





# Livre complexe

(présentant de nombreux cas d'impressions.)

Philippe Perret

Icare Éditions





Ce livre doit permettre de tester énormément de choses dans le détail. Grâce à l’affichage (par la recette) de la grille de référence, on peut voir dans le livre produit le résultat. Il va y avoir beaucoup de changements de fontes et de tailles (elles sont toutes définies dans le fichier recette).

*Ce texte est écrit dans la police “Helvetica” en taille 21 et style italique (c’est une ligne de pfbcode qui le détermine juste au-dessus). Elle doit juste s’adapter aux lignes de références qui sont placées dans cette partie (par défaut) avec une hauteur de ligne de 24 points.*

On poursuit avec un texte en police “Times-Roman” en taille 12 et style romain. De la même manière, ce texte est assez long pour pouvoir se placer sur plusieurs lignes, afin de voir si le traitement par ligne fonctionne correctement et que le texte se place bien et naturellement sur les lignes de référence tracées.

Un paragraphe dans la police normale, mais avec des *\*textes en italiques\**, des **\*\*textes en gras\*\*** et des textes soulignés. Il possède aussi un 1<sup>er</sup> exposant et une 1<sup>re</sup> ré-utilisation d’exposant ainsi qu’une note<sup>^1</sup> qui doit être placée en dessous du paragraphe, dans un style un peu différent.

<sup>^1</sup> C’est le commentaire de la note qui a été placée plus haut, qu’on doit correctement formater en fonction des choix dans le livre de recette {{ Cette définition doit être ajoutée }}

\* Traitement du formatage pseudo-markdown

\* Traitement des veuves

\* Traitement des orphelines

\* Traitement des lignes de voleur

\* La modification du `THIEF_LINE_WIDTH` à la volée, pour pouvoir modifier localement la longueur d'une ligne de voleur. On doit pouvoir aussi le faire dans la recette (autre test ?)

\* Penser à ajouter la définition de l'aspect des notes dans la recette (manuel). Ajouter les valeurs par défaut dans `RECIPE_DEFAULT`.

\* Ajouter la page d'infos, avec toutes les infos (l'idée est qu'il y ait plus d'infos que de lignes, pour voir comment on va s'y prendre => doubler les lignes et mettre les caractères plus petits pour s'adapter.

\* Pour éviter d'avoir une ligne aux caractères trop serrés, il faut appliquer le ``character_spacing`` à tout le paragraphe (et donc le calculer par rapport à tout le paragraphe, pas seulement par rapport à la ligne) — la différence est notable quand on a des gros caractères

\* Le faux-titre doit être sur une belle page

- \* Les lignes de la page de faux-titre doivent se poser sur les lignes de référence

- \* Par défaut, les pages doivent être numérotée

- \* Quand changement de fonte, la première ligne est mal placée (je pense qu'elle se place comme pour la fonte précédente.

Comment gérer les lignes complexes (les lignes qui ne sont pas simples...), à commencer par les items de liste. Mais on peut imaginer que les tables en sont aussi. Il faut vraiment parvenir à “sortir” le traitement par ligne. Par exemple en ayant un constructeur (Printer ?) à qui on envoie :

- \* un texte,

- \* peut-être une fonte générale

- \* des paramètres définissant la largeur :width et la position :left,

- \* le pdf (Prawn::document)

- \* un hauteur (cursor)



et qui calcule et imprime le texte. L'essai sera concluant si on parvient à imprimer une table (mais une fausse table) avec quatre colonnes de tailles différentes par exemple dont les cellules contiennent des textes assez longs, avec des fontes différentes, et des lignes (des lignes ajoutées par dessin, donc, puisque qu'on n'utiliserait pas les vraies).

Il faudrait cependant que les vraies tables puissent être utilisées, qui zapperaient l'alignement sur la grille de référence [Peut-être que l'utilisation actuelle fonctionnerait].