

Le textallion, un simple traitement de texte

Eric Forgeot

2021-04-23

Table des matières

Manuel	v
Présentation	vii
Obtenir le Textallion	ix
Paquets optionnels	ix
Pré-requis (installation)	x
Linux	x
Windows	x
.	x
.	xi
*BSD	xii
Mac OS X	xii
Utilisation	xiii
HTML	xvi
PDF	xvi
ePUB	xvi
Syntaxe	xviii
Rappel et mémo sur txt2tags	xviii
Symboles supplémentaires pour la typographie	xviii
.	xix
.	xx
.	xx

.....	xxi
.....	xxii
.....	xxii
.....	xxii
Liens	xxiii
Rajout de sons, de musiques ou de vidéos	xxiii
Taille du texte	xxiv
Zones en italique ou en gras, zone de code	xxiv
Exergue (épigraphe)	xxiv
Couleurs	xxv
Centrage de texte	xxvi
Colonnes	xxvi
Tables	xxvii
Code \LaTeX complet	xxviii
.....	xxix
Exemple code unix	xxx
Unicode, Ligatures... (rendu XeTeX)	xxxii
Installation de nouvelles polices pour LaTeX	xxxii
Module de courrier	xxxiii
Module de création de livres-jeux	xxxiv
Exemples de prose	xxxv
La tour de Nesle	xxxvi
Ressources diverses	xxxvii
Outils concurrents, similaires ou associés	xxxix

License and credits	xi
Changelog	xliii
Todo and known bugs	i
Syntax, backend and docs	li
HTML	lii
PDF (LaTeX and XeTeX)	liii
EPUB	liv
Windows	lv
txt2cyoa	lvi

Manuel



TEXTALLION (tekstaljɔ̃) n.m. — (2009 ; de *texte*, et suffixe inconnu *-llion*, probabl. d'o. ani-male ou indiquant l'idée de la prem. pers. du pluriel. On l'écrit indiff. *textallion* ou *TeXtallion*.)

Système de traitement de texte utilisant une syntaxe wiki, générant une présentation finale attrayante, que cela soit au format PDF, HTML ou ePUB, et permettant une mise en forme simplifiée au clavier au lieu d'avoir à cliquer dans des menus. Accessoirement, la source d'un document peut être lue sans logiciel particulier. V.[txt2tags](#).

Citation :

« *Je t'envoie un fichier .t2t en pièce jointe, ne me dis plus que tu n'as pas le bon logiciel pour l'ouvrir!* »

GUILLAUME GRUMBOLD, BIBLIOTHÉQUAIRE

Présentation

Le projet **textallion** permet à ses utilisateurs d'éditer et de publier simplement leurs textes, tout en leur donnant un aspect agréable.

Il est né de l'admiration pour l'apparence et la qualité du placement des lettres dans le système `LaTeX`, ainsi que pour la facilité d'utilisation de `txt2tags`.

Mais il est né également de la frustration de ne pas pouvoir utiliser simplement `LaTeX` pour écrire de la prose, sans avoir à subir sa syntaxe particulièrement intrusive et sa complexité d'apprentissage ¹.

Le but est donc de pouvoir sortir un roman ou un texte littéraire, en utilisant `txt2tags` comme pré-processeur, et ainsi garder des marques de formatage non distrayantes, tout en pouvant exporter en `LaTeX` et `html` selon le type du document final envisagé : publication papier (`pdf`), page internet (`html`, ainsi qu'en [version de présentation avec slidy](#)), ou encore livre électronique (`epub`), et cela à partir d'une [source unique pour tous ces documents](#).

Ainsi, cet outil s'adresse surtout à ceux qui souhaitent écrire facilement et rapidement des textes de type poésie ou prose, sans entrer dans les méandres des commandes `LaTeX`. (`LaTeX` est parfois le contraire du `wysiwyg` (what you see is what you get), par certains aspect cela devient du `wysiwyhtst` : "What you see is what you have tested several times".)

Il présente par défaut : mise en gras, italique, soulignage, barré, texte coloré, organisation hiérarchique (titres et sous-titres), listes à puce ou numérotées, inclusion d'images et de liens hypertextes, numérotation des pages, haut de page avec titre et ligne de séparation, support des notes de bas de page, exergues (épigraphes), table des matières, colonnes, modification de taille de polices (très basique), ornements type lettrine ou petites feuilles aldines.

Il inclut également un système pour la création de « livres-jeux » (livres dont vous êtes le héros).

Cet outil est relativement simple à utiliser, mais pas simpliste : il est nécessaire pour en tirer partie d'avoir quelques bases en informatique, du fait de l'obligation de l'installation et de l'utilisation des ressources de plusieurs autres projets en même

1. Remarque qui peut également concerner dans une moindre mesure la syntaxe `html`

temps (python, latex). Dit autrement, c'est un peu un outil pour geeks, mais malgré tout il sera possible d'avoir un confort d'écriture nullement atteint avec les traitements de texte habituels, et l'installation de ces autres logiciels ne nous semble pas insurmontable avec un peu de bonne volonté.

On peut imaginer 3 manières d'aborder textallion :

- En se contentant du style fourni de base, et sans même se préoccuper d'une quelconque syntaxe wiki spécifique, en écrivant comme dans un bloc-note, on peut déjà produire un document PDF au format A4, réductible en A5, très présentable². Rappelez-vous que la plupart des romans n'ont pas de mise en page très audacieuse et ne font pas appel non plus à tout l'arsenal que l'on peut trouver par exemple dans un texte scientifique ou une documentation (colonnes, index, changement de polices...)
- En se laissant guider dans l'édition du fichier de style utilisateur (sample.sty), on modifiera les options permettant de personnaliser son document PDF³. Par exemple à laquelle s'ajoute celle de textallion.

Ce n'est donc pas un logiciel clé en main qui permettra de réaliser tout et n'importe quoi, mais c'est la réorganisation d'outils existants qui, mis ensemble avec quelques règles de styles, permettent de réaliser une tâche précise.

Ce présent manuel se focalise surtout sur la création de documents pdf, destinés en particulier à l'impression, mais tout est normalement adaptable pour la publication en html pour internet, et le format ePUB pour les liseuses numériques (lecteurs d'ebooks, livres électroniques, téléphones portables, smartphones).

Note : Ce projet avait pour nom « txt2tex » à l'origine.

2. Pour comparer avec la version de base de txt2tags, dans l'export \LaTeX les accentuations françaises ne sont pas supportées, et le rendu est très basique.

3. Malheureusement, toutes ces options sont indiquées en anglais pour le moment

Obtenir le Textallion

La dernière version en développement se trouve à cette adresse :

Elle s'obtient :

```
hg clone https://farvardin@hg.code.sf.net/p/textallion/code textallion-code
```

Une copie du dépôt est sur github à cette adresse :
<https://github.com/farvardin/textallion> On peut l'obtenir :

```
git clone https://github.com/farvardin/textallion.git
```

Il y a également une copie ici (pas forcément toujours à jour) :
<http://anamnese.online.fr/site2/textallion/textallion.zip>

Il est possible de retrouver les anciennes versions dans
<http://anamnese.online.fr/site2/textallion/archives/>

Paquets optionnels

Il est recommandé d'automatiser vos créations de documents en utilisant 'GNU make', à partir du makefile d'exemple. Vous pouvez également ajouter des modifications postscripts (édition pdf au niveau vectoriel) avec des outils tels que pstopdf, psutils (pdfnup, psbook) et pdfjam.

Parfois l'utilisation de \LaTeX n'est pas toujours aisée. Pour un export simple vers le format PDF, il est possible d'utiliser `wkhtmltopdf`

Pré-requis (installation)

Le Textallion a été testé avec succès sur ces plateformes :

Linux

LaTeX et python 3 doivent être présent sur votre ordinateur. La plupart des distributions, sinon toutes, fournissent cela. Par exemple Archlinux, Mandriva et Debian, via TeXlive. (Vous devez avoir au moins les paquets `texlive-core` et `texlive-humanities`).

(sous Debian/Ubuntu/Mint : `apt-get install python3 texlive-base texlive-latex-base texlive-extra-utils texlive-latex-recommended texlive-latex-extra texlive-fonts-extra texlive-lang-french latexml tipa texlive-fonts-recommended texlive-xetex tidy calibre wkhtmltopdf`)

Vous pouvez utiliser le petit script d'installation `textallion_install.sh` (à lancer en root), pour copier l'ensemble des fichiers nécessaires dans `/usr/share/textallion/`, ou le dossier de votre choix.

```
sudo bash textallion_install.sh
```

Pour désinstaller ce qui a été précédemment installé par ce script, il suffit d'effacer le dossier `/usr/share/textallion/` et le fichier `/usr/bin/textallion`

Windows

1^{ère} méthode

- Installer `cygwin`, un logiciel permettant d'accéder à des outils Unix. Installer notamment en plus des paquets de base python, make, tidy, psutils
Ouvrir le fichier `c:\et` modifiez la ligne commençant par `PATH=` (vers la ligne 23), comme suit :

```

PATH="/usr/local/bin:/usr/bin:/cygdrive/c/Program
Files (x86)/MiKTeX 2.9/miktex/bin:/cygdrive/c/Program
Files/MiKTeX 2.9/miktex/bin:/cygdrive/c/Program
Files (x86)/MiKTeX 2.8/miktex/bin:/cygdrive/c/Program
Files/MiKTeX 2.8/miktex/bin:/cygdrive/c/Program Files
(x86)/:/cygdrive/c/Program Files:/cygdrive/c/Program
Files (x86)/Mozilla Firefox:/cygdrive/c/Program
Files/Mozilla Firefox:/cygdrive/c/Program Files
(x86)/Geany/bin:/cygdrive/c/Program Files/Geany/bin:/cygdrive/c/Program
Files (x86)/Calibre2:/cygdrive/c/Program Files/Calibre2:/cygdrive/c/Program
Files (x86)/SumatraPDF:/cygdrive/c/Program Files/SumatraPDF:${PATH}"

```

- Utilisation de \LaTeX avec [MiKTeX](#)⁴ pour la création des fichiers pdf : Choisissez la version "basic" de MiKTeX, et si lors de la première utilisation il manque des paquets lors de la création de vos documents, le logiciel ira les chercher directement sur internet ("install missing packages on the fly"). Il peut y avoir une douzaine de ces paquets manquant lors de la première utilisation de textallion. Si vous utilisez MiKTeX 2.9, il faudra installer manuellement le paquet `symbol`, avec le Package Manager.
Pour créer des ePUBs, vous aurez besoin d'installer [Calibre](#)

Autre méthode

[Python](#) doit être installé sur votre ordinateur. Choisissez la version 3 de python de préférence, la version 2 devrait être encore compatible, mais python3 est maintenant utilisé par défaut.

- décompressez l'archive contenant textallion dans par exemple C:(())(pour le reste des exemples, on a considéré que cela se trouvait dans ce dossier)
- Utilisation de \LaTeX avec [MiKTeX](#)⁵ pour la création des fichiers pdf : Choisissez la version "basic" de MiKTeX, et si lors de la première utilisation il manque des paquets lors de la création de vos documents, le logiciel ira les chercher directement sur internet ("install missing packages on the fly"). Il peut y avoir une douzaine de ces paquets manquant lors de la première utilisation de textallion.

4. Cela ne semble **pas** fonctionner avec teTeX fourni avec cygwin sous windows, il est sans doute trop vieux, mais peut-être qu'en installant toutes les dépendances cela fonctionnera. MiKTeX est quand même recommandé.

5. Cela ne semble **pas** fonctionner avec teTeX fourni avec cygwin sous windows, il est sans doute trop vieux, mais peut-être qu'en installant toutes les dépendances cela fonctionnera. MiKTeX est quand même recommandé.

- `Sed` est nécessaire, pour le moment, pour certains remplacement dans les fichiers ePUBs.

L'outil `make` peut être un plus (utilisation d'un makefile optionnel), ainsi que divers outils pour la manipulation de fichiers postscript (`pstopdf`, `psutils` (`pdfnup`, `psbook`), `pdfjam` etc). Veuillez récupérer le "Setup program" indiqué et l'installer. Vous pouvez ensuite indiquer le dossier `C:Files32` comme variable de `PATH` dans windows (propriété système > variables d'environnement). De façon moins "propre", vous pouvez copier le contenu de ce dossier (`make.exe` et quelques dll) dans votre projet courant.

Pour créer des ePUBs, vous aurez besoin d'installer [Calibre](#)

*BSD

Même procédure que sous Linux, à la différence de taper `gmake` à la place de `make` car le makefile utilise GNU `make`. Comme `texlive` n'est pas officiellement supporté par FreeBSD, veuillez utiliser `teTeX` à la place.

Mac OS X

Non testé. Vous devriez pouvoir utiliser les instructions pour BSD plus haut, conjointement avec [macports](#) pour rajouter les logiciels spécifiques.

Utilisation

Vous pouvez vous inspirer du document `examples.t2t` comme base pour votre propre document.

Le fichier `textallion.t2t` est le moteur du système. Pour l'export PDF, il y a également le style `textallion.sty`, qui est indispensable pour son fonctionnement. Il n'y a normalement pas à modifier ces 2 fichiers. Vous pouvez les placer à l'endroit que vous voulez sur votre système, et juste faire référence à `textallion.t2t` depuis votre document de travail. De plus, selon l'export utilisé, il est nécessaire de rajouter des modèles, que l'on trouve dans le dossier "`templates`" : pour le moment `xhtml.html` pour les exports HTML et ePUB, et `latex.tex` pour l'export PDF.

Note: si vous avez utilisé auparavant une ligne dans ce genre : `%!style(tex): /usr/share/textallion/core/textallion`, elle devient maintenant `%!postproc(tex): './' '/usr/share/textallion/'`

- Première méthode :

Pour commencer à créer un document, recopiez l'entête de `examples.t2t` dans un nouveau fichier.

Sous Linux⁶, le script « `textallion.sh` », se charge de faire cela pour vous, à partir des informations entrées en ligne de commande, en créant en plus un fichier `.sty` pour modifier le style du PDF final, et un fichier `.css` pour modifier le style du fichier HTML final (ces 2 modifications étant optionnelles, vous pouvez vous laisser guider par les commentaires dans le fichier de style. Vous pourrez changer simplement la taille de page, la taille du texte affiché (définissant ainsi implicitement les marges, la polices, les espacements. . .).

- Seconde méthode :

Pour commencer un nouveau document, vous pouvez juste créer un fichier texte vide, passer 4 lignes pour l'entête de fichier (en écrivant optionnellement le titre sur la première, votre nom sur la deuxième et la date sur la troisième), et inclure le « moteur » de cette façon, dans la version en ligne :

```
\%!postproc(tex): 'Textallion - documentation' 'titre de votre document'
```

6. ou sous Windows avec les outils cygwin

```

\%!postproc(tex): 'Eric Forgeot' 'votre nom'
\%!postproc(tex): 'documentation, manual, manuel' 'mots clés de votre document'

\%!style(tex): sample.sty
\%!style(xhtml): sample.css

\%!includeconf: textallion.t2t
\%!postproc(tex): '../' '../'

```

Veillez télécharger auparavant `textallion.t2t`, `sample.css`, `sample.sty` et `textallion.sty` depuis <https://textallion.sourceforge.io/> dans votre dossier courant, ou dans un dossier de votre choix. En revanche sous Windows la référence du nom doit être écrit avec des slashes à la place des anti-slash, par exemple `c:/temp/textallion/core/textallion.t2t`

Par la suite, vous aurez sans doute le désir de modifier votre style personnel, pour cela vous pouvez vous référer à la première méthode de préférence.

- Écriture : ensuite, à vous de rédiger votre texte. La syntaxe à utiliser, très simple, se trouve décrite dans le paragraphe suivant.

- Création des documents finaux :

Pour compiler le document final, en html, pdf ou ePUB, veuillez ouvrir un terminal dans le répertoire de votre document, modifiez dans le `makefile` la ligne avec `DOCUMENT = sample_fr` vers votre propre titre de document et tapez au choix :

- `make html` pour générer une page html
- `make slidy` pour générer une présentation html
- `make pdf` pour générer un document \LaTeX puis un fichier pdf. Si la compilation se passe correctement, il n'y aura pas d'erreur affichée, et cela devrait finir par "entering extended mode". La commande de génération du pdf se lance 2 fois pour obtenir la table des matières.
- `make xetex` est une variante de la précédente, pour utiliser `xetex` à la place de `latex`, ce qui apporte l'utilisation complète de l'unicode, des ligatures et autres fonctionnalités avancées. Toutes les options de formatage disponibles pour le rendu avec \LaTeX (`make pdf`) ne sont pas encore supportées.
- `make pdfsmall` pour faire tenir le document avec 2 pages sur une seule (nécessite `pdfjam`, et les liens ne seront plus actifs)
- `make booklet` pour créer une brochure gérant le recto-verso (nécessite `psutils`, et les liens ne seront plus actifs). Les colonnes ne semblent pas bien alignées pour le moment.
- `make epub` pour créer un fichier ePUB (livre électronique), nécessite le logiciel libre `calibre` pour réaliser la conversion à partir du fichier html.

- `make pdfnolatemx` OU `make pdfweb` pour créer un document pdf à partir de la source html, sans passer par \LaTeX et en utilisant un moteur de rendu webkit (`wkhtmltopdf`)
- `make clean` pour nettoyer le dossier
- `make cleanall` pour nettoyer et effacer les fichiers pdf et html générés, avec confirmation (attention si vous avez d'autres documents sous ces formats dans le même dossier)
- `make configuration-update`, pour comparer avec un outil graphique (par défaut `meld`), les fichiers de configuration par défaut pour `makefile`, `css` et `sty` de la dernière version installée de `textallion`, avec votre projet en cours. Vous pourrez ainsi éventuellement mettre à jour plus facilement les dernières modifications rapportées, tout en gardant vos paramètres personnels.
- `make website` pour créer une page d'index destinée à mettre en ligne vos fichiers html, pdf et ePUB, ainsi qu'un qr-code permettant d'importer l'adresse de cette page internet dans un téléphone ou une tablette (en vue d'y envoyer le fichier ePUB par exemple).
- `make all` pour générer html, ePUB, pdf et l'index de site internet et nettoyer les fichiers temporaires

Vous pouvez également taper `export TEXTALLIONDOC=your_document_name && make pdf` (sans l'extension `.t2t` à la fin de `your_document_name`) et cela utilisera une variable d'environnement ce qui vous évitera de modifier votre `makefile` et permettra de scripter vos travaux.

Le fichier `sample.css` est un exemple de mise en forme pour la version html de vos documents. Il est utilisé également dans une forme dérivée pour la création du livre électronique au format ePUB, si nécessaire.

Sous windows, vous pouvez utiliser le fichier `textallion.bat` qui vous demandera le nom de votre document, et sous quel format l'exporter (html ou pdf). Cela vous permet de vous passer du `makefile` et de l'installation de `make` si vous ne l'avez pas déjà, au prix de fonctionnalités réduites (mais la qualité du fichier html ou pdf exporté sera la même). Vous devrez peut-être modifier ou adapter les chemins indiqués. Par défaut on a considéré que le dossier racine de `textallion` était `C:(()` Vérifiez également que ces chemins sont corrects :

- `C:26.exe`
- `C:Files2.9`
- `C:Files2`

Le cas échéant, modifiez le fichier `textallion.bat` pour qu'il soit conforme à l'installation de ces logiciels.

Voici ensuite un rappel des avantages des divers formats de publications proposés :

HTML

Format du « web » par excellence, le HTML permet de s'afficher partout, quasi-universellement, dans les navigateurs internet⁷.

Le style par défaut, à modifier à votre convenance si nécessaire est `sample.css` (dans le dossier `includes`), à moins qu'un script d'initialisation l'ai renommé du nom de votre projet.

Nous avons également inclu par défaut l'affichage d'une police spécifique, récupérée sur les serveurs de google. Dans le cas où l'ordinateur visualisant le document HTML ne serait pas connecté à internet, la police sera automatiquement remplacée par une police serif de base disponible sur tous les navigateur. Et si vous souhaitez modifier ce comportement, soit pour éviter de se connecter sur un service externe, soit pour utiliser votre propre police, soit pour préférer des polices plus classiques (serif, arial etc), il suffit de retirer la partie "CustomFont" dans la ligne `font:1.2em/1.625em CustomFont, serif ;` et le bloc `@font-face`.

PDF

Format pour l'impression et l'affichage à l'écran, le PDF est nécessaire lorsqu'il s'agit d'avoir partout la même mise en page, quelque soit l'ordinateur qui l'affiche, ainsi que pour la création d'un livre destiné à l'imprimeur.

Le style par défaut, à modifier à votre convenance si nécessaire est `sample.sty` (dans le dossier `includes`), à moins qu'un script d'initialisation l'ai renommé du nom de votre projet.

ePUB

Format pour la lecture numérique, l'ePUB est adapté pour l'utilisation dans une liseuse ou sur un téléphone portable : il n'a pas de mise en page fixe, tout comme le HTML et contrairement au PDF, par contre il a l'avantage d'être optimisé pour l'affichage sur de tels écrans, tout en ne présentant qu'un unique fichier contenant les images et les différents chapitres, ce qui est plus commode pour l'utilisateur final.

Le style par défaut, à modifier à votre convenance si nécessaire est `epub.css` (dans le dossier `includes`), à moins qu'un script d'initialisation l'ai renommé du nom de votre projet.

7. Textallion ne supporte que les navigateurs utilisant les standards actuels, comme Firefox, Opera, Chrome etc), aussi l'affichage pourra être dégradés sur certaines versions d'Internet Explorer, a priori surtout les plus anciennes

Notre exportation ePUB est très loin d'être parfaite, on utilise simplement un export xhtml et quelques simplifications, pour la conversion avec le logiciel Calibre. Vous pouvez si nécessaire utiliser un outil comme [Sigil](#) pour reformater ensuite manuellement votre document ou optimiser son apparence.

Syntaxe

La syntaxe générale est celle de txt2tags, à laquelle on a rajouté quelques symboles pour permettre des fonctionnalités avancées.

Rappel et mémo sur txt2tags

Nous rappelons brièvement ici cette syntaxe :

- ****gras**** : **gras**
- *//italique//* : *italique*
- __soulignage__ : soulignage
- --barré-- : barré
- ++ titre ++ (à différents niveaux) : + Partie +, ++ Chapitre ++, +++ Section +++ etc.
- - liste
- + liste numérotée
- "code"
- [image.jpg]
- [lien vers un site http://www.site.com] : lien vers un site
- | table |

Symboles supplémentaires pour la typographie

Les nouveaux symboles sont généralement définis par 4 signes entre accolades. Ainsi on a ces conventions :

- Une lettrine se fait avec le symbole {*~::~} avant la première lettre.
- Un espace moyen entre 2 sections se fait avec *-*- ou {*-*-}
- Un paragraphe plus conséquent avec {-@- }, rajoute une petite feuille aldine, aussi nommée cœur floral 🍀

- Autre version avec `{-/@-}` : ☪
- Une fin de section avec trois étoiles en triangle (cul-de-lampe ou astérisme) se fait avec `{** *}`
- Vous pouvez également avoir ces symboles (centrés et sur une nouvelle ligne) :

Description	symbole	code
lune premier quartier	☾	{)) }
pleine lune	☉	{ () }
lune dernier quartier	☽	{ ((}
soleil	☼	{()}

Il est possible d'entrer ces symboles directement dans votre source si vous le souhaitez, mais ils seront peut-être plus difficiles à lire dans un éditeur de texte. Vous ne pourrez entrer de nouveaux symboles en unicode si vous voulez publier en pdf (à moins de modifier les directives de préprocesseur dans votre document, sur le modèle de ce qui a été fait dans le fichier `textallion.t2t`), car cette version de latex ne supporte pas unicode. Mais la sortie html pourrait le faire.

Certains lecteurs ePUB ne supportent pas ces symboles, aussi ils ont été remplacés par des étoiles * dans l'export ePUB.

Pour signifier des passages importants vous avez également ces symboles :

(+) pour un passage positif (-) pour un passage négatif

(!) pour un passage à faire attention (?) pour une question (i) pour une information générale

{x} (pour une erreur ?)

Notes de bas de page

Les notes de bas de pages sont possibles en entourant ces notes avec `°°` des deux côtés (voir exemple plus bas, et le code source du document pour des exemples plus précis).

Pour plus de logique avec la syntaxe ajoutée à `txt2tags`, il est possible de créer des notes de bas de page avec le symbole `{ °° }` (celui-ci doit rester accolé à la portion de texte).

La version \LaTeX renumérottera automatiquement les notes, tandis que la version html utilisera une étoile pour faire référence à la note.

Pour apporter plus de souplesse dans l'exportation html, qui ne peut générer cette renumérotation, on peut ajouter manuellement à la fin de la note, entre parenthèses, un numéro pour remplacer l'étoile, par exemple :

Ceci est un texte avec une annotation[°]Ceci est la note.(1)[°].

donnera :

Ceci est un texte avec une annotation⁸.

Le symbole ° peuvent être tapé sur les claviers qwerty avec `alt+248` sous Windows, et `alt gr + 0` sous Linux.

Table des matières et Index

La table des matières est automatiquement générée par `txt2tags`. C'est la commande `%%toc` qui indique l'endroit de cette table des matières dans votre document, sinon, par défaut c'est au début du document.

L'index est optionnel. Pour l'activer, il suffit de vérifier que les options `{makeidx}` et soient bien présentes dans le fichier de style `.sty`, et que la commande `%%index` soit appelée à l'endroit où vous souhaitez placer cet index. Les entrées d'index se font avec les mots à indexer entre le symbole `{^}`. Pour indexer sous un autre nom, il suffit d'écrire par exemple `{^}pomme{^fruit}`.

Dialogues, guillemets, espaces insécables et autres règles typographiques

Pour débiter des dialogues, veuillez écrire un triple tiret (`-`), qui générera un tiret cadratin (`—`) dans le document final.

Il est nécessaire de laisser une ligne entre chaque réplique.

Les guillemets à la française peuvent être ajoutés tels quels, mais il est également faisable d'écrire à la place `{" }` et `{ " }` (sans espace avec les mots concernés).

Si vous voulez ajouter un espace insécable dans votre texte, vous pouvez utiliser le symbole `⁂`.

8. Ceci est la note.

Txt2tags offre une réelle souplesse quand aux remplacements de portions de texte, aussi si votre source ne contient pas les espaces nécessaires à la ponctuation française (par exemple avec les points d'interrogation collés aux mots les précédant), vous pouvez utiliser dans l'entête de votre document ces remplacements :

```
%!preproc: ' ' ' ?'
%!preproc: ' ' ' !'
%!preproc: ' ;' ' ;'
%!preproc: '« ' '« '
%!preproc: ' »' ' »'
```

Textallion corrige automatiquement l'espace avant ; par un espace insécable. Mais si le point virgule est collé ; rien ne sera corrigé (vous pouvez le faire manuellement avec une règle preproc).

Textallion ne corrige pas automatiquement les trois points de suspension...

Retours à la ligne

Txt2tags est conçu pour ne pas prendre en compte un simple retour à la ligne dans l'affichage final.

Pour faire un véritable saut de ligne, il faut donc passer 2 lignes, mais il est également envisageable de faire débiter chaque ligne où vous voulez un retour, d'un espace simple (syntaxe rajoutée et spécifique à textallion). Cela peut être ainsi plus pratique et élégant pour afficher de la poésie, qui utilise de nombreux sauts de ligne. En cas de deux espaces ou plus avant le premier mot, cela fera un retour à la ligne suivi d'une indentation de paragraphe (dans la version PDF uniquement), ceci n'est pas un alinéa, c'est juste un effet visuel lié à LaTeX. Pour réaliser un véritable alinéa, il faut passer 2 lignes.

Vous pouvez forcer un retour à la ligne avec `{// ..}`, ou la version simplifiée `// ...`. Si possible, vous devriez essayer malgré tout de garder la syntaxe complète avec les accolades pour garder une unité, mais c'est aussi selon votre goût.

Si vous souhaitez copier et coller rapidement du texte d'internet ou d'une autre source, vous trouverez sans doute ennuyeux de devoir inclure à chaque fois les marques spéciales de retour à la ligne (`*--*`), aussi vous pouvez activer cette option dans l'entête de votre document pour pallier cela :

```
\%!postproc: "\$" '\{//..\}'
```

Mais cela peut conduire à d'autres contraintes, aussi c'est à utiliser avec parcimonie.

Saut de page

Un saut de page se fait avec le symbole `{/...}`.

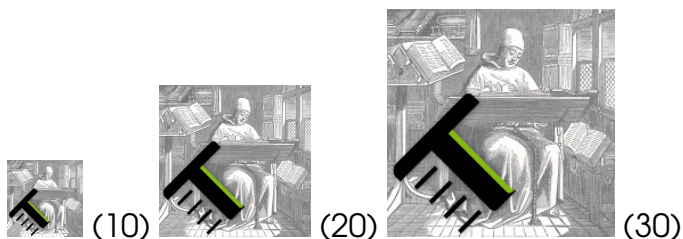
Exposants

On peut utiliser `^^` ou `{ ^^ }` pour entourer la partie à utiliser en tant qu'exposant. Par exemple `2^^ème^^` ou `24^^5^^`.

Taille d'image et alignement

Pour inclure des graphismes, il suffit d'utiliser la syntaxe générale de `txt2tags`, et de rajouter optionnellement accolé à votre image, entre crochets, la taille souhaitée (valeur en mm) :

par exemple : `(chemin_vers/mon_image.png)(24)`



Dans la version html, cela affichera cette valeur en pourcentage de la page.

Il est possible de centrer l'image en utilisant une tabulation avant celle-ci

Pour entourer ou envelopper l'image avec du texte (wrapping), on peut utiliser la forme `(image.png){~~~~}` (image à gauche) ou `{~~~~}(image.png)` (image à droite)

Cette dernière notation vient après l'éventuelle notation de taille.



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim

ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Il est toujours possible d'indiquer l'information de taille d'image, ce qui sera d'ailleurs souhaitable pour maîtriser pleinement l'effet de la vignette.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Liens

Si vous trouvez plus logique d'utiliser la syntaxe (description URL) pour vos liens, vous pouvez l'écrire ainsi, cela sera converti en interne par textallion, vers (description URL).

Site internet de txt2tags

Note : Même si vous écrivez ((description | http://www.monsite-internet.com/URL avec des espaces.html)), cela ne sera pas correctement retraduit en lien avec espaces, puisque cela sera transformé en (description http://www.monsite-internet.com/URL avec des espaces.html) et que seul espaces.html sera considéré comme le lien. Ici le | est juste un repère visuel pour séparer l'URL. Pour ajouter des espaces dans une URL, veuillez indiquer

Rajout de sons, de musiques ou de vidéos

Plus spécialement utile pour le module de jeux textuels (voir plus bas), il est possible de rajouter de la musique avec la syntaxe : (musique.ogg), ce qui s'affichera dans le navigateur sous forme de lecteur de musique compatible balise HTML5 dans les exports HTML. Seul le format ouvert et libre ogg/vorbis est supporté, car contrairement au mp3, il est directement compatible avec la plupart des navigateurs : Firefox, Chrome, Opera.

Exemple de musique :

../media/tbd_musique01.ogg

De la même manière vous pouvez inclure des vidéos au format ogg/theora, avec l'extension .ogv ou .webm à la place de l'extension .ogg : (video.ogv)

Exemple de vidéo (téléchargée depuis un site internet extérieur) :

http://upload.wikimedia.org/wikipedia/commons/7/79/Big_Buck_Bunny_small.ogv

Exemple de vidéo : http://v2v.cc/~j/theora_testsuite/320x240.ogg

Taille du texte

- Vous pouvez **augmenter** la taille du texte en débutant le passage à modifier avec `{ ++ }` et en le terminant avec `{/++ }`.
- Vous pouvez **diminuer** la taille du texte en débutant le passage à modifier avec `{ -- }` et en le terminant avec `{/-- }`.

Zones en italique ou en gras, zone de code

Txt2tags ne permet de formater en italique ou en gras que dans des passages sans retour à la ligne. Si vous souhaitez mettre en italique un passage contenant plusieurs paragraphes, il vous suffit de le débuter avec `{ // }` et de le terminer avec `{/// }`.

De même pour des passages en gras, qu'il vous suffit de débuter avec `{ ** }` et de clore avec `{/** }`.

Du code (écriture non-proportionnelle), s'écrit avec `{####}` et `{/###}`. Le Verbatim de txt2tags (trois fois `) fonctionne bien avec HTML, mais peut poser problème en \LaTeX (sort du cadre de la page si la ligne est trop longue)

Exergue (épigraphe)

En exergue on place traditionnellement une citation avec son auteur sous un titre de chapitre. Nous avons fait en sorte de décaler le texte à droite, avec une marge assez importante à gauche, c'est pour cela que nous avons indiqué comme symboles pour débuter `{ ~}` et pour terminer `{/ ~}`. Il est possible de modifier l'apparence finale en modifiant le style \LaTeX ou le style CSS. Vous pouvez indiquer un auteur à cette citation en l'accolant juste après `{ ~}` ou après `{/ ~}`, soit entre accolades, soit directement.

Veuillez éviter d'écrire sur une seule ligne : `{ ~}Citation{/ ~}`

Ainsi :

```
{ ~}
```


Ce qu'il y a d'admirable dans le fantastique, c'est qu'il n'y a plus de fantastique : il n'y a que le réel.

{/ ~~}André Breton

deviendra :

Ce qu'il y a d'admirable dans le fantastique, c'est qu'il n'y a plus de fantastique : il n'y a que le réel.

ANDRÉ BRETON

Couleurs

Pour illustrer ponctuellement des documents avec du texte en couleur, il est possible d'utiliser la syntaxe `@@COLOR@@nom de la couleur@@Texte à coloriser@@/COLOR@@`. On commence à observer une syntaxe un peu intrusive, mais c'est le prix à payer pour obtenir un peu de couleur, ce type d'indication ne pouvant se symboliser autrement. Cela reste plus simple que leurs équivalents respectifs en \LaTeX ou html.

Les couleurs de base sont :

black blue brown cyan darkgray gray green lightgray lime magenta olive orange
pink purple red teal violet yellow

Et vous pouvez également utiliser les codes couleurs suivants, qui sont compatibles à la fois dans \LaTeX et en HTML (couleurs svg, mais la plupart sont sans doute inutiles) :

AliceBlue AntiqueWhite Aqua Aquamarine Azure Beige Bisque Black
BlanchedAlmond Blue BlueViolet Brown BurlyWood CadetBlue Chartreuse
Chocolate Coral CornflowerBlue Cornsilk Crimson Cyan DarkBlue DarkCyan
DarkGoldenrod DarkGray DarkGreen DarkGrey DarkKhaki DarkMagenta
DarkOliveGreen DarkOrange DarkOrchid DarkRed DarkSalmon DarkSeaGreen
DarkSlateBlue DarkSlateGray DarkSlateGrey DarkTurquoise DarkViolet DeepPink
DeepSkyBlue DimGray DimGrey DodgerBlue FireBrick FloralWhite ForestGreen
Fuchsia Gainsboro GhostWhite Gold Goldenrod Gray Green GreenYellow Grey
Honeydew HotPink IndianRed Indigo Ivory Khaki Lavender LavenderBlush LawnGreen
LemonChiffon LightBlue LightCoral LightCyan LightGoldenrod LightGoldenrodYellow
LightGray LightGreen LightGrey LightPink LightSalmon LightSeaGreen
LightSkyBlue LightSlateBlue LightSlateGray LightSlateGrey LightSteelBlue
LightYellow Lime LimeGreen Linen Magenta Maroon MediumAquamarine
MediumBlue MediumOrchid MediumPurple MediumSeaGreen MediumSlateBlue
MediumSpringGreen MediumTurquoise MediumVioletRed MidnightBlue MintCream
MistyRose Moccasin NavajoWhite Navy NavyBlue OldLace Olive OliveDrab
Orange OrangeRed Orchid PaleGoldenrod PaleGreen PaleTurquoise PaleVioletRed

PapayaWhip PeachPuff Peru Pink Plum PowderBlue Purple Red RosyBrown RoyalBlue
 SaddleBrown Salmon SandyBrown SeaGreen Seashell Sienna Silver SkyBlue SlateBlue
 SlateGray SlateGrey Snow SpringGreen SteelBlue Tan Teal Thistle Tomato Turquoise
 Violet VioletRed Wheat WhiteSmoke Yellow YellowGreen

On peut indiquer une couleur avec son code html à la place de son nom, par exemple #6D7D2E, ce qui donne : Ma belle couleur verte. Par contre LaTeX n'affichera pas cette couleur, cela sera remplacé par du gris.

Centrage de texte

Vous pouvez centrer du texte en commençant une ligne par un | (avec LaTeX uniquement) :

texte centré (mais pas en html)

ou encore en entourant un paragraphe avec `{->--}` et `{-<--}` :

Un bel haiku
 Texte poétique bien centré
 Voilà ce qu'on recherche ici

N'oubliez pas de rajouter une espace avant chaque ligne pour préserver les retours à la ligne.

ou encore avec `@@begin((center))**` et `@@end((center))**`

Un bel haiku
 Autre méthode bien compliquée
 Pour les amoureux du TeX

Troisième possibilité, pour aller avec le symbole de l'exergue, vous pouvez utiliser `{^^}` et `{/^^}`

Colonnes

Vous pouvez utiliser le symbole `{|n|}` pour signifier que vous voulez avoir votre texte sur n colonnes, n étant une valeur comprise entre 0 et 4. On considère que 4 colonnes c'est déjà un maximum pour une présen-

tation de texte. À la fin de la zone à mettre en colonnes, on termine par cette même commande avec $n=0$ ou $n=1$ pour signifier le retour à la normale.

Tables

Les tables sont natives dans `txt2tags`. Mais dans l'export \LaTeX (pdf) par défaut, si une table est trop longue, elle sortira en bas de la page et la fin ne sera pas affichée, aussi nous utilisons le paquet `longtable` et `tabu` au lieu de `longtabu`. Si vous préférez revenir à `longtabu`, veuillez rajouter ceci dans votre en-tête :

```
%!postproc(tex): 'longtable' 'longtabu''
```

```
%!postproc(tex): 'longtabu' 'longtabu''
```

In the case your tables has so many columns that it will print outside the width of the page, use this trick to wrap long texts, for example if your table has 4 columns:

```
%!postproc(tex): '\{\|1\|1\|1\|1\|\}' {|p{3cm}|p{3cm}|p{3cm}|p{3cm}|}
```

*(we must escape the `{` and `|` symbols when using `*proc` in `txt2tags`)*

If your document has only one table, it will be ok. You can also customise the width of each column. If it has more tables and you wish to use different widths, you can use a second trick:

add extra spaces at the beginning of one of the colum, like this:

```
| left | left | left |   right |
```

so this table will be unique to the document and you can use this now:

```
%!postproc(tex): '\{\|1\|1\|1\|r\|\}' {|p{2cm}|p{3cm}|p{5cm}|p{3cm}|}
```

Reminder: tables are aligned like this:

- only one single space around the pipes:

```
| left align | left align | left align |
```

- right align is done with one or more extra space before the text:

```
| left align |    right align | left align |
```

- center align is done with more than one space around the text, with the same quantity of space for both side:

```
| Lorem ipsum dolor | sit amet, consectetur | adipiscing elit. Donec a diam lectus. |
| left align |    right align | center align |
```

Lorem ipsum dolor	sit amet, consectetur	adipiscing elit. Donec a diam lectus.
left align	right align	center align

- Pour centrer vos tables, débutez la ligne par un espace :

```
| Lorem ipsum dolor | sit amet, consectetur | adipiscing elit. Donec a diam lectus. |
| left align |    right align | center align |
```

Lorem ipsum dolor	sit amet, consectetur	adipiscing elit. Donec a diam lectus.
left align	right align	center align

Code \LaTeX complet

Même si ce n'est pas prévu pour, vous pouvez également rajouter du code spécifique à LaTeX, en remplaçant les `par` par des `@@` et les `{` et `}` par respectivement `((` et `))`. Si vous voulez que la commande disparaisse totalement du document html, vous aurez besoin de finir la commande par deux étoiles `**` pour indiquer la fin du code.

Il est également possible, de façon plus simple, d'encadrer votre code latex avec la syntaxe `txt2tags` pour le verbatim (trois `"`) ou le nouveau format `tagged` (trois `"`). Mais contrairement à la syntaxe décrite précédemment, cela inclura le code \LaTeX dans le document html, ce qui n'est pas forcément souhaitable, sauf si vous avez uniquement besoin de l'exportation LaTeX.

Le code source du présent document vous donnera plus d'exemples sur comment rajouter directement des options latex dans votre texte. Par exemple une formule mathématique (qui n'apparaîtra pas dans la version html) :

$$\frac{a}{b} + \sqrt[n]{abcd}$$

En version verbatim :

```
\begin{displaymath}
\frac{a}{b} + \sqrt[n]{abcd}
\end{displaymath}
```

Ou une boîte :

Test de mise en boîte

Ce dernier exemple apparaîtra bizarrement dans la version html car il n'y a pas les ** de fin de commande spéciale.



Exemple des notes :

On pourra ainsi include des notes⁹ (version avec le code de remplacement)

Mais attention, de cette manière, celles-ci n'apparaîtront pas lors de l'export html¹⁰

Il vaut mieux utiliser cette forme pour avoir des notes sauvegardées¹¹

Enfin, il y a également cette forme alternative¹²

Équations LaTeX

9. Essai note de pied de page.

10. Cette note sera perdue en html.

11. Cette note sera gardée dans la version html du document, incluant également du *formatage* éventuel.

12. Uniquement pour garder l'idée des symboles entourés d'accollades. . .

Elles se créent en entourant les équations avec le symbole { \$\$ } (des deux côtés).

Elles seront affichées telles quelles en html. Vous pouvez les visualiser en ligne avec <http://www.codecogs.com/latex/eqneditor.php>

En version en ligne : $\delta = 3 \times \sum X_i^2$, ainsi que $4 - \sqrt{X}$

En version hors ligne :

$$\delta = 3 \times \sum X_i^2$$

Exemple code unix

Si vous avez besoin d'inclure des caractères que l'on trouve dans le code txt2tags, vous pouvez protéger leur exécution en les entourant de double quote ("")

```
find . -print | cpio -o -H newc | gzip -9 > ../initrd.gz
```

Unicode, Ligatures. . .(rendu XeTeX)

L'utilisation de XeTeX à la place de L^AT_EXse fait avec la commande `make xetex` au lieu de `make pdf`. Cela permet d'utiliser des caractères unicodes dans la source :

- japonais :
- persan :
- russe :
- grec :

mais cela ne fonctionne pas avec toutes les langues (à l'étude. . .). Dans les exemples présentés ici, seuls le russe et le grec seront affichés dans la version pdf générée par XeTeX.

Les ligatures :

- roman : fi fl ffl fj sp st ct sl ss fs ff
- italique : *fi fl ffl fj sp st ct sl ss fs ff*

(toutes ne sont pas supportées par les polices par défauts, pour le moment Linux Libertine)

Cela permet également de modifier l'apparence des chiffres (chiffres elzéviens) :
1234567890

(!) Le modèle de document `xetex` ne contient pas de page de garde.

Installation de nouvelles polices pour LaTeX

Vous pouvez trouver des exemples de belles polices pour LaTeX et des moyens pour les installer depuis [The LaTeXFont Catalogue](#). Notamment en utilisant `getnonfreefonts` vous pouvez installer la belle police `dayroman`.

Module de courrier

La commande `make lettre` utilisée sur un modèle tel que celui fourni dans le fichier `samples/exemple_lettre.t2t` permet de créer un courrier (pdf) en utilisant le style Lettre de \LaTeX :

- <http://fr.wikibooks.org/wiki/LaTeX/Lettre>

Module de création de livres-jeux

Cette partie de la documentation a été déplacée dans le fichier `textallion_cyooa_fr`

Exemples de prose

La tour de Nesle

*Il y avait à la tour de Nesle un corps-de-garde auquel se logeait le guet
pendant la nuit.*

BRANTOME.

« Valet de trèfle! — Dame de pique! de gagne! » Et le soudard qui perdait envoya d'un coup de poing sur la table son enjeu au plancher.

Mais alors messire Hugues, le prévôt, cracha dans un brasier de fer avec la grimace d'un cagou qui a avalé une araignée en mangeant sa soupe.

— « Pouah! les chaircuitiers échaudent-ils leurs cochons à minuit? Ventre dieu! c'est un bateau de feurre qui brûle en Seine! »

•

L'incendie qui n'était d'abord qu'un innocent follet égaré dans les brouillards de la rivière, fut bientôt un diable à quatre tirant le canon et force arquebusades au fil de l'eau.

Une foule innombrable de turlupins, de béquillards, de gueux de nuit accourus sur la grève, dansaient des gigues devant la spirale de flamme et de fumée.

Et rougeoyaient face à face la tour de Nesle, d'où le guet sortit l'escopette sur l'épaule, et la tour du Louvre, d'où, par une fenêtre, le roi et la reine voyaient tout sans être vus.

par *Aloysius Bertrand* dans Gaspard de la Nuit.

Voir [exemples.html](#) pour plus d'exemples de textes.

Ressources diverses

- Guide de référence en une page pdf (ou en html).
- <http://lab.arc90.com/experiments/readability/> : permet d'habiller encore différemment une page html (optimisé pour la lecture à l'écran).
- <http://www.framasoft.net/article2225.html> : Règles typographiques
- <http://www.interpc.fr/mapage/billaud/typopao.htm> : autres règles
- Quelques éditeurs de texte supportant la syntaxe txt2tags :
 - <http://kate-editor.org/>
 - <http://www.geany.org/>
 - <http://www.scintilla.org/SciTE.html>
 - <http://projects.gnome.org/gedit/>
- Lecteurs ePUB :
 - <http://www.fbreader.org/> (pour divers OS, dont Android)
 - <http://calibre-ebook.com/> (également pour la gestion des livrels)
 - <http://coolreader.org/e-index.htm> (Android)
 - <http://www.epubread.com/> (extension firefox)
 - <http://readium.org/> (extension chrome)
 - <http://ibisreader.com> (lecture en ligne)
 - <http://www.2epub.com/> (conversion)

Outils concurrents, similaires ou associés

- Ulysses : <http://www.the-soulmen.com/ulysses/>
- Scrivener : <http://www.literatureandlatte.com/scrivener.php>
- Storyist : <http://storyist.com/>
- <http://www.texts.io/>
- Patoline / Patobook (logiciel libre) : <https://patoline.github.io>
- Troff :
 - Mom (logiciel libre) : <http://www.schaffter.ca/mom/mom-01.html>
 - <http://bardinflor.perso.aquilenet.fr/frundis/intro-fr>
- <http://fountain.io/> (pour scénarios)
- <http://asciidoctor.org/> (pour asciidoc)
- [http://nestededitor.sourceforge.net/\(libre\)](http://nestededitor.sourceforge.net/(libre))
 - création d'ePUB : sigil, Writer2ePub, writer2xhtml
 - html :
 - pdf : openoffice, LaTeX, Lout, writer2latex)
 - convertisseur latex 2 epub : <https://github.com/rzoller/tex2ebook> (voir partie serveur pour adapter à textallion)
 - cyoa : renpy, Undum, ChoiceScript, Inform, <http://www.freegameengines.org/gamebookengine/>, <http://homepage.mac.com/siyanlis/Xhoromag/Advelh.htm>, <http://felix.plesoianu.ro/pub/ramus3.html>
- typo : <http://typeplate.com/>
- scribe <https://github.com/schacon/git-scribe> (asciidoc)

License and credits

All textallion code and hacks are released under the [BSD license](#)

Copyright (c) 2008-2020, Éric Forgeot.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

All personal artwork and documentation, if applicable, are licensed under the [CC-BY-SA license](#).

You can contact me about Textallion:

- on the discussion-list: <http://groups.google.com/group/textallion-discuss>
- on the bug-tracker: <https://sourceforge.net/p/textallion/tickets/>
- using this form there: <http://anamnese.online.fr/site2/index.php?id=contact>

Textallion uses [txt2tags](#) as a backend. Txt2tags is covered by the GNU GPL.

Credit is due to:

- Aurelio Jargas and the `txt2tags` team,
- Kovid Goyal for `Calibre`
- nycosweb on Linuxfr for the new name inspiration
- Christophe Chailloleau-Leclerc for the help with the `adventurebook` perl script
- Hans Henderson for the help with some translations
- all contributors to TeX, LaTeX, the W3C standards etc.
- "Tout ce que vous avez toujours voulu savoir sur L^AT_EX sans jamais oser le demander" par Vincent Lozano (édition framabook / In Libro Veritas)
- <http://www.grappa.univ-lille3.fr/FAQ-Latex/6.7.html>
- Html footnote effect taken from <http://andreas.web-graphics.com/footnotes>
- Qr-Code generator from <http://blog.strangeblackhole.com/Un-QR-Code-sur-mon-blog-avec.html>
- Inno Setup for the windows installer
- <http://www.auntiepixelante.com/twine/> for the help with the `twine` syntax
- <http://www.mcdemarco.net/> for the `twee` patch

Changelog

The release filenames are in the form: textallion_year-month-day_artistname.zip. If some additions are made during the same month, there won't be a full release with a new artist's name, but the update's filename will show a new date, for example: textallion_2010-08-13_chasseriau.zip and then textallion_2010-08-15_chasseriau.zip

- Full changelog from the repository: <https://sourceforge.net/p/textallion/code/ref/default>
- 18 mai 2020:
- Frédéric-François d'Andiran release:
 - updated installation script - adding more free audio/video formats - use .desktop with the installer - use tabu instead of long-table for the tables - CYOA: adding templates for inform7, twee - CYOA: new target: Inkle (Ink syntax) - CYOA: new target: Dialog (can render to zcode, html and C64) - now default txt2tags version is for python3 - change mercurial repository from Bitbucket to Sourceforge
- 28 aug 2015:
 - Paul Delaroche release:
 - various bugfixes
 - adding support for wkhtmltopdf (render pdf from html using webkit)
 - use CONVERTindexnommeSTARTcategory!keywordEND for index in latex
 - updated Ramus cyoa
 - adding smallcaps
 - updated ms windows installer
- 18 nov 2013:
 - Léonide Bourges release:
 - CYOA: fixed bold, monospace and image syntax in twee.
 - CYOA: fixed many things in the twee (twine) export so it will work with named chapters
 - CYOA: adding new syntax to allow named chapters: (#name) and simpler numbered chapters: (#number) instead of (number #number)
 - PDF: Fixed some latex display problems.
 - updated windows installer
- 28 july 2013:
 - Suzanne Valadon release:
 - New syntax for a boxed area.

-
- CYOA: updated twee.
 - HTML: small tweak to the CSS.
 - PDF: added microtype feature in LaTeX.
 - PDF: fixed French guillemets in the Lettre template
 - PDF: fixed the TOC in French even for English documents
 - PDF: fixed the headnote which was only displaying "TABLE OF CONTENT" everywhere
 - SHELL: with pdfjam we can remove extra page in the letter template ("make lettre-1page")
- 22 nov 2012:
 - Pierre Puvis de Chavannes release:
 - CYOA: fixed twee, added undum and choicescrpt. Separated doc for CYOA.
 - SHELL (textallion.sh): submenu for new cyoa systems
 - improved exponent. Added subscript.
- 27 june 2012:
 - Félicien Rops release:
 - EPUB: updated CSS, and various ajustement to visually match the pdf version
 - HTML/EPUB: added htsplit (not used yet)
 - HTML: nicer output with smaller toc and colored h2 and h3
 - PDF: fixed table break by page (using longtabu), added instructions to use text wrap in columns.
 - SHELL (textallion.sh): submenu for new document: general doc, letter or cyoa
 - fixed centered tables with blank space before pipe.
- 20 apr 2012:
 - Odilon Redon release:
 - EPUB: added variable for epub info
 - PDF: fixed various \TeX errors
 - PDF: added "lettre" template for writing letters in French
 - PDF: improved general heading appearance (centered, bigger fonts)
 - CYOA: updated Ramus output.
 - English docs were improved, and various other corrections.

- 24 july 2011:
 - Cléo de Mérode release:
 - CYOA: Added inform 7 support
 - Added sound and video using HTML5 (ogg vorbis/theora)
 - Corrected quickref document
- 11 june 2011:
 - Georges Seurat release:
 - CYOA: improved the renpy output (images, end of game)
 - CYOA: added Ramus HTML rendering (<http://felix.plesoianu.ro//index.php/page:S>)
 - .desktop file (for unix desktops)
- 27 mai 2011:
 - Paul Signac release:
 - CYOA: Now textallion can use txt2cyoa for generating HTML, PDF and EPUB gamebooks.
 - CYOA: fade in js effect when displaying new chapter.
 - CYOA : tweecode, hyena and renpy export
 - New logo
- 31 mar 2011:
 - Édouard Vuillard release:
 - Now use templates.
 - Added a SVG sample cover (can generate a personalised png cover with imagemagick)
 - PDF: Fixed wrapping images within paragraphs, using the `fp latex` package.
 - PDF: now the metadata are defined into the `t2t` document, not the `style.sty` file.
 - PDF: adding xetex backend (`make xetex`).
 - PDF: Now the "book" document class is default, with Part, Chapter, Section.
 - Created a windows installer using Inno Setup.
 - Updated most \LaTeX variables, because underscore was not recognised by LaTeX.
 - Updated `textallion.sh`, now can handle updates, installation etc.
- 27 feb 2011:
 - Charles-François Daubigny release:

-
- Double brackets option for links.
 - \LaTeX equations.
 - Index symbol.
 - Corrected the line break markup.
 - Corrected CSS for a better rendering.
 - Corrected table of content for EPUB output.
 - Updated doc.
 - Moved txt2tags to a contrib section
 - Use of [webfonts](#).
 - Auto generation of a homepage with links to the html, pdf, epub and qr-code.
 - make configuration-update to diff between the latest textallion install and your makefile, css and sty project files.
 - Improved initialisation script (textallion.sh).
- 30 dec 2010:
 - [Alexandre-Hyacinthe Dunouy](#) release:
 - New symbols for adding italic and bold area
 - New symbol for adding page break
 - Real footnotes effect for html (css3)
 - Epigraph style
 - French guillemets symbols
 - French doc improved
 - Code markup
 - Line break
 - A script for creating a new skeleton project.
- 30 nov 2010:
 - [Jean-Auguste-Dominique Ingres](#) release:
 - Updated to latest txt2tags 2.6 (and "2.7" devel)
 - New target: slidy, from txt2tags upstream, for making html presentations (using js)
 - New reorganisation of the folders
 - Now the CSS file are included by default, to be more easily portable (this behavior can be changed in the makefile if needed)
 - New symbols for changing the size of the text (only 3 alternatives, normal, bigger, smaller)

- 13 oct 2010:
 - [Gustave Moreau](#) release:
 - Added columns options
 - Added a quick reference cheatsheet
 - Fixed some bugs (sub-lists)
- 21 aug 2010:
 - [Théodore Chassériau](#) release:
 - Corrected line break (added noindent) for LaTeX, when using a space at the beginning of a line.
 - Added non-breaking space symbol
 - Added page break for new section (heading 1) in LaTeX
 - Reorganised samples documents (a new document with all examples was separated from the manual)
 - User may use some unicode symbols directly into the source code (instead of the symbolised version)
 - Non supported symbols in the epub version are replaced by stars
 - Can use the environment variable TEXTALLIONDOC in the makefile
 - Corrected the initial letters for the CSS
 - Separated changelog and license info into external documents.
- 08 jul 2010:
 - [Marcellin Desboutin](#) release:
 - Changed the symbol representations to a more consistent one (4 signs between braces), with less possibility for a clash with existing texts.
 - Fixed page numbering in pdf so it won't be displayed on the first page cover.
- 25 apr 2010:
 - [Théodore Géricault](#) release:
 - Better display for the epub version (fixed alignment and margins)
- 20 mar 2010:
 - [Douanier Rousseau](#) release:
 - Support for colors in html and LaTeX
- 23 feb 2010:
 - [Daumier](#) release:
 - Update of the makefile for use with the latest Calibre
 - Support for image size information

-
- Simplification and improvement of the style for the html css
 - Use of a code name for the releases (French/European painters or artists from the 19ème century)
 - 24 dec 2009:
 - Beginning of a support for packaged version in a system-wide installation (/usr/share/textallion)
 - A few bugfixes and additions.
 - 20 nov 2009:
 - Project renamed from "txt2tex" to "Le TeXTallion",
 - Thanks to nycosweb, ffx and the linuxfr readers for the inspirations.
 - 15 nov 2009:
 - Better support for centered text for writing poetry.
 - Fix for allowing to create a table of content in the pdf output.
 - 28 jun 2009:
 - now the core engine and the samples are separated so it looks more like a real project.
 - Added epub output in the makefile.
 - 06 jan 2009: a few bugfixes
 - 24 dec 2008: initial release

Todo and known bugs

Syntax, backend and docs

- use of (description www.link.com)
- Syntax for creating references, to be inserted into an index at the end
- pb indents
- a graphical GUI for conversion (a simple tk interface?), and a graphical GUI for edition (based on RedNotebook or SciTE for example?). Or a Qt Gui with some kparts using the kate syntax. Something based on <http://qapote.tuxfamily.org/> or <http://latexila.sourceforge.net/>
- When using UTF-8, ensure it's not using UTF-8 with BOM, otherwise the first bytes of the document may get extra characters and it will modify the title, adding something not interpretable with LaTeX, with this kind of error for example: "Unicode char 8: not set up for use with LaTeX."
- English doc is lagging behind the French one:
- Lettrine + French apostrophe
- Lettrine + capital accented letters (À)
- script textallion.sh: update config (makefile)
- create an updater for the windows version
- create desktop icon (start a term if it's launched from the menu)
- rename the "images" folder to "media" (now there is music in it)
- add syntax: turn to, go to, (turn to ####) etc
- special syntax for warning {<!>} and information {<i>}

HTML

- Use templates for html
- Using an embeded free font, like Gentium Basic?
- hidden menu adding link to read the text in
 - <https://www.readability.com/bookmarklets>
 - or <http://www.tidyread.com/widget.html> ?
- Find more ideas from
 - <http://epubzengarden.com/>
 - <http://librairie.immateriel.fr/> (example)
 - and from <http://typekit.com/> ?
- support for <http://labs.cavorite.com/presentacular>
- In midori and android browser the webfont is not displaying aldine leaves and such. (ok in other webkit engines such as chromium)
- footnotes not yet in the slidy version
- In webkit, when clicking a footnote it will scroll down a bit.
- ~~Wrapping + text modification (for ex. reduce size) are not working~~
- ~~Wrapping is not working on the Android browser (looks like fixed width)~~
- Slidy menu are too big

PDF (LaTeX and XeTeX)

- Use templates for LaTeX
- More templates, for different languages.
- Fix the TOC in French even for English documents
- Improving templates for LaTeX, especially the xetex version
- Monospaced content in xetex
- It's not possible to include a .sty file from internet (latex limitation?)
- When using a depth of 4 in the TOC, the title doesn't display (textallion.sty).
- Use book style by default in the future.
- differentiate odd and even pages.
- beamer is broken (doesn't work as expected, needs rework)
- long links don't reflow to a new line
- can't use both lettrine & wrap image
- section headnote are in UPPERCASE and it's ugly (see markright)
- note : if you want to separate chapters on different pages, use "book". If you want to write a memo with every parts alltogether to save space, use "article". Important: article won't support "chapters", so they will be ignored and break the hierarchy you could have made. Use the replacements proposed at the beginning of new documents.

EPUB

- Extra unwanted entries in TOC, such as footnotes and linked documents
- Epub files looks bad with FBReader desktop edition (but ok in Calibre)
- decorations are not centered as expected

Windows

- Easier updates

txt2cyoa

- Clean and simplify the js code for fading
- Renpy-output
- renpy: check use of " === , - and big space
- undum-output?
- Add sound support
- Ramus: ~~works for numbered chapter, but not for named ones~~ (will work with named ++ chapter ++ with the (#name) or (description|#name) syntax, not with ++ 5 ++(name))
- Ramus: Dice will scroll up
- Ramus: footnotes don't work.
- make a backup of the current work ("snapshot") with current date+time
- Dialog: support @bold, @italic span (see <https://linusakesson.net/dialog/docs/io.html>)
- Dialog: support resources such as images