





# Livre complexe

(présentant de nombreux cas d'impressions.)

Philippe Perret

Icare Éditions



[illegible]





# Complexité testée

# Présentation

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 (cette phrase possède une référence “grille\_reference”). Il va y avoir beaucoup de changements de fontes et de tailles (elles sont toutes définies dans le fichier recette). La page de reference\_ulterieure est la page n°(( 34 ))

Noter quand même qu’on ne doit pas créer un gros fichier comme ça avec trop de choses, sinon, il serait difficile de voir où le bât blesse.



## Premiers tests

*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 commentaires, de la texte sur "première" des  
textes souligné plus placé, de qui on doit exactement  
formater en fonction des choix dans la liste de cette note  
sur "première" mais aide à la plate "deuxième" qui  
doit être bien dans un style de première de l'écriture  
autre note sur deuxième^^.

Un paragraphe juste en dessous de la note pour  
voir s'il serait bien placé sous le trait et non pas  
dessus ce qui serait inélégant.





Paragrapbes avec  
formatage  
personnalis 

- ☐ Un premier paragraphe de style “simplenote” qui se caractérise par une puce “pictophil” (lettre “n”) placée au bon endroit en décalant le texte qui sera toujours bien placé.
- ☐ Une autre simple note pour voir si les caractères qui ne s’affichent pas vont s’afficher cette fois avec une ligne inférieure (en fait le problème vient de .otf).



Une note pour de la documentation.





Tests restant à  
faire



– 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.
- Utiliser les § (caplock §) pour désigner le numéro du paragraphe dans la numérotation hybride.







Bogues à corriger  
/ implémenter

- Le tiret des items de list doivent avoir la fonte du texte (voir le premier qui prend la fonte du titre)
- Plus grave, les tirets d'item de liste ne doivent être mis que lorsque l'on sait où va se retrouver le paragraphe. S'il doit passer à la ligne, le tiret doit lui aussi passer à la ligne
- La largeur de l'item de liste doit être amputée du décalage left (pour le moment, il est normalement réglé (avec margin-left, mais le programme n'en tient aucun compte)





# Choses à implémenter









# Réflexions

## Second tour

Pour le second tour, on n'a absolument pas besoin de repartir du texte. On va plutôt repartir de tous les paragraphes, et modifier les références puisque normalement c'est la seule chose qui nécessite ce second tour (on a un troisième tour avec les définitions du scénodico, mais c'est un tout autre problème qui sera résolu).

Ici se trouve la référence à `reference_ulterieure(( ))`. La première allusion à “grille de référence” se trouve page n°(( 8 )).

C'est en fait les corrections faites par `AnyParagraph.__parse` qui sont conservées.