

Données des niveaux de titres

Chaque niveau de titre peut définir les données/propriétés ci-dessous, à commencer par définir les données de fonte (police).

- **font** | La *fonte-string* à utiliser pour le niveau de titre en question, si elle doit être radicalement différente de la fonte par défaut des titres (cf. ci-dessus). Ou on peut se limiter à définir la taille par `"/ /42/"` mais dans ce cas, peut-être vaudrait-il mieux utiliser la propriété `size` ci-dessous.
- **size** | Pour définir la taille de police, en points.
- **style** | Pour (re)définir le style de la police. Sinon, c'est le style par défaut qui sera utilisé. Ce style, bien entendu, doit être défini dans les fontes du livre ².
- **color** | Pour (re)définir la couleur du titre. Sinon, c'est la couleur par défaut qui sera utilisée.

² Cf. « Définition des fontes (embarquées) » (p. 23).

On peut également indiquer l'emplacement du titre en terme de page :

- **next_page** | Si `true` (« vrai » en anglais), le titre sera placé sur une nouvelle page (paire ou impaire en fonction du flux du texte). Par défaut, seul le titre de premier niveau (`level1`) est placé sur une nouvelle page.
- **belle_page** | Si `true`, le titre sera toujours placé sur une *belle page*, c'est-à-dire une page impaire, à droite. Par défaut, seul le titre de premier niveau est placé sur une *belle page*.
- **alone** | (« seul » en anglais) Si `true` le titre sera seul sur la page (`next_page` sera automatiquement mis à `true`). Le texte qui suit le titre sera placé sur la page suivante. Dans le cas d'un titre seul, la propriété `lines_before` (« lignes avant » en anglais) permet de placer le titre dans la page (plus ou moins haut). Par défaut, seul le titre de premier niveau est placé seul sur une page.

On peut ensuite définir les lignes qu'il faut laisser avant et après le titre grâce aux propriétés :

- **lines_before** | (« lignes avant » en anglais) Nombre de lignes avant le titre.
- **lines_after** | (« lignes après » en anglais) Nombre de lignes après le titre.

Les *lignes* dont il est question ci-dessus sont des *lignes de référence*³, car les textes sont toujours alignés, dans un livre bien formaté, et il en va de la même manière pour les titres.

³ Cf. « *Alignement sur la grille de référence* » (p. 31).

Prawn-For-Book, c'est sa nature, adopte autant que possible un comportement intelligent par rapport à ces définitions. Par exemple, si le titre se trouve en haut de page — et que ça n'est pas un titre seul dans la page (cf. ci-dessus `alone`) — alors il n'ajoutera pas de lignes avant même si elles sont définies.

De la même manière, quand deux titres se suivent, les lignes ne s'additionnent pas (ce qui laisserait des écarts trop importants, disgracieux). C'est la valeur la plus grande qui est appliquée. Par exemple, si le titre précédent définit « 4 lignes après le titre » (`lines_after: 4`) et que le titre suivant définit « 3 lignes avant le titre » (`lines_before: 3`), ce ne sont pas 7 lignes qui sont laissées entre les deux titres, mais 4.

On peut également définir d'autres modifications de détail du titre :

- **align** | (« alignement » en anglais) Cette propriété permet de définir l'alignement du titre. Les valeurs peuvent être `left` (« gauche » en anglais) pour un alignement à gauche, `right` (« droite » en anglais) pour un alignement à droite ou `center` (« centre » en anglais) pour un centrage dans la page.
- **left** | (« gauche » en anglais) À la place de la propriété `align` on peut définir précisément le décalage avec la marge gauche en donnant une valeur numérique (p.e. 40) ou une dimension (p.e. 52mm) à cette propriété.
- **right** | (« droite » en anglais) À la place de la propriété `align` on peut définir précisément le décalage avec la marge droite en donnant une valeur numérique (p.e. 40) ou une dimension (p.e. 52mm) à cette propriété.
- **caps** | Si `true` (« vrai » en anglais), les titres du niveau concerné seront toujours mis en capitales.
- **leading** | (« interlignage » en anglais) Permet de définir l'interlignage, quand le titre tient sur plusieurs lignes et qu'il faut le définir précisément. Cette valeur peut être négative, pour rapprocher les lignes du titre.

Comportement (par défaut)

Dans cette partie sont décrits tous les comportements naturels (ou pour dire « par défaut ») de **Prawn-For-Book** et ce qui fait sa force unique et son utilité.

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 automatiquement alignées sur une ce qu'on appelle une *grille de référence* qui produit un affichage professionnel.

Cette grille de référence ne dépend que du paramètre `line_height` (« hauteur de ligne » en anglais) dans la recette — voir `texte_detail/hauteur_de_ligne` (p.).

Note : pour l'exemple, nous avons demandé l'affichage de la *grille de référence* — voir « Afficher grille de référence et marges » (p. 51) afin de constater par vous-même 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 lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem ipsum ipsum lorem 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 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 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 lorem ipsum ipsum lorem ipsum ipsum
lorem ipsum ipsum lorem ipsum ipsum lorem ipsum

Les pages du livres

Introduction

Un livre est composé de deux grands types de pages :

- les pages de titre et de texte, qui composent le contenu principal de l'ouvrage,
- les pages liminaires et finales, que nous appelons *pages spéciales* dans **Prawn-For-Book** et donc vous allez trouver la définition ici.

Les pages spéciales

Les pages considérées comme *spéciales* dans **Prawn-For-Book** sont les pages suivantes. Elles seront toutes abordées en détail par la suite.

- page de faux-titre,
- page de garde,
- page de titre,
- page de mention légale,
- page de dédicace,
- table des matières,
- page de remerciements,
- page de bibliographies,
- page des index,
- page des crédits (colophon)

La page de Faux-titre

La page de *faux titre* (« half-title page » en anglais, pour information) est souvent la toute première page imprimée du livre. Elle contient le titre seul, parfois accompagné du sous-titre. Elle est parfois précédée d'une page de garde.

Dans notre nomenclature, c'est le vis-à-vis 1 (si pas de page de garde) ou 2 (si page de garde).

Cette page ne contient ni pagination (numéro de page) ni entête.

Page contenant juste le titre et l'auteur, avant la *page de titre* proprement dit. Par défaut, elle n'est pas placée dans un livre produit par **Prawn-For-Book**.

Pour la graver dans le livre, mettre la propriété `faux_titre` (ou `half_title_page`) de la section `inserted_pages` de la recette à `true` ou à une table définissant précisément l'aspect. Ici, encore, comme d'habitude, vous avez le choix entre l'aspect par défaut (impeccable comme ça) avec `true` :

```
inserted_pages:  
  faux_titre: true
```

Ce *faux-titre* par défaut utilise la police par défaut, légèrement augmentée au niveau de la taille, et la place à peu près au tiers de la hauteur de la page.

Si votre livre possède un sous-titre, il sera affiché en dessous du titre, plus petit. Nous verrons que vous pouvez tout à fait supprimer ce sous-titre pour la page de faux-titre.

Réglage de la page de faux-titre

Vous pouvez effectuer un réglage plus fin de cette page :

```
inserted_pages:
  faux_titre:
    title
      font: "police/style/taille/couleur"
      line: 12 # ligne sur laquelle poser le titre
      leading: 0 # interlignage (si tient sur plusieurs lignes)
    # Pour ne pas afficher le sous-titre, décommentez simplement
    # la ligne suivante
    # subtitle: false
    # Dans le cas contraire, vous pouvez régler les propriétés
    # de la page de sous-titre
    subtitle:
      font: ...
      line: ...
      leading: ...
    author:
      font: "police/style/taille/couleur"
      line: 16 # ligne sur laquelle poser l'auteur
      leading: 0 # interlignage (si plusieurs auteurs, qui
                  # tiennent sur plusieurs lignes)
```

Notez que partout où les lignes ne sont pas définies, **Prawn-For-Book** gardera les valeurs par défaut, ce qui permet d'ajuster facilement un seul élément.

Ajouter -grid en construisant le livre pour voir les lignes

Toutes les valeurs des fontes-strings non définies sont remplacées par les valeurs par défaut

La page de garde

Page de garde est un terme qui trouve dans le langage plusieurs acceptions très différentes et il y a souvent confusion pour savoir de quoi l'on parle. Elle peut être :

- la couverture d'un rapport (de stage, par exemple, ou de thèse),
- la toute première page du document, qui sera *collée* au verso de la première de couverture (c'est donc la 2^e de couverture),
- un page (double page, en fait) située juste avant la *page de titre*, donc entre la (double-)page de *faux-titre* et la (double-)page de *titre*.

Dans **Prawn-For-Book**, elle concerne la troisième acception. Si la page de *faux-titre* est gravée, cette page de garde sera une double page avant la double page de titre, qui empêchera de voir la page de titre par transparence (cela fait donc 3 pages ajoutées — consultez l'annexe, annexes/pages_dun_livre (p.), pour bien comprendre).

Par défaut, cette *page de garde* n'est pas imprimée dans un livre produit par **Prawn-For-Book**. Pour qu'elle le soit, il suffit de le préciser dans la recette du livre ou de la collection en mettant sa valeur à `true` (« vrai » en anglais) :

```
(( line ))
~~~yaml
inserted_pages:
page_de_garde: true
~~~
```

Page de titre

La *page de titre* présente un grand nombre d'informations au lecteur et notamment le titre, l'auteur et l'éditeur du livre.

Ses données sont définies dans la section `inserted_pages` de la recette du livre ou de la collection (cf. « Les Recettes » (p. 21)) grâce à la propriété `page_de_titre` (ou `title_page`).

Dans sa version la plus simple, utilisant les valeurs par défaut, on aura juste à indiquer :

```
inserted_pages:
  page_de_titre: true
```

... pour que la page de titre soit affichée en début de livre.
(mettre `false` — "faux" — au lieu de `true` — "vrai" — pour que la page de titre ne soit pas affichée)

Mais comme tout autre élément de **Prawn-For-Book**, on peut définir la page de titre de façon plus précise. Trouvez ci-dessous les propriétés définissables et retrouvez-les dans l'exemple de recette plus bas.

Vous pourrez aussi trouver, tout en bas de cette section, la recette utilisée pour produire la page de titre un peu trop bariolée (pour cette illustration) de ce manuel, dans les premières pages.

Propriétés de la *page de titre*

Ci-dessous, les sous-propriétés `font` sont des « font-string » (cf. « Les "Fontes-String" » (p. 75)) et les propriétés `line` sont les lignes de référence sur lesquelles doivent être imprimés les éléments. Pour les voir sur les pages, vous pouvez demander la construction du livre avec l'option `-grid` qui les fera apparaître sur les pages.

Noter qu'une propriété ne sera imprimée que si elle est définie dans cette section `inserted_pages: page_de_titre:`.

- **title** | Définition de l'aspect du *titre du livre* sur la page de titre. Définit les propriétés `font` et `line` (ligne sur laquelle imprimer le titre).
- **subtitle** | Définition de l'aspect du *sous-titre du livre* s'il en possède un. Définit les propriétés `font` et `line`.
- **author** | Définit l'aspect de l'affichage de l'*auteur* (ou des auteurs) par les propriétés `font` et `line` (ligne sur laquelle imprimer l'auteur)
- **collection** | Définit l'aspect du titre de la collection à l'aide des propriétés `font` et `line`.

- **publisher** | Définit l'aspect de la maison d'édition ou de l'éditeur (« publisher » = éditeur). En plus des propriétés `font` et `line`, on trouve la propriété `logo` qui doit définir la taille en hauteur (`height`) et la ligne (`line`) du logo. On peut trouver aussi la propriété `path` qui conduit au logo si c'est une autre image que celle définie dans la recette (cf. « Éditeur / Maison d'édition » (p. 24)).

Choix des lignes pour la page de titre

Les placements des éléments de page de titre sont calculés pour que la page soit la plus harmonieuse possible. Cependant, si l'on veut définir soi-même les lignes à utiliser (propriété `line`), on peut demander une première fois de construire le livre en affichant les *lignes de référence* grâce à l'option `-grid` (`pdf build -grid`).

Il suffira alors de relever sur quelle ligne on veut voir gravé tel ou tel élément et de l'indiquer dans la recette pour chaque élément modifié.

Propriétés du fichier recette

```
---
inserted_pages:
  page_de_titre:
    title:
      # Police du titre (toute valeur non définie prendra
      # la valeur par défaut)
      font: "<fonte>/<style>/<taille>/<couleur>"
      line: 4
      leading: 0.7
    subtitle:
      # idem
    author:
      # idem
    collection:
      # idem
    publisher:
      # idem
    logo:
      # S'il faut prendre un autre logo que le logo
      # officiel, on définit son chemin ici
      path: "chemin/vers/autre/logo"
      height: <hauteur>
      line: <ligne>
```

Extrait de la recette actuelle

```
---
# ...
inserted_pages:
  page_de_garde: true
  faux_titre: false
  page_de_titre:
    title:
      font: Numito/normal/30/FF0000
```

```
subtitle:
  font: Numito/italic/16/007700
author:
  font: Numito/normal/18/000077
collection:
  font: Times-Roman//12/000000
publisher:
  font: Numito/normal/12/FF8000
logo:
  height: 40
copyright: "@Tous droits réservés"
credits_page:
  disposition: distribute
```

La page de mention légale

C'est la page du fameux *copyright*. Cette page se trouve après la page de titre. Si le copyright est défini dans la recette, cette page est automatiquement créée sans que vous ayez rien d'autre à faire.

Le copyright doit être défini dans :

```
inserted_pages:
  copyright: |
    @Philippe Perret, 2023-2023
    Le Code de la propriété intellectuelle interdit toute
    reproduction et toute utilisation etc.
```

Le trait droit (« | ») après le `copyright` : indique à YAML comment doit être lu le contenu qui commence sur la ligne suivante. Avec ce trait droit, un double retour de chariot sera transformé en un retour à la ligne tant qu'un retour de chariot simple (comme ci-dessus entre « toute » et « reproduction ») sera remplacé en une espace typographique.

La page de dédicace

Sur cette page est adressée une dédicace.

Comme « La page de mention légale » (p. 35), il suffit que la propriété `dedicace` soit définie dans `inserted_pages` pour que cette page soit insérée dans le livre, correctement formatée et positionnée.

```
inserted_pages:
  dedicace: |
    Je dédis cet ouvrage à tous les chiens malheureux de
    la terre, afin que leur souffrance s'abrège.
```

On peut également la définir très précisément avec :

```

inserted_pages:
  dedicace:
    content: |
      Je dédis cet ouvrage à tous les chiens malheureux de
      la terre, afin que leur souffrance s'abrège.
    font: "<police>/<style>/<taille>/<couleur>"
    line: <numéro de ligne>
    leading: <interlignage>

```

La propriété font est une *fonte-string* dont vous pouvez tout apprendre à la page 75 dans l'annexe.

Dédicace sur plusieurs pages

Une dédicace peut être très longue et tenir sur plusieurs pages. Dans ce cas, il est peut-être bon de ne pas la définir dans la recette mais plutôt dans le livre. Noter qu'une dédicace sur plusieurs pages doit être numérotée donc qu'il n'y a rien à faire par rapport à ça.

Indiquons juste qu'on peut choisir facilement la première ligne sur laquelle doit commencer la dédicace grâce à la commande `move_to_line` (« se déplacer à la ligne » en anglais).

Une dédicace sur plusieurs pages pourra donc ressembler à ça ce qui suit.

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

```

(( new_page ))
(( move_to_line(12) ))
(( {style: :italic, size: 14, font: 'Courier', indentation: 40} ))
Je voudrais, par ces quelques lignes, adresser une dédicade à celle sans qui ce manuel
n'aurait jamais vu le jour, je veux parler de ma chienne Poupette qui reste pour moi bla bla
bla sur plusieurs pages.
(( new_page ))

```

Le livre final (document PDF) contiendra :

*Je voudrais, par ces quelques lignes, adresser une
dédicade à celle sans qui ce manuel n'aurait jamais vu le
jour, je veux parler de ma chienne Poupette qui reste
pour moi bla bla bla sur plusieurs pages.*

Table des matières

Page 11, dans notre comptage intégral, on imprime la table des matières à l'endroit voulu à l'aide de la marque :

`((tdm))` pour "T"able "D"es "M"atières

ou :

`((toc))` pour « T"able "Of » "C"ontent (« table des matières » en anglais).

Noter que la table des matières, sauf indication contraire, se placera toujours sur une *belle page*, c'est-à-dire sur une page de droite.

La *table des matières*, contrairement aux autres pages d'un livre **Prawn-For-Book**, va nécessiter une attention particulière, en tout cas lorsqu'elle tient sur plusieurs pages.

{{À REPRENDRE PAR RAPPORT À LA NOUVELLE APPROCHE}}

On peut cependant empêcher ce comportement en mettant la propriété `not_on_even` (« pas sur la paire » en anglais) à `true` dans la recette.

Table des matières en début d'ouvrage

Si on inscrit la table des matières en début d'ouvrage, il faut calculer le nombre de pages qu'elle va occuper et le définir dans la propriété `page_count` de la section `table_of_content` (« table des matières » en anglais). Ce nombre doit être pair et il vaut 2 par défaut.

Dans la pratique, estimez grossièrement la longueur de la table des matières en fonction de la taille de l'ouvrage et du nombre de titres et ajoutez 2 pour garder une certaine latitude. Ajustez le nombre final une fois l'ouvrage presque achevé pour supprimer les pages superflues (ou ajoutez-en de nouvelles si la table des matières « mange » sur le texte).

Table des matières en fin d'ouvrage

Si on inscrit la table des matières à la fin de l'ouvrage, il n'y a aucune précaution à prendre concernant le nombre de pages qu'elle couvrira.

Pour ce manuel, dont on trouve deux tables des matières, une en début d'ouvrage et l'autre en fin d'ouvrage. Seule la table des matières en début d'ouvrage a dû faire l'objet d'un calcul du nombre de pages.

Aspect de la table des matières

Comme les autres réglages, on peut définir précisément la table des matières dans |« Les Recettes » (p. 21), dans la partie `table_of_content` (qui signifie « table des matières » en anglais).

On trouve ces propriétés :

- **pages_count** | Définit le nombre de pages réservées par la table des matières. Ce nombre doit impérativement être pair pour conserver l'agencement des *belles pages* (pages impaires) et des *fausses pages* (pages paires) dans le livre. Par défaut, on compte 2 pages pour la table des matières. Cette valeur vaut 2 par défaut. Noter que les pages supplémentaires (il y en aura toujours deux) peuvent aussi être ajoutées explicitement dans le livre par des `((new_page))`.

- **level_max** | Niveau maximum. Le niveau de titre qui sera affiché dans la table des matières. Par défaut, il est à 3, ce qui signifie que les titres jusqu'à 3 dièses seront affichés dans la table des matières (sauf exclusion).
- **title** | Le grand titre à utiliser pour la table des matières. Par défaut, en français, c'est « Table des matières ».
- **title_level**. Le niveau de titre pour ce grand titre. Par défaut, c'est 1.
- **no_title**. S'il ne faut pas imprimer de grand titre « Table des matières » (ou autre valeur de **title**), alors il faut mettre cette propriété à **true** (vrai).
- **font**. Pour la fonte générale (*fonte string* (p. 75))
- **number_font**. Fonte à utiliser pour les numéros de page (*fonte string* (p. 75)).
- **number_size**. Si on ne veut changer que la taille du numéro (pas toute la fonte), on peut utiliser cette propriété.
- **numeroter**. Si **false** (faux), on ne numérote pas la table des matières (éviter).
- **lines_top** | Définit le nombre de lignes au-dessus du premier titre de chaque page de table des matières.
- **lines_bottom**. Nombre de lignes au-dessous du dernier titre de chaque page de table des matières.
- **line_height**. Hauteur de la ligne (entre chaque titre). Attention de ne pas donner une valeur plus petite que la taille des titres, sinon ils n'apparaîtront pas.
- **not_on_even** | Si **true** (vrai), on n'impose pas de commencer la table des matières sur une *fausse page* (à gauche). **false** par défaut.
- **vadjust_number**. Nombre de points-post-scripts pour ajuster verticalement le numéro de la page en face du titre (un nombre positif fait descendre le numéro, un nombre négatif le fait monter),
- **vadjust_line**. Nombre de points-ps pour ajuster verticalement la ligne pointillée d'alignement entre le titre et le numéro de page,
- **dash_line**. Les *experts* peuvent modifier la ligne pointillée en jouant sur cette propriété. Cf. plus bas, à propos de la ligne d'alignement.
- **levelX** où X va de 1 au niveau maximum de titre défini par **level_max**. Donc, par défaut, **level1**, **level2** et **level3**

Chaque niveau de titre peut définir :

- **font**. Sa fonte/style/taille/couleur,
- **size** ou seulement sa taille, en gardant la police par défaut de la table des matières,
- **indent**. Indentation du titre, donc son décalage à droite par rapport à la marge gauche.
- **number_size**. La taille du numéro (mais pour un meilleur aspect, il vaut mieux garder la taille générale).
- **caps**. La modification (casse) du titre. Les valeurs possibles sont **all-caps** (tout en majuscule), **all-min** (tout en minuscule), **title** (titre normal, en fonction de la langue du livre) ou **none** pour le laisser tel quel.
- **number_font**. La police fonte/style/taille/couleur à utiliser pour les numéros de ce niveau de titre (mais il vaut mieux une grande cohérence avec l'ensemble et éviter de la définir).

Exclusion de titres dans la table des matières

On peut exclure de la table des matières des titres du niveau voulu (**level_max**) en ajoutant **{no_toc}** à leur titre, soit à la fin soit après les dièses. Par exemple :

```
### {no_toc} Titre hors TdM
```

ou :

```
## Titre hors TdM {no_toc}
```

Vous pouvez vérifier que le titre dans le texte en exemple ci-dessous ne sera pas imprimé dans la table des matières.

(rappel : « toc » signifie « table of content », c'est-à-dire « table des matières » en anglais — on peut aussi utiliser `{no_tdm}`)

Numérotation de la table des matières

Le « numéro de page » utilisé dans la table des matières dépend directement de la pagination utilisée pour le livre. C'est le numéro de page seule si la pagination est à pages, c'est le numéro de paragraphe si la pagination est à paragraphes et c'est à page et paragraphe si la pagination est hybrid.

Ligne d'alignement de la table des matières

La *ligne d'alignement* est une ligne, souvent pointillée, qui relie le titre (à gauche) au numéro de page (à droite), et permet de mieux faire correspondre visuellement titre et numéro de page.

Elle est définie dans la recette à l'aide de la propriété `dash_line`.

`dash_line` est une table définissant `{:length, :space, :color}` où `:length` est la longueur du tiret (1 pour un point), `:space` est l'espace entre deux tirets et `:color` est la couleur hexadécimale du trait.

Astuce : on peut obtenir une ligne très fine, très discrète, en mettant `dash_line` à `{length:1, space:0, color:"DDDDDD"}`. Noter que c'est le `space: 0` qui, en ne laissant aucun espace entre les points, génère une ligne.

Propriétés qu'on peut trouver dans la recette

```
---
...
table_of_content:
  # Voir ci-dessus les explications de chaque propriété
  # - Titre -
  title: "Table des matières"
  title_level: 1
  no_title: false
  # - Contenu -
  level_max: 3
  numeroter: true
  # - Aspect -
  lines_top: 4
  lines_bottom: 4
  font: "<font>/<style>/<size>/<color>"
  number_font: "<font>/<style>/<size>/<color>"
  number_size: 10
  line_height: 14
  vadjust_line: 0      # ligne pointillée
  vadjust_number: 0    # numéro page
  dash_line: {length: 1, space: 3}
  # - Aspect des titres par niveau -
  level1:
    font: "<font>/<style>/<size>/<color>"
    size: 12
```



```
number_size: null # => même que défaut
caps: all-caps
indent: 0
level2:
  ...
  indent: 1cm
level3:
  ...
  indent: 15mm
```

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

{no_toc} TITRE HORS TDM

Vous pouvez vérifier que ce titre ne sera placé ni dans la table des matières de début de livre ni dans celle de fin de livre.

Le livre final (document PDF) contiendra :

TITRE HORS TDM

Vous pouvez vérifier que ce titre ne sera placé ni dans la table des matières de début de livre ni dans celle de fin de livre.

La page des remerciements

C'est toujours bien de remercier, avant le texte, les gens qui vous ont épaulé dans la conception de votre projet de livre.

La page des remerciements (qui peut tenir sur plusieurs pages), se trouve toujours sur une *fausse page*, donc une page paire, à gauche, en face de l'avant-propos ou de la préface s'ils existent.

Elle est créée par vous, dans le texte du livre, juste après la marque de la table des matières — cf. « Table des matières » (p. 38).

Pour s'assurer de bien se trouver sur une page paire, plutôt que la marque ((new_page)), on peut utiliser la marque ((fausse_page)) ou ((even_page)).

Si elle tient sur une seule page, il ne faut pas la numéroter. Avec d'inscrire le ((even_page)) il faut donc penser à ajouter un ((stop_pagination)) et penser à redémarrer cette pagination à la fin.

Une page de remerciements pourra donc ressembler à :

```
(( stop_pagination ))
(( even_page ))
Je voudrais remercier...
... ma chienne Poupette
... ma femme Michèle
(( restart_pagination ))
```

```
(( new_page ))
```

Notez que si vous êtes absolument certain que cette page de remerciements ne tiendra que sur une seule page, vous pouvez simplement utiliser la marque `((no_pagination))` après être passé sur la bonne page (`no_pagination` concerne la page sur laquelle on se trouve et seulement la page sur laquelle on se trouve — cf. « Pagination » (p. 52)).

```
(( even_page ))
(( no_pagination ))
Je voudrais remercier...
... ma chienne Poupette
... ma femme Michèle
(( new_page ))
```

Page des crédits (Colophon)

On appelle *page des crédits*, *colophon* ou même *achevé d'imprimé*, la page placée à la fin du livre qui donne toutes les informations sur le livre, aussi bien au niveau de l'éditeur ou la maison d'éditions (son nom, son adresse, son email) qu'au niveau de toutes les personnes qui ont contribué à conception du livre, rédacteur ou rédactrice, metteur ou metteuse en page, graphiste, concepteur, directeur ou directrice de collection, en passant par d'autres informations comme le numéro ISBN du livre et sa date de parution ou les remerciements. C'est une sorte de générique de film... pour un livre.

Si vous êtes de niveau *Expert*, vous pouvez bien entendu mettre cette page en forme de façon tout à fait personnalisée dans le détail, en faisant appel aux éléments de la recette (pour rappel, dans les modules ruby, cela s'obtient en faisant appel à la méthode `book.recipe` qui permet d'atteindre toutes les valeurs de la recette, et donc les informations de cette page.

Mais sans être *expert*, on peut obtenir une mise en page tout à fait correcte et professionnelle ici encore grâce aux comportements par défaut. Sans rien préciser d'autre que les informations minimales requises pour le livre ou la collection, la page des infos sera placée à la fin du livre si `pages_speciales: credits_page:` est à `true` dans la recette.

Impression de la page de crédits

La première chose à faire pour s'assurer que la page de crédits sera imprimée est de mettre la valeur `inserted_pages: credits_page:` à `true` (vrai).

Ensuite, il faut s'assurer que les valeurs minimales soient fournies. Vous pouvez lancer une première fois la construction du livre pour connaître les informations manquantes.

Informations requises pour la page de crédits

Toutes les informations requises pour la page des informations se trouvent dans les données `book_making` (« fabrication du livre » en anglais) et `credits_page` de la recette du livre et/ou de la collection.

Toutes les données de `book_making` — le ou les concepteurs du livre, le ou les rédacteurs, le ou les metteurs en page, graphiste, concepteurs de la couverture ou correcteurs — comprennent deux informations :

- **patro** | Pour le patronyme, seul ou une liste. Le patronyme s'écrit toujours avec la même convention : le prénom en minuscule avec capitale au début, le nom tout en capitales. S'il y en a plusieurs, on les met entre crochets (cf. l'exemple ci-dessous). Par défaut, les noms seront corrigés pour être corrects au niveau de la typographie (« Philippe PERRET » dans la donnée recette s'affichera « Philippe Perret »). Pour que le nom reste identique, il suffit de le faire précéder d'un signe égal).
- **mail** | Le mail du **patro** ci-dessus. S'il y a plusieurs personnes, on indique les mails dans le même ordre, entre crochets (comme ci-dessous).

La page des crédits contient aussi les informations sur l'éditeur ou la maison d'édition, qui doivent se trouver dans la section `publisher` de la recette. La seule information requise est le nom de l'éditeur (`publisher: name:`).

Vous trouverez ci-dessous, dans l'exemple de recette, toutes les informations utilisables dans la page de crédits.

Disposition de tous les crédits

Les crédits du colophon peuvent se placer en haut, en bas ou au milieu de la page. On définit cette position grâce à la propriété `credits_page: disposition:` qui peut prendre les valeurs "distribute" (ou "middle", milieu), "top" (haut de page) ou "bottom" (bas de page).

La valeur par défaut des "distribute".

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

```
---
inserted_pages:
  credits_page: true # pour qu'elle soit imprimée
book_data:
  isbn: ...
  url: "https://url/du/livre"
book_making:
  conception:
    patro: "Prenom NOM"
    mail: "mail@chez.lui"
  writing:
    patro: "Prenom NOM"
    mail: "mail@chez.lui"
  page_design: # Mise en page
    patro: ["Prénom NOM", "Prénom NOM"]
    mail: ["premier@chez.lui", "deuxieme@chez.lui"]
  cover: # Couverture
    patro: "=MM & PP" # '=' => non transformé
    mail: null
  correction:
    patro: ...
    mail: ...
```

```

  acknowledgements: # remerciements
    patro: ...
publisher:
  name: "<nom éditeur ou édition>"
  address: <Rue et numéronLieu ditnVille et code postal
  url: "<url du site internet>"
  mail: "<mail@editeur.fr>"
  contact: "<mail contact>"
  siret: "<numéro siret>"
credits_page:
  disposition: "distribute" # ou "bottom", "top"
  label:
    font: "fonte/style/taille/couleur"
  value:
    font: "fonte/style/taille/couleur"
  printing:
    name: "à la demande" # p.e. pour KDP
    lieu: null

```

Extrait de la recette actuelle

```

---
# ...
book_making:
  conception:
    patro: Philippe PERRET
    mail: philippe.perret@icare-editions.fr
    site:
  editor_in_chief:
    patro:
    mail:
    site:
  writing:
    patro:
    mail:
    site:
  page_design:
    patro: Prawn-For-Book
    mail:
    site:
  cover:
    patro: "=MM & PP"
    mail:
    site:
  correction:
    patro: Marion MICHEL
    mail:
    site:
  printing:
    name: Prawn-For-Book
    lieu:
credits_page:

```

```

inserted_pages:
  page_de_garde: true
  faux_titre: false
  page_de_titre:
    title:
      font: Numito/normal/30/FF0000
    subtitle:
      font: Numito/italic/16/007700
  author:
    font: Numito/normal/18/000077
  collection:
    font: Times-Roman//12/000000
  publisher:
    font: Numito/normal/12/FF8000
  logo:
    height: 40
  copyright: "@Tous droits réservés"
  credits_page:
    disposition: distribute

```

Page d'index

Si l'on peut faire autant d'index que l'on désire avec **Prawn-For-Book** (cf. ?? ?), il n'en reste pas moins qu'il existe une page d'index à proprement parler, comme il en existe dans de nombreux livres.

Ajout d'un mot dans la page d'index

Pour qu'un mot soit dans la page d'index, il suffit qu'il soit entouré de `index(<le mot>)`. Ce mot devient alors la clé de l'index ainsi que son *nom humain*. Nous verrons plus tard qu'il sera possible de le redéfinir.

Si un mot doit être affiché différemment que sa clé, il suffit d'ajouter sa clé dans la parenthèse, après un trait droit. Typiquement, cela survient lorsque le mot est composé et qu'il doit être employé au pluriel. Par exemple, nous voulons avoir « chef-d'œuvre » comme mot indexé. Nous pouvons alors utiliser :

C'est un `index(chef-d'œuvre)` et ce sont des `index(chefs-d'œuvre|chef-d'œuvre)`.

Bien entendu, la deuxième fois, il faut s'assurer que la clé soit bien la même.

Utilisation d'une clé virtuelle pour les index

Afin de simplifier la clé des index — "chef-d'œuvre" est une clé un peu difficile, comme « Comité International Olympique », et elles sont *dangereuses* dans le sens où l'erreur de frappe peut vite survenir... — on peut utiliser des clés virtuelles.

Par exemple, nous voulons utiliser « cio » au lieu de « Comité International Olympique ». Nous ferons alors :

Dans le Comité Olympique – qui s'appelle en vérité le Comité International Olympique, il y a plusieurs membres.

Pour que la clé fonctionne, nous devons définir dans un fichier ruby (`prawn4book.rb`, `formater.rb`, etc.) la table INDEX qui va définir le rapport entre la clé et le texte qui devra être gravé dans la page d'index. Elle se définit de la manière suivante :

```
module Prawn4book
  INDEX = {
    cio: "Comité International Olympique",
    <clé>: "<mot dans index>",
  }
end
```

Les clés de cette table doivent impérativement être en minuscule, même si la clé utilisée dans le texte ne l'est pas.

Format de la page d'index

Dans la recette (du livre ou de la collection) on peut définir l'aspect de l'index. Cela revient à définir l'aspect du *canon* (le mot indexé) et l'aspect des numéros de page (ou de paragraphe suivant le type de pagination).

Voyez ci-dessous, dans l'exemple de recette, les propriétés définissables.

Dans la recette

```
page_index:
  aspect:
    canon:
      font: "police/style/taille/couleur"
    number:
      font: "police/style/taille/couleur"
```

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

`index(le mot)` sera indexé sous la clé "le mot" tandis que `index(les mots|le mot)` sera également indexé sous la clé "le mot" même si c'est un mot différent. On peut d'ailleurs mettre `index(ce que l'on veut|le mot)` à partir du moment où la clé est définie.

Le livre final (document PDF) contiendra :

le mot sera indexé sous la clé « le mot » tandis que les mots sera également indexé sous la clé « le mot » même si c'est un mot différent. On peut d'ailleurs mettre ce que l'on veut à partir du moment où la clé est définie.

Aides & Assistants

Manuel et autres aides

Tout au long de la conception de son livre — et sa production — on peut obtenir de l'aide sur **Prawn-For-Book** de plusieurs façon.

La façon la plus complète consiste à ouvrir ce *manuel autoproduit* qui contient l'intégralité des fonctionnalités de **Prawn-For-Book** expliquées de façon détaillée. C'est incontestablement **la bible de l'application**. Pour l'ouvrir, il suffit de jouer :

```
> pfb manuel
```

On peut obtenir une aide beaucoup plus succincte, rappelant les commandes de base, en jouant au choix l'une de ces commandes :

```
> pfb
> pfb aide
> pfb -h
> pfb --help
```

Aide **Prawn-For-Book** en ligne de commande

On peut obtenir une aide rapide sur un sujet donné, ou un mot, ou une fonctionnalité, en développant la commande `pfb aide` :

```
> pfb aide "le mot ou l'expression recherché"
```

Après avoir lancé cette commande, **Prawn-For-Book** affiche tous les endroits du manuel qui contiennent l'expression recherchée (par pertinence) et permet de développer le passage.

Rechercher régulière dans l'aide

On peut même faire une recherche *régulière* avec une *expression rationnelle* (si vous ne comprenez pas, cette fonctionnalités n'est peut-être pas pour vous...). L'expression rationnelle se trouvera entre guillemets et commencera et terminera avec une balance (« / »).

```
> pfb aide "/expression à rechercher>/"
```

Quelques exemples :

- Pour rechercher deux mots qui doivent se trouver dans la même phrase, et dans l'ordre donnée : `pfb aide "/mot1(.+ ?)mot2/"`.
- Pour chercher plusieurs mots : `pfb aide "/(mot1|mot2)/"`.
- Pour chercher un mot exact, mais qui peut être au pluriel : `pfb aide "/\bmots ?\b/"` (le `\b` désigne un délimiteur de mots).

Les « Snippets »

Les « snippets »⁴ (ou *complétions*) sont particulièrement utiles lorsque l'on désire gagner du temps dans la rédaction de son livre (même si, philosophiquement, "gagner du temps" est un concept *contre-artistique*...)

Ils consistent à taper quelques lettres, parfois une seule, de jouer la touche tabulation, et un texte vient remplacer automatiquement ces lettres.

Par exemple, si nous avons un personnage qui s'appelle *Rostatopoulos* et que nous ne voulons pas taper chaque fois ce prénom (avec toutes les erreurs de typo possibles...), nous créons un snippet avec la lettre « R » majuscule et chaque fois que nous tapons « R » suivi de la touche tabulation, **Prawn-For-Book** remplace ce « R » par « Rostatopoulos ».

⁴ Les snippets ne sont pas à confondre avec les variables textuelles qui s'écrivent textuellement dans le texte du fichier et seront ensuite remplacées par leur valeur fixe ou dynamique. Un snippet est remplacé immédiatement dans le texte par sa valeur définie.

Exclusivement dans Sublime Text

Mais cette fonctionnalité pratique ne fonctionne que dans un IDE adapté. En l'occurrence, à l'heure où nous écrivons ces lignes, elle ne fonctionne que dans l'application Sublime Text.

Installer les snippets

Pour pouvoir fonctionner, les snippets doivent être *installés* dans l'éditeur, il ne suffit pas qu'ils soient définis. Chaque fois que vous travaillez sur un nouveau livre, il faut installer les snippets définis.

On installe ces *snippets* à l'aide de la commande (dans une console ouverte au dossier du livre) :

```
> pfb install
```

Programmer un snippet

Programmer un snippet est simplissime, il suffit d'appeler la commande **pfb snippet**. Si l'on veut gagner du temps, il suffit de lui donner d'abord les lettres à taper (par exemple « R » dans notre exemple) suivi du texte de remplacement (« Rostatopoulos » dans notre exemple). Cela donnera :

```
> pfb snippet R Rostatopoulos
```

Si le texte de remplacement contient plusieurs mots, il est indispensable de les mettre entre guillemets.

```
> pfb snippet R "Rostatopoulos Alexis Triponov"
```


Messages d'erreurs et notices

On peut signaler des erreurs (messages rouges) et des messages de notices (messages bleus) au cours de la construction du livre grâce, respectivement, aux méthodes `erreur(« message> »)` (ou `error(« message> »)`) et `notice(« message> »)`. Ces méthodes doivent être placées seules sur une ligne, entre des doubles parenthèses.

((erreur(« message d'erreur> »)))

Attention : ces messages ne seront jamais gravés dans le livre, ils n'apparaîtront qu'en console lorsque l'on construira le livre.

Position

Un des grands avantages de ces messages est qu'ils indiquent clairement la source de l'erreur ou de la note. Ils indiquent le numéro de page dans le livre, ainsi que le numéro de ligne dans le fichier source ou le fichier inclus. De cette manière, on retrouve très rapidement l'endroit concerné par la note ou l'erreur.

Utilisation

On peut utiliser ces méthodes par exemple pour signaler une erreur dans le livre, qu'on ne peut pas corriger au moment où on la voit. Exemple :

((erreur(« Il manque ici le chapitre 13 »)))

Ou simplement pour signaler une chose qu'il faut garder en tête. Par exemple :

((notice(« Bien s'assurer que l'image qui suit soit en haut de la page. »)))

Exemple

À cet endroit précis nous avons placé un appel à une note avec le code :

((notice(« Une note depuis le manuel. »)))

Vous devriez la voir si vous lancez la « Reconstruction du manuel » (p. 74).

Exemple dans texte.pfb.md

Un paragraphe.

((add_notice("Une simple notice pour l'exemple.")))

Un autre paragraphe.

((add_erreur("Une erreur signalée.")))

Un troisième paragraphe.

Afficher grille de référence et marges

Pour régler finement l'aspect général du livre, et notamment les marges et les *lignes de référence*, on peut demander à **Prawn-For-Book** de révéler ces marges et ces *lignes de référence* par des traits de couleur.

On peut le faire soit en utilisant les *options* en ligne de commande :

```
> pfb build -margins -grid
```

... soit, si on veut les voir affichés pendant un long moment, en définissant les valeurs de `show_margins` et `show_grid` dans le fichier recette (du livre ou de la collection) :

```
# ./recipe.yaml ou ../recipe_collection.yaml
book_format:
  page:
    show_margins: true # mettre à false quand fini
    show_grid:     true # idem
```

Quel que soit le moyen utilisé, il faut penser à retirer ces lignes avant de graver le document final à envoyer à l'imprimerie. Dans le cas contraire, ces lignes apparaîtraient dans le livre !

Exportation du texte

Pour corriger orthographiquement le texte, par exemple avec Antidote, on peut bien sûr prendre le texte du fichier `texte.pfb.md` en faisant abstraction de tout ce qui relève de la mise en forme et des codes éventuels.

Mais lorsque ces codes « extra-texte » représentent une quantité non négligeable, il est préférable de procéder à un export du seul texte du livre.

Pour se faire, il suffit de jouer la construction du livre avec l'option `-t`.

```
> cd chemin/vers/mon/livre
> pfb build -t
```

Le texte seul du livre est alors mis dans un fichier `only_text.txt` (« seulement le texte » en anglais) à la racine du livre. C'est ce fichier qu'il faut corriger.

Pagination

La *pagination* concerne la numérotation des pages et — fonctionnalité propre à **Prawn-For-Book** — la possibilité aussi de numéroter les paragraphes.

Les types de pagination

La *pagination* concerne la numérotation des pages et — fonctionnalité propre à **Prawn-For-Book** — la possibilité aussi de numéroter les paragraphes.

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 ou sur les *marques en ligne* `{{TODO: Faire un lexique et mettre ce mot dedans}}` dans le texte.

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 ((### REF: couleur ###))).

(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 le fichier `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
```

Suspension de la pagination

Il est nécessaire parfois de suspendre la pagination sur quelques pages. Nous l'avons vu par exemple pour « La page de dédicace » (p. 35).

Pour ce faire, il suffit d'utiliser, seul sur une ligne, les marques `((stop_pagination))` pour interrompre la pagination et la marque `((restart_pagination))` pour la reprendre (« restart » signifie « redémarrer » en anglais).

Le livre final (document PDF) contiendra :

Ci-dessous nous allons utiliser la marque `((stop_pagination))` sur une ligne pour interrompre la pagination puis la marque `((new_page))` sur une autre ligne pour passer sur la page suivante⁵.

⁵ *Noter que c'est au moment de la création de la page que **Prawn-For-Book** doit savoir qu'il ne doit pas numéroté la page. Si la marque `((new_page))` était placée avant la marque `((stop_pagination))`, alors la nouvelle page créée par la première marque serait numérotée. De la même manière, la marque `((restart_pagination))` doit être placée avant la marque `((new_page))` pour que la nouvelle page créée soit numérotée.*

Cette page ne doit pas être numérotée.

Numérotation des pages vierges

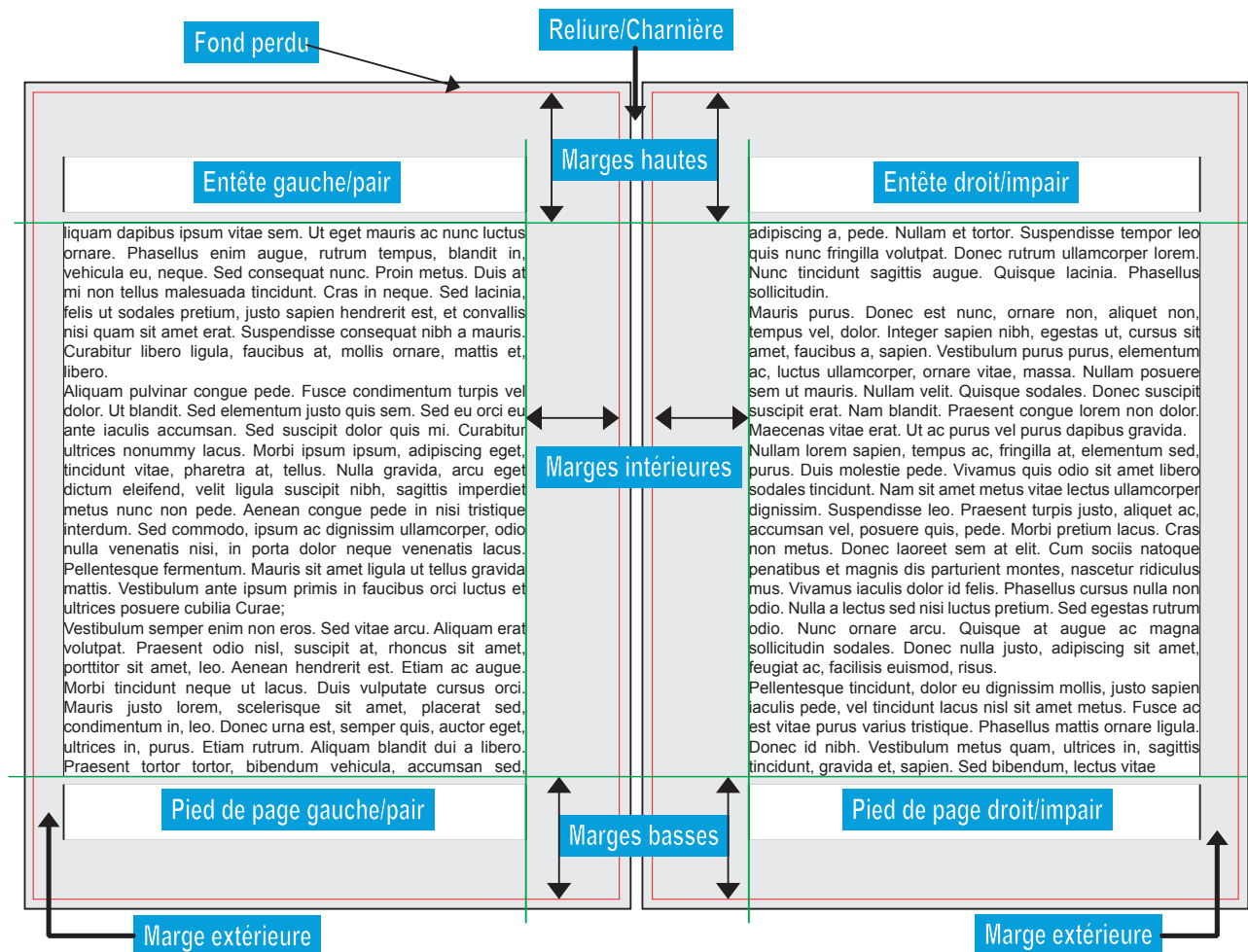
Les pages vierges peuvent être numérotée en modifiant la donnée `book_format: page: :no_num_if_empty:` (« no num if empty » signifie « pas de numéro si la page est vide »).

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

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

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 paire ou impaire, 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'imprimerie 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 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 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.

Page en multi-colonnes

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

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

Pour arrêter d'être en double colonnes, il suffit d'utiliser :

$$((\text{colonnes}(1))$$

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

Définition plus précise des colonnes

On peut définir la gouttière (espace entre chaque colonne) grâce à la propriété `gutter` à mettre en deuxième paramètre :

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

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

```
(( colonnes(2, width: PAGE_WIDTH/2) )) 6
```

⁶ Vous remarquez ci-dessus l'utilisation d'une constante (cf. « Constantes » (p. 80)).

Précaution pour les colonnes

Attention à toujours terminer par ((colonnes(1))), surtout si c'est la dernière page, dans le cas contraire les pages multi-colonnes ne seraient pas gravées.

Le livre final (document PDF) contiendra :

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

[illegible]

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. 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. Fin du texte.
--	---	---

On revient ensuite à un texte sur une colonne avec la marque ((colonnes(1))). Et c'est la fin de l'usage des colonnes multiples, on revient sur une page normale.

La double colonne suivante est obtenue quant à elle grâce au code : `((colonnes(2, width:PAGE_WIDTH/1.5, gutter:50)))` qui est placé juste sous cette ligne.

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. . Fin du texte.

On revient à nouveau à un texte sur une colonne avec la marque ((colonnes(1))).

Par défaut, **Prawn-For-Book** laisse une ligne vide au-dessus et une ligne vide au-dessous d'un texte en multi-colonnes. On peut contrecarrer ce comportement à l'aide des propriétés `lines_before` et `lines_after`. À `false` ou `0`, aucune ligne ne sera ajoutée.

On peut même mettre `line_after` à -1 lorsqu'il arrive, parfois, qu'une ligne supplémentaire soit ajoutée par erreur (les calculs de Prawn sont parfois impénétrables).

Avant les doubles colonnes suivantes, nous avons écrit le code : `((columns(2, lines_before:0, space_after: -LINE_HEIGHT)))` (nous avons volontairement utilisé la traduction anglaise « columns »).

Début du texte en double colonnes. Un
texte en double colonnes qui ne devrait
pas présenter de ligne vide de
séparation ni au-dessus avec le texte

avant ni au-dessous avec le texte après.
Tous ces textes devraient être collés. Fin
du texte en double colonnes.

Paragraphe sous le texte en double colonnes collées. Ci-dessus, nous avons dû jouer sur `space_after`, avec une valeur négative, pour arriver à nos fins car `lines_after` restait inefficace. Au-dessus, on peut aussi jouer avec `space_before` si on veut définir l'espace avant. Notez que le texte est quand même remis sur des lignes de référence à chaque fois.

Ligne en trop dans les multi-colonnes

Parfois il peut arriver que **Prawn-For-Book** compte une ligne de trop dans les colonnes, ce qui produit cet alignement pas très heureux :

Premier	Deuxième en regard
Deuxième	Troisième en regard
Troisième	
Premier en regard	

Pour palier cet écueil, on met la propriété `no_extra_line_height` à `true` dans la définition des colonnes. On obtient alors :

Premier	Premier en regard
Deuxième	Deuxième en regard
Troisième	Troisième en regard

Le Mode Expert

Description

Le *mode Expert* permet d'élargir de façon exponentielle les possibilités de **Prawn-For-Book** afin de produire les contenus les plus variés et les plus originaux.

Il demande une compétence particulière dans le langage Ruby ainsi qu'une bonne connaissance de la gem Prawn qui permet de produire les livres avec **Prawn-For-Book**.

Indication du contexte d'erreur

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

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

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

Injection

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

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

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

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

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

Évaluation du code ruby

Tous les codes qui se trouveront entre « `#{...}` » (ou entre « `#{{{...}}}` » lorsque le code contient des accolades) 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 ou le numéro de version du livre pour l'insérer dans les premières pages à titre de repère.

Évaluation au second tour

Certaines opérations ou certains calculs ne peuvent s'opérer qu'au second tour de l'application⁷ — typiquement, le nombre de pages du livre —. On utilise alors la tournure suivante pour réaliser ces opérations.

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

⁸

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.

⁷ 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.

⁸ Noter ici l'utilisation des trois accolades obligatoires lorsque le code lui-même a recours aux accolades.

Si le fichier « `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 25 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`

Mode Multi-colonnes

En tant qu'expert, vous pouvez utiliser le mode multi-colonnes (affichage du texte sur plusieurs colonnes) à l'intérieur des formateurs.

Bien entendu, vous pouvez, si vous êtes parfaitement à l'aise avec ça, utiliser les `column_box` de **Prawn**. Mais **Prawn-For-Book** propose là aussi des outils plus élaborés qui permettront une mise en page assistée.

Imaginons l'index `entree` qui permet de gérer les entrées du dictionnaire que vous construisez. À la fin du livre, vous voulez afficher cette index sur trois colonnes. Vous allez donc implémenter la méthode `CustomIndexModule#print_index_entree` dans le fichier `formater.rb`.

Elle ressemblera au code de la page suivante.

```
module CustomIndexModule
```

```
  # Méthode appelée automatiquement si le code
  #   `(( index\\(entree) ))`
  # est utilisé dans le texte
  #
  def print_index_entree(pdf)

    # Les paramètres d'instanciation du multi colonnes
    params = {
      column_count: <nombre de colonnes>,
      width:        <largeur si autre que page complète>,
      gutter:       <gouttière si autre que valeur défaut>,
      lines_before: <nombre de lignes avant si autre que 1>,
      lines_after:  <nombre de lignes après si autre que 1>,
      space_before: <espace avant si nécessaire>,
      space_after:  <espace après si nécessaire>
    }
    # On instancie un texte multi-colonnes
    mc_block = Prawn4book::PdfBook::ColumnsBox.new(book, **params)
    # items contient la liste de toutes les entrées
    items.each_with_index do |item_id, occurrences, idx|
      # On traite les items pour obtenir le texte
      str = ...
      # On en fait une instance de paragraphe
      para = Prawn4book::PdfBook::NTextParagraph.new(
        book: book,
        raw_text: str,
        pindex: idx,
      )
      # On peut indiquer la source, pour les messages d'erreur
      # éventuels
      para.source = "Construction de l'index des entrées"
      # On peut supprimer l'indentation éventuelle
      para.indentation = 0
      # Et on injecte le paragraphe dans le bloc multi-colonnes
      mc_block << para
    end
    # On diminue la fonte à utiliser et on l'utilise
    fonte = Prawn4book::Fonte.dup_default
    fonte.size = 10
    pdf.font(fonte)
    # Il suffit maintenant d'imprimer ce bloc multi-colonnes
    mc_block.print(pdf)
    # Done!
  end
end #/module
```

Formateurs

En mode expert, on a un accès complet à toutes les possibilités qu'offrent **Prawn-For-Book**, c'est-à-dire, n'ayant pas peur de le dire, *un monde infini, sans limite*.

Pour se faciliter la vie, certaines méthodes propres permettent une implémentation plus rapide. C'est le cas de la méthode `Printer#pretty_render` qui, comme son nom l'indique, permet d'obtenir un bon rendu dans le livre sans effort (et, notamment, un traitement du texte sur les lignes de référence — cf. page 29)

Voici un code exemple :

```
# Par exemple dans ./formater.rb
def mon_helper(pdf)
  # On définit une fonte particulière
  mafonte = Prawn4book::Fonte.new(name:'Arial', size:8, style:
:normal)
  # On écrit le texte voulu dans le document.
  Printer.pretty_render(
    pdf:      pdf,
    fonte:    mafonte,
    text:     "Mon texte qui sera bien disposé [etc.]",
    options:  {left: 40, right: 80},
    owner:    nil,
  )
end
```

Le livre final (document PDF) contiendra :

Ci-dessous, je vais appeler la méthode `mon_helper` définie ci-dessus avec le code :

```
#{-mon_helper(pdf)}9.
```

Mon texte qui sera bien disposé sur les lignes de référence malgré sa taille plus petite que le texte normal de ce mode d'emploi. C'est une police Arial, de taille 8, qui va se placer à 40 points de la marge gauche et à 80 points de la marge droite, car le `:left` des options a été mis à 40 et le `:right` a été mis à 80. Ça permet d'avoir un texte qui se place où on veut dans la page, plus serré que les marges.

⁹ Le « - » devant l'appel de la méthode permet de n'imprimer aucun retour de méthode. Rappel : sinon, c'est toujours le retour de l'appel qui est imprimé.

Modification de la hauteur de ligne

En tant qu'expert, vous pouvez modifier localement la hauteur de ligne (`line_height`) de deux façons.

Noter cependant que cette modification n'est pas à prendre à la légère puisqu'elle va bousculer l'harmonie des lignes du livre. Elle doit se faire en priorité sur une *belle page* (une page de gauche, paire) afin que les lignes des pages en vis à vis restent alignées et se réserver à l'annexe du livre, par exemple, ou à la page des index, quand des tailles de police plus petites sont utilisées.

Modification de la hauteur de ligne dans le texte

Au fil du texte, la hauteur de ligne peut être redéfinie avec :

```
(( line_height(<nouvelle hauteur>, {fname:<police>,
  fname:<taille>, fstyle:<style>})) ))
```

Par exemple :

```
(( line_height(10, {fname:'Garamond', fsize:9, fstyle:'normal'})
))
```

Modification de la hauteur de ligne par formateur

On peut appeler une méthode définie dans `prawn4book.rb` par exemple :

```
# ./prawn4book.rb
module Prawn4book
  def redefinir_hauteur_de_ligne(pdf, valeur)
    pdf.line_height = valeur
    Fonte.default = ...
  end
end
```

Et dans le texte par exemple (mais cette méthode pourrait être appelée de n'importe où) :

Un texte quelconque.#{redefinir_hauteur_de_ligne(10)}

Annexe

Reconstruction du manuel

Pour voir certaines fonctionnalités, il est nécessaire de relancer la construction de ce manuel autoproduit. Pour ce faire, suivez les étapes suivantes :

- dans une fenêtre Terminal, jouez la commande `pfb open`,
- dans la liste, choisir « Ouvrir le dossier du manuel »,
- ouvrir un Terminal dans ce dossier (control-clic sur le dossier puis choisir « Nouveau Terminal au dossier » ou similaire),
- jouer la commande `pfb build` dans ce Terminal (parfois il pourrait être demandé d'ajouter l'option de débuggage, dans ce cas il faudra jouer `pfb build -debug`).

Le manuel autoproduit se reconstruit alors en quelques secondes.

Marques markdown

Vous trouverez ci-dessous toutes les marques Markdown utilisables.

Si le fichier « *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 ### REF: tables ###.

Le livre final (document PDF) contiendra :

italique	* ... *	texte en italique
gras	** ... **	texte en gras
souligné	~ ... ~	texte souligné

Le format YAML de la recette

exposants 1^{er} 1^{er}

Le format **YAML** est un format très simple de présentation et de consignation des données. Il est utilisé dans **Prawn-For-Book** pour définir recette/juste_titre (p.), que ce soit pour un titre unique (cf. page « Recette du livre » (p. 22)) ou pour une collection (cf. page « Recette de la collection » (p. 22)). 2^e

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. X^{1e}

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"

```

Les "Fontes-String"

Pour définir les polices dans les éléments, à commencer par la recette, on utilise de préférence ce qu'on appelle dans **Prawn-For-Book** les « **fonte-string** » (« string » signifie quelque chose comme « caractère » en anglais).

Ces **fonte-strings** se présentent toujours de la même manière, par une chaîne de caractères (de lettres) contenant 4 valeurs séparées par des balances, dans l'ordre :

- le **nom** de la police,
- le **style** de la police (qui doit être définie,
- la **taille** à appliquer au texte (en points-pdf),
- la **couleur** éventuelle (pour la définition de la couleur, voir « Définition de la couleur » (p. 75)).

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

```

---
font: "<police>/<style>/<taille>/<couleur>"
# Par exemple
font: "Numito/bold_italic/23/FF0000"

```

Définition de la couleur

En règle générale, la couleur dans **Prawn-For-Book** peut se définir de deux façons :

- en hexadécimal (comme en HTML) à l'aide de 6 chiffres/lettres hexadécimal (donc de 0 à F),
- en quadrichromie, avec CMJN (Cyan, Magenta, Jaune, Noir).

COULEUR HEXADÉCIMALE

En hexadécimal, on a 3 paires de deux chiffres/lettres qui représentent respectivement les quantités de rouge, vert et bleu. Par exemple « AAD405 » signifiera « AA » de rouge, « D4 » de vert et « 05 » de bleu.

"000000" représente le noir complet, « FFFFFFFF » représente le blanc complet. Lorsque toutes les valeurs sont identiques (par exemple « CCCCCC ») on obtient un gris (mais il existe d'autres moyens d'obtenir du gris).

Quelques couleurs hexadécimales aléatoires :

- Couleur 995407
- Couleur 7e6004
- Couleur 4a54b7
- Couleur b86dfa
- Couleur 844a0e
- Couleur 9aaaa7
- Couleur 3f78d8
- Couleur a8350f
- Couleur 256593
- Couleur 237408

COULEUR QUADRICHROMIQUE

La quadrichromie, représentée souvent par « CMJN » (ou « CMYK » en anglais), est le format de la couleur en imprimerie. On aura une liste (crochets) contenant les 4 valeurs de 0 à 127 pour le Cyan (C), le Magenta (M), le Jaune (Y) et le noir (K). Par exemple « [0, 12, 45, 124] » signifiera qu'il n'y aura pas de Cyan, qu'il y aura 12 de magenta, 45 de jaune et 124 de noir. [0, 0, 0, 0] représente le blanc complet, [127, 127, 127, 127] le noir complet.

Quelques couleurs quadrichromiques :

- Couleur [0,0,0,127]
- Couleur [127,0,0,127]
- Couleur [127,0,0,0]
- Couleur [0,127,0,127]
- Couleur [0,127,0,0]
- Couleur [0,0,127,127]
- Couleur [0,0,127,0]
- Couleur [127,127,127,127]
- Couleur [127,127,127,0]
- Couleur [99, 104, 72, 52]
- Couleur [120, 118, 50, 33]
- Couleur [69, 123, 104, 97]
- Couleur [22, 66, 87, 43]
- Couleur [15, 99, 56, 116]
- Couleur [33, 80, 62, 58]
- Couleur [52, 86, 60, 39]
- Couleur [88, 92, 119, 76]
- Couleur [122, 103, 118, 59]
- Couleur [34, 45, 91, 31]

Outils couleurs *online*

Vous trouverez sur le net, en tapant simplement « Couleur HTML CMJN » dans un moteur de recherche, des outils *en ligne* vous permettant de choisir des couleurs précises et d'obtenir leur code.

Rogner une image SVG

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

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

Le livre final (document PDF) contiendra :

Image non rognée :



(note : la page vide précédente est occupée par le blanc de l'image rognée)

La même image, rognée cette fois :



Amet exercitation ut dolore in in nulla adipisicing laborum.

Synopsis de création

Cette section présente une sorte de synopsis que vous pouvez suivre pour créer votre livre.

- ...
- ajustez parfaitement vos images en jouant sur leur `adjust` :. N'hésitez pas à ajouter des lignes avec (`line`) pour bien séparer les choses si nécessaire
- ...

Installation de l'application

Pour que l'installation de l'application fonctionne, il faut avoir une version ruby « up and running » sur son ordinateur et avoir chargé le dossier principal dans son dossier `/Applications`.

Il faut créer ensuite un alias `pfb` qui lancera l'application, grâce à la commande :

Sur MacOS

```
> ln -s /Application/Prawn4book/prawn4book.rb /usr/local/bin/pfb
```

Sur Windows

{{TODO}}

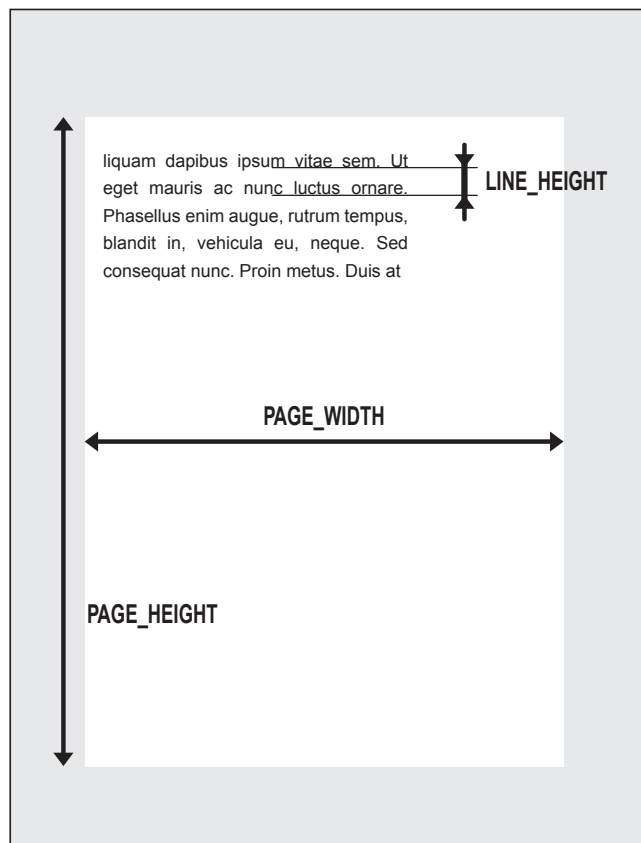
Constantes

Trouvez ci-dessous la liste des constantes utilisables dans les paramètres des méthodes.

- **LINE_HEIGHT** | (« Hauteur de ligne » en anglais) Hauteur courante de la ligne.
- **PAGE_HEIGHT** | (« Hauteur de page » en anglais) Hauteur de la page du livre hors marge (c'est-à-dire qu'on ne compte pas les marges dans cette longueur, c'est la hauteur à l'intérieur des marges).

- **PAGE_WIDTH** | (« Largeur de page » en anglais) Largeur de la page du livre hors marge.

Vous pouvez retrouver toutes ces informations dans l'image ci-dessous.



Les pages d'un livre

C'est un extra de **Prawn-For-Book**, parce qu'on ne trouve jamais l'information complète nulle part (à part sur le blog des éditions Icare dont s'occupe votre serviteur — <https://www.icare-editions.fr>), c'est la liste normalement complète des pages autres que les pages de texte qu'on trouve dans un livre professionnellement imprimé.

Liste exhaustive des fonctionnalités

- définition de la taille du livre
- pagination automatique et personnalisable
- définition de la fonte par défaut
- définition des pages spéciales à insérer « Les pages spéciales » (p. 31)
- justification par défaut des textes
- colorisation des textes

- suppression des veuves et des orphelines
- suppression automatique des lignes de voleur
- nombreuses sortes de puces et puces personnalisées puces/puces (p.)
- 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
- gestion par défaut ou personnalisée des images et de leur légende
- génération dynamique de contenu
- 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 (p.)
- placement sur n'importe quelle ligne de la page
- exportation seulement du texte produit
- exportation comme livre numérique (pur PDF)

AVANT-PROPOS	11
FONCTIONNALITÉS	15
Généralités	17
Aperçu rapide des commandes	17
Les Forces de <i>Prawn-For-Book</i>	17
Définitions minimales	19
Les Recettes	22
Recette du livre	22
Recette de la collection	22
Définition des fontes (embarquées)	23
Définition de la fonte par défaut	23
Éditeur / Maison d'édition	24
Le Texte du livre	25
Texte au format pseudo markdown	25
Définition des titres	25
Comportement (par défaut)	28
Alignement sur la grille de référence	29
Les pages du livres	31
Introduction	31
Les pages spéciales	31
La page de Faux-titre	31
Page de titre	33
La page de mention légale	35
La page de dédicace	35
Table des matières	38
La page des remerciements	41
Page des crédits (Colophon)	42
Page d'index	45
Aides & Assistants	47
Manuel et autres aides	47
Les « Snippets »	48
Messages d'erreurs et notices	49
Afficher grille de référence et marges	49
Exportation du texte	50

Pagination	51
Les types de pagination	51
Suspension de la pagination	52
Format du livre et des pages	55
Définition des marges	56
Page en multi-colonnes	59
Le Mode Expert	62
Description	63
Indication du contexte d'erreur	63
Injection	63
Évaluation du code ruby	64
Bibliographies en mode expert	66
Mode Multi-colonnes	66
Formateurs	68
Modification de la hauteur de ligne	68
ANNEXE	71
Reconstruction du manuel	72
Marques markdown	72
Le format YAML de la recette	72
Les "Fontes-String"	73
Définition de la couleur	73
Rogner une image SVG	75
Synopsis de création	78
Installation de l'application	78
Constantes	78
Les pages d'un livre	79
Liste exhaustive des fonctionnalités	79

Icare Éditions

Conception
Philippe Perret
(philippe.perret@icare-editions.fr)

Mise en page
Prawn-for-book

Couverture
MM & PP

Correction & relecture
Marion Michel

Imprimé par Prawn-For-Book