

EXTENSION DE CLASSE T_EX JAZZ-HANDBOOK

DOCUMENTATION — v.0.1H — 12 FÉVRIER 2021

EJAZZ*

RÉSUMÉ

L'extension de classe L^AT_EX « *handbook* » a pour finalité de présenter des manuels de cours et de formation les plus structurés et « propres » possibles. Pour ce faire, appel est fait à un grand nombre d'extensions de style venant consolider cet objectif. Plutôt qu'un discours exhaustif des coulisses, cette documentation se veut pragmatique et opérationnelle. Néanmoins, après avoir présenté les essentiels à l'utilisateur, quelques rouages internes sont également exposés.

SOMMAIRE

1	Propos liminaires et mise en œuvre	2.5	Terminaux et consoles	22
1.1	Prérequis d'installation	2		
1.2	Chargement de l'extension	4	3	Autres fonctionnalités
1.3	Prise en main et premiers pas	5	4	Installation de T _E X Live
2	Interface utilisateur		5	Copyright et licence
2.1	Structure, fontes, francisation et graphisme	7	6	Historique des versions
2.2	Contenus pleine page vs. marge	9	7	Commentaires circonstanciés
2.3	Mise en exergue et signalétique	13	8	Implémentation
2.4	Code et listing	17		Index

* <ejazz.fr@gmail.com>

1 Propos liminaires et mise en œuvre

Cette extension de classe \LaTeX , comme son nom le laisse préjuger, a l'ambition de proposer une mise en page de document pour servir des objectifs de supports écrits de formation et autres manuels technoscientifiques. Effectivement, si on accorde un minimum de soin à la mise en forme d'un document, cela contribue à apporter une plus-value au discours de fond, ne serait-ce que par la sollicitation de la mémoire visuelle et l'appétence à poursuivre sa lecture.

Cette proposition ne prétend en rien d'être parfaite, elle permet surtout de correctement structurer un manuel à l'aide de fonctionnalités et d'environnements dédiés à cet effet : notes de marge et de bas de page, remarques diverses, codes de programmation, exercices et quiz, bibliographie et glossaires, etc.

1.1 Prérequis d'installation

Avant de présenter l'utilisation de la classe « **handbook** », il faut vérifier que l'on dispose de tous les éléments nécessaires pour compiler un document avec succès. Cette extension a été testée comme fonctionnelle avec une distribution \TeX Live mise à jour à la date de publication du présent document. Il faut donc au préalable installer les fontes et les styles n'appartenant pas à la distribution^a \TeX Live.

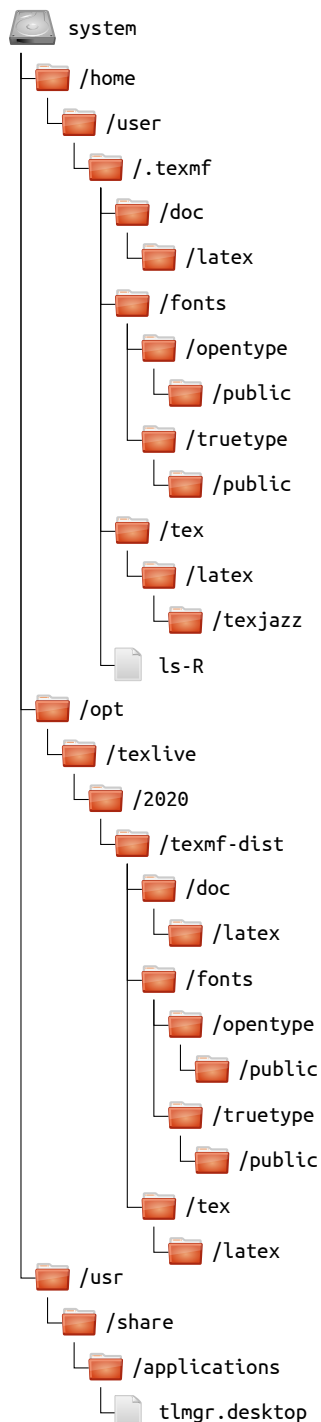
Pour ce qui concerne les fontes, il s'agit de polices de caractères existantes qui ont été légèrement complétées de quelques glyphes courants au moyen du programme FONTFORGE. Il s'agit d'une part, de version étendues des polices de caractères Canter et Ubuntu pour les titres (**XCanter-Bold.otf**), sous-titres (**XCanter-Light.otf**) et fonte à chasse fixe (**UbuntuXMono-R.ttf**) et, d'autre part, de police de caractère pour les mathématiques *Fira Light* (**FiraMaths-Light.otf**) récupérée directement du dépôt du projet car toujours expérimentale. En effet, la classe **handbook** utilise la famille de polices de caractères *Fira Sans* et comme fonte de corps de texte, *Fira Sans Light*. Pour compléter la police *FontAwesome*, quelques glyphes de symboles ont été repris de la police *Fontmifizz* dans une fonte supplémentaire intitulée **xfontawesome.otf** et, enfin, la dernière police à installer est une fonte gothique employée pour l'alphabet du glossaire principal (**QuaeriteRegnumDei.otf**).

Par souci de clarté et d'efficacité, il est conseillé d'installer les fontes dans l'arborescence personnelle créée à l'installation (par défaut sous les systèmes de type UNIX^b dans un répertoire caché souvent nommé **.texmf** — raccourci pour \TeX et METAFONT — à la racine du répertoire utilisateur) et en respectant l'architecture de \TeX Live TDS pour \TeX Directory Structure. Si jamais il y a doublon avec la distribution \LaTeX , l'ordre de priorité est le répertoire du document à compiler, l'architecture personnelle puis enfin celle de la distribution. Ainsi ici, on a :

- **.texmf** ▶ **fonts** ▶ **opentype** ▶ **public** ▶ **firamath** ▶ **FiraMath-Light.otf**
- **.texmf** ▶ **fonts** ▶ **opentype** ▶ **public** ▶ **firamath** ▶ **FiraMath-Regular.otf**

a. Il est supposé ici que l'utilisateur sait installer une distribution \TeX Live et connaît la mise en œuvre d'un document à l'aide d'extensions de classe et de styles. Si ce n'est pas le cas, voir § 4.

b. Pour Windows, se référer à la documentation de \TeX Live ou du portage Mik \TeX plus répandu sur cette plateforme.



- `.texmf>fonts>opentype>public>xcanter>XCanter-Bold.otf`
- `.texmf>fonts>opentype>public>xcanter>XCanter-Light.otf`
- `.texmf>fonts>opentype>public>xcanter>Canter-Outline.otf`
- `.texmf>fonts>opentype>public>xfontawesome>xfontawesome.otf`
- `.texmf>fonts>opentype>public>gothic>QuaeriteRegnumDei.otf`
- `.texmf>fonts>truetype>public>ubuntu-extended>UbuntuXMono-R.ttf`
- `.texmf>fonts>truetype>public>ubuntu-extended>UbuntuMono-R.ttf`
- `.texmf>fonts>truetype>public>ubuntu-extended>UbuntuMono-RI.ttf`
- `.texmf>fonts>truetype>public>ubuntu-extended>UbuntuMono-B.ttf`
- `.texmf>fonts>truetype>public>ubuntu-extended>UbuntuMono-BI.ttf`

Les choses sont similaires pour les extensions de style. En dehors du fichier de classe `texjazz-handbook`, il faut installer les fichiers `texjazz-askreply` pour les exercices et les quiz, `texjazz-assignpoints` pour l'attribution de points, `texjazz-piechart` pour les diagrammes circulaires et, pour disposer de glyphes complémentaires `xfontawesome` (cf. ci-contre arborescence TDS utile), à savoir :

- `.texmf>tex>latex>texjazz>texjazz-handbook.cls`
- `.texmf>tex>latex>texjazz>texjazz-askreply.sty`
- `.texmf>tex>latex>texjazz>texjazz-assignpoints.sty`
- `.texmf>tex>latex>texjazz>texjazz-piechart.sty`
- `.texmf>tex>latex>texjazz>xfontawesome.sty`

La classe `handbook` a elle-même été segmentée en plusieurs fichiers de définition et de configuration de manière à rendre plus lisible le code principal. Concrètement, hormis le fichier de classe, il s'agit de :

- `.texmf>tex>latex>texjazz>texjazz-bibliography.def`
- `.texmf>tex>latex>texjazz>texjazz-bookmark.def`
- `.texmf>tex>latex>texjazz>texjazz-caption.def`
- `.texmf>tex>latex>texjazz>texjazz-coding.def`
- `.texmf>tex>latex>texjazz>texjazz-color.def`
- `.texmf>tex>latex>texjazz>texjazz-common.def`
- `.texmf>tex>latex>texjazz>texjazz-entry.def`
- `.texmf>tex>latex>texjazz>texjazz-glossary.def`
- `.texmf>tex>latex>texjazz>texjazz-graphic.def`
- `.texmf>tex>latex>texjazz>texjazz-coding.def`
- `.texmf>tex>latex>texjazz>texjazz-headfoot.def`
- `.texmf>tex>latex>texjazz>texjazz-listings.def`
- `.texmf>tex>latex>texjazz>texjazz-ltxlistings.def`
- `.texmf>tex>latex>texjazz>texjazz-margin.def`
- `.texmf>tex>latex>texjazz>texjazz-reference.def`
- `.texmf>tex>latex>texjazz>texjazz-shell.def`
- `.texmf>tex>latex>texjazz>texjazz-tabular.def`
- `.texmf>tex>latex>texjazz>titlepage.def`
- `.texmf>tex>latex>texjazz>texjazz-titling.def`
- `.texmf>tex>latex>texjazz>texjazz-toclof.def`
- `.texmf>tex>latex>texjazz>texjazz-typography.def`

Pour que l'arborescence personnelle soit prise en compte, il faut la rafraîchir à l'aide de la commande `texhash` via la mise à jour d'un fichier `ls-R` qui répertorie

les chemins d'accès aux fichiers de l'arborescence ; pour cela, ouvrir un terminal et effectuer la saisie à suivre si c'est l'endroit choisi à l'installation.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:~$ texhash ~/.texmf
```

Par ailleurs, l'extension **handbook** fait appel au paquet **minted** pour l'affichage de certains listing et le rendu des consoles PYTHONTEX. Ces derniers sont dépendants du module PYGMENTS de PYTHON, dont il faut vérifier la présence et la configuration si l'on veut bénéficier des codes couleurs des extensions de l'ensemble TEXJAZZ. Ainsi, après avoir constaté que PYTHON 3 est bien présent sur la station de travail, il faut installer la bibliothèque PYGMENTS et établir les liens symboliques nécessaires avec la distribution¹ L^AT_EX. Pour T_EX LIVE et un système LINUX de type DEBIAN/UBUNTU, il faut aller dans le répertoire **usr** **bin** et saisir la suite de commandes à venir (attention, cela nécessite d'avoir les droits d'administration). Auparavant, on peut vérifier la présence de PYTHON 3 dans le système au moyen de la commande **which**.

1. Ici, les exemples se fondent sur une distribution T_EX LIVE 2020 installée sous **opt** avec un système DEBIAN/UBUNTU.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:~$ which python3
/usr/bin/python3
user@host:~$ cd /usr/bin
user@host:/usr/bin$ sudo apt-get update -y
user@host:/usr/bin$ sudo apt-get install -y python-pygments
user@host:/usr/bin$ sudo ln -s /bin/python3 /bin/python
user@host:/usr/bin$ sudo ln -s /opt/texlive/2020/texmf-dist/scripts/pythontex
/pythontex3.py pythontex
```

De manière à également profiter des glossaires, il faut lier le *script* **xindy** de la distribution L^AT_EX au chemin (PATH) du système d'exploitation.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:/usr/bin$ sudo ln -s /opt/texlive/2020/bin/x86_64-linux/xindy xindy
```

1.2 Chargement de l'extension et options de classe

Pour charger l'extension, il suffit de saisir en préambule ce qui suit.

```
\documentclass{texjazz-handbook}
```

Pour une installation dans les règles de l'art, merci de se référer à la section précédente. La classe « **handbook** » répond aux options ci après exposées. En premier lieu il faut choisir le type de document ainsi que son langage : français ou anglais.

```
\documentclass[options]{texjazz-handbook}
```

Les options de classe se passent directement au document, c'est-à-dire sans système de clef-valeurs (*key-values system* en anglais) et séparées par une virgule.

workbook

Par défaut : **handbook**

Cette option active la rédaction d'un cahier de laboratoire ou d'exercice et non un

manuel de formation. Néanmoins, pour le moment, elle est *très expérimentale* et *non opérationnelle*, par conséquent, *il est fortement conseillé de ne pas l'employer*.

english

Par défaut : **french**

En soi, cette option est explicite. Elle aussi est non fonctionnelle pour le moment.

modelling

(initialement vide)

Réservée à la modélisation informatique (UML et MERISE), cette option charge les extensions de styles appropriées. Elle ne se justifie pas hors de cette application.

showframe

(initialement vide)

Cette option permet d'afficher le cadre de mise en page du document. Elle est pratique en phase rédactionnelle.

printfiles

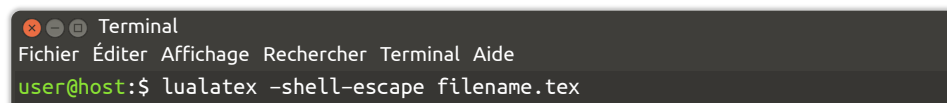
(initialement vide)

Plutôt adressée aux auteurs d'extensions, cette option affiche à la toute fin du document l'ensemble des paquets utilisées par la classe **handbook**.

1.3 Prise en main et premiers pas

Plusieurs extensions utilisées par la classe **handbook**, comme **minted** pour les listings ou **pythontex** pour les consoles PYTHON IDLE, imposent que la compilation soit réalisée avec l'option² **-shell-escape**. Ainsi, en ligne de commande cela donne :

². Dans toute cette documentation, **filename.tex** désigne le fichier maître d'un projet.



```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:~$ lualatex -shell-escape filename.tex
```

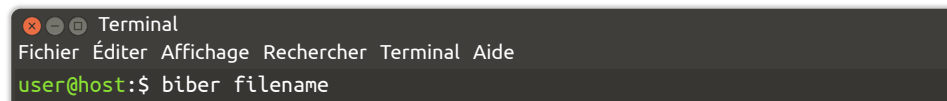
Seul le moteur Lua \LaTeX est supporté (encodage UTF-8, fontes OTF et TTF) et il faut *a minima* quatre compilations pour que toutes les couches graphiques et de références croisées se stabilisent.

Il est bien entendu également possible de configurer un éditeur \LaTeX pour bénéficier d'une interface graphique et d'un environnement de développement intégré, comme **TEXMAKER** (Qt) ou **\LaTeX ILA/GNOME \LaTeX** (Gtk).

La prise en compte de la bibliographie se fait au moyen du moteur **BIBER** et du style **biblatex** sur un fichier d'extension **.bib** au préalable saisi et introduit dans le fichier maître par la commande **\addbibresource**. Par exemple :

```
\addbibresource{./Bibliography/filename.bib}
```

Après la première compilation, il faut lancer le programme **BIBER** sur le fichier *sans extension* pour trier les références bibliographiques émises lors de la première passe et pour que **biblatex** en tienne compte.



```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:~$ biber filename
```

La procédure est similaire pour les glossaires. Les extensions employées sont **glossaries** et **glossaries-extra** et le moteur est **XINDY**. Pour les glossaires « locaux », c'est-à-dire internes au document, ces extensions se chargent automatiquement de leur prise en compte car ils ne sont pas triés, mais juste composés dans l'ordre des

entrées saisies. En revanche, pour le glossaire principal en fin de document, il est à trier par ordre alphabétique et nécessite une phase supplémentaire.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ xindy -L french -C utf8 -I xindy -M "filename" -t "filename.glg"
-o "filename.gls" "filename.glo"
```

Tout ceci est conditionné par la présence non seulement des glossaires locaux dûment répertoriés et, bien entendu du glossaire principal. Il est impératif de déclarer tous les fichiers de glossaire et de les appeler en préambule de document, notamment au moyen de la commande `\makeglossaries` pour ouvrir tous ces fichiers à des fins d'exploitation. Par exemple, on peut considérer en préambule :

```
%- Defining new glossaries for the local glossaries display
\newglossary*{local01}{Glossaire contextuel}
\newglossary*{local02}{Glossaire contextuel}
[...]

\makeglossaries% Open glossary files (mandatory)

\loadglsentries{./Glossaries/filename-glossary.def}
\loadglsentries{./Glossaries/filename-glossary-local01.def}
\loadglsentries{./Glossaries/filename-glossary-local02.def}
[...]
```

Pour introduire des consoles IDLE PYTHON au sein du document — retour de commande lancées depuis le fichier source \LaTeX —, il est fait appel à l'extension `pythontex` et il faut, là aussi, adopter une démarche identique dans le principe pour que l'activation des codes PYTHON s'opère. Pour ce faire, soit créer un lien symbolique entre le script `pythontex3` et le système d'exploitation (cf. supra §1.1), soit donner tout le chemin d'accès au script `pythontex3.py`. Pour une distribution \TeX LIVE installée dans le répertoire `opt>texlive` et un système DEBIAN/UBUNTU :

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ pythontex filename.tex
user@host:$ /opt/texlive/2020/texmf-dist/scripts/pythontex/pythontex3.py
filename.tex
```

Par ailleurs, la classe `handbook` crée automatiquement un répertoire de travail nommé « `PythonTeX` » pour accueillir certains résultats (comme par exemple, des figures élaborées avec le module `MATPLOTLIB`). En anticipant sur la suite de cette documentation, `handbook` crée également un répertoire d'exercices `Exercices-jobname` et un autre pour les quiz `Quizzes-jobname`, où `jobname` indique le fichier maître.

Enfin, le dernier point à connaître est relatif à la coloration syntaxique des codes affichés avec `minted` et `pythontex`. Après avoir vérifié que PYTHON et le module `PYGMENTS` sont bien installés, il faut copier le fichier `jazzcode.py` au sein du répertoire `usr>lib>python3>dist-packages>pygments>styles`. Attention, les droits d'administrateur sont requis. Ce faisant et toutes ces étapes étant mises en œuvre, le document est prêt à être recompilé plusieurs fois³ jusqu'à obtenir le résultat voulu.

3. Selon la nature de ce qu'on rédige ou corrige, il n'est pas nécessaire de compiler quatre fois le document à chaque modification.

2 Interface utilisateur

Les aspects de compilation étant établis, comment alors travailler avec l'extension de classe **handbook** ? De multiples commandes et environnements dédiés à des tâches établies viennent soutenir la rédaction d'un document. Les commandes de sectionnement (`\chapter`, `\section`, `\subsection`, etc.) qui permettent de structurer le document sont supposées connues de l'utilisateur. L'accent est surtout mis sur les nouvelles commandes et environnements directement offerts par la classe **handbook** et ses extensions de style associées.

Néanmoins, quelques concepts fondateurs sont rappelés en conjonction avec les extensions externes de styles qui sont employées. Dans l'ensemble, la classe **handbook** ne fait que reprendre et configurer l'existant.

2.1 Structure, fontes, francisation et graphisme

Outre les commandes de sectionnement traditionnelles d'un document \LaTeX , la classe **handbook** introduit de nouvelles commandes à cet effet. Il s'agit ainsi de pouvoir faire appel à `\appendixchapter` pour les annexes, `\backchapter` pour les *ad-denda* (bibliographie, glossaire et index) et `\overparagraph`, intermédiaire entre les commandes de niveaux `\subsubsection` et `\paragraph`.

`\appendixchapter`[*{optional title for tables of contents}*]{*{appendix chapter title}*}

`\backchapter`[*{optional title for tables of contents}*]{*{backchapter title}*}

Ces commandes indiquent un nouveau chapitre qui, soit est repéré par ordre alphabétique et non numérique (`\appendixchapter`), soit non numéroté (`\backchapter`).

`\overparagraph`[*{optional title for tables of contents}*]{*{overparagraph title}*}

Cette commande introduit un échelon supplémentaire entre les sous-sous-section et les paragraphes. Elle trouve son utilité dans un document dont, selon les cas les sous-sous-sections sont impliquées, mais qu'il est encore nécessaire qu'une segmentation soit disponible car les paragraphes sont trop singuliers.

La conception de documents à l'aide de l'extension de classe **handbook** suit quelques règles simples de mise en forme. Pour une meilleure lisibilité à l'écran et une approche plus moderne à l'impression, la fonte — ou police de caractères — est choisie *sans empattement* (police dite « bâton ») — *sans serif* en anglais. La préférence s'est portée sur la famille de fontes *Fira Sans*, développé à l'initiative de la fondation MOZILLA pour le projet FIREFOX OS, semble-t-il avorté à ce jour.

La famille de fontes *Fira Sans* est une des polices bâton de licence ouverte⁴ des plus complètes en termes de graisses des caractères (*Hair*, *UltraLight*, *Light*, *Regular*, *Medium*, *SemiBold*, *Bold*, *ExtraBold* et *Heavy*) et de glyphes (petites capitales et symboles mathématiques).

Pour la police à chasse fixe — *monotype* en anglais —, utilisée dans les listings ou la transcription de commandes système, la fonte *Fira Mono* possède une chasse trop importante et difficile à maîtriser ; le dévolu s'est donc jeté sur la fonte **Ubuntu mono**. Cela a l'avantage de correspondre aux interfaces graphiques du système d'exploitation UBUNTU MATE 20.04 pris en exemple comme outil de base.

4. SIL Open Font Licence.

5. Cela s'avère être la proposition par défaut de tous les éditeurs de texte disponibles.

Dans les us et coutumes de typographie générale, il est communément admis que la lisibilité optimale d'un texte est obtenue pour des longueurs de ligne d'environ quatre-vingts caractères⁵ au maximum. Ce faisant et pour disposer d'un texte suffisamment aéré, le corps des caractères et la longueur des lignes de texte sont établis pour avoir des lignes de l'ordre de soixante-cinq caractères.

Par conséquent, la mise en page offre une marge suffisamment importante pour accueillir la plupart des illustrations et des notes additionnelles sans pour autant perturber le fil du texte principal.

`\sidenote{<text>}`

`\footnote{<text>}`

`\parnote{<text>}`

6. À l'usage, très majoritairement.

À ce sujet la classe **handbook** permet aussi bien les notes de marge⁶ que de bas^c de page. Il est conseillé de ne pas employer comme ici les deux systèmes en même temps pour ne pas perdre le lecteur. Pour être complet, l'extension **handbook** autorise également les notes de texte ou plutôt de paragraphe* via le paquet **parnotes** pour lequel on renvoie à sa documentation.

* Ceci est une note de paragraphe, qui peut même s'employer dans des titres de sectionnement. La commande `\parnote` s'utilise comme ses homologues; seul l'endroit où placer la note ou les notes définies est laissée au rédacteur avec la commande `\parnotes`.

Sur ces bases, la classe **handbook** emploie donc les polices de caractères *Fira Sans* pour le corps de texte (*Fira Sans Light*) les intitulés de sectionnement (*Fira Sans Book*) et les titres et sous-titres (*Fira Sans* en petites capitales espacées) autres que ceux du document en tant que tel (fontes *Canter* étendues). La gestion des fontes est réalisée au moyen des paquets **fontspec** et **unicode-math**, mais aussi **amsmath**.

Pour ce qui concerne les règles de typographie spécifique au français, la classe **handbook** utilise le système BABEL avec l'extension de langue « **french** ». On renvoie à la documentation de ce paquet pour de plus amples renseignements : espaces fines, énumérations, notes de bas de page, etc.

Toujours au sujet des fontes, la gestion des lettrines de début de chapitre est confiée à l'extension de style éponyme, **lettrine**.

Pour les éléments graphiques, c'est un peu une évidence que de mentionner l'appel aux extensions **graphicx** (import et affichage des illustrations externes) et **tikz** (dessin et mise en œuvre de graphismes composés). Pour compléter cette panoplie, l'extension **handbook** utilise les styles **pgfplots** qui apporte des facilités dans l'affichage de courbes ou de diagrammes et **tcolorbox** pour la gestion d'environnement et boîtes⁷ en couleur. Là encore renvoi est fait aux documentations très complètes de ces paquets remarquables.

7. En \LaTeX , tout ou presque tout est « boîte ».

La couleur est régie par le paquet **xcolor**. Plusieurs couleurs sont définies pour répondre à un code de deux couleurs principales et deux secondaires :

1. un bleu électrique intitulé « *firstcolor* » pour les hyperliens et les éléments graphiques d'agencement du manuel — RGB = (83, 104, 120);

c. Très épisodiquement à l'usage.

2. un **rouge-brun** nommé « *secondcolor* » complémentaire au bleu, mais aussi pour les avertissements et certains encarts — RGB = (153, 25, 25);
3. un **vert canard** (*teal* en anglais) ou « *thirdcolor* » pour les encarts d'explication des questionnaires à choix multiples — RGB = (0, 128, 128);
4. une **palette de marrons** quasi exclusive aux illustrations, dont la référence est appelée « *fourthcolor* ».

La gestion du sectionnement, y compris entête et pied de page, est confiée au paquet **titlesec** et pour la stricte mise en page (largeur et hauteur du texte, des marges, etc.) **handbook** s'appuie sur le paquet **geometry** et définit plusieurs contextes.

\symmetricalpage

\asymmetricalpage

\appendixpage

\referencepage

Ces commandes changent la mise en page au sein du document. Leurs noms sont explicites : **\symmetricalpage** pour les pages de titre, **\asymmetricalpage** pour les pages conventionnelles à deux cotés pair et impair, **\appendixpage** pour les pages d'annexes et **\referencepage** pour celles des références bibliographiques.

Enfin, au sujet des légendes, la classe **handbook** emploie les paquets **caption** et **subcaption** pour leur mise en œuvre (cf. leur documentation pour les détails). Les extensions **hyperref** et **cleveref** sont également utilisées pour respectivement les liens hypertextes et la gestion des références croisées.

2.2 Éléments pleine page *versus* de marge

Hormis les annotations, la classe **handbook** définit plusieurs environnements se plaçant en marge du texte principal, tout comme des environnements de pleine page, marge comprise. Ceux qui bénéficient d'une signalétique particulière sont décrits en section suivante (cf. § 2.3).

La gestion des éléments de marge implique bien entendu la commande \LaTeX **\marginpar**, mais aussi les extensions de style **marginnote** (positionnement absolu sur la page, en utilisant et contrôlant les propriétés du format PDF avec les différents moteurs \LaTeX et non plus avec la commande « flottante » **\marginpar**) et **marginfix** (positionnement « intelligent » des contenus de marge, utilisé avec **\marginpar**). Par ailleurs, pour finement ajuster les marges des pages impaires et paires, il est fait appel aux paquets **changepage** (commandes pour les pages paires/impaires), **ifoddpage** (vérification et conditionnelles de parité des pages).

Des commandes de « bas niveau » sont alors définies pour encapsuler d'autres contenus. C'est surtout le cas pour les éléments de marge.

\begin{fullwidth}[{key-value options}]

Cet environnement de pleine largeur est prévu pour envelopper d'autres contenus rédactionnels. Si ces derniers ne sont que du texte, l'environnement supporte le chevauchement sur plusieurs pages. Si ce sont d'autres entités — figures, tableaux

ou codes —, cette propriété n'est plus vérifiée et il faudra utiliser plusieurs environnements de ce type à chaque nouvelle page ou employer d'autres environnements (cf.infra). Les options sont celles du paquet **tcolorbox**.

\marginelement[*{size}*]{*{contents}*} Par défaut : **\footnotesize**
 Exemple d'utilisation autonome de la commande **\marginelement**. Cette commande peut s'utiliser en autonomie ou comme fondation d'autres commandes. Elle est construite à l'aide la commande du noyau **LaTeX** **\marginpar** et peut se voir comme un raccourci de celle-ci. Si le contenu est du texte, il est justifié et par défaut sa taille est **\footnotesize**.

\margincontents[*{size}*]{*{contents}*} Par défaut : **\footnotesize**
 Exemple d'utilisation en autonomie complète de la commande **\margincontents**. Cette commande est le pendant de **\marginelement** pour des contenus alignés à droite ou à gauche selon la parité de la page. Comme pour **\marginelement**, la commande admet en option un changement de taille des caractères.

\marginnote[*{optional left contents}*]{*{mandatory right or one side contents}*}[*{voffset}*]
 Exemple de contenu affiché au moyen de **\marginnote**. La commande **\marginnote** provient du paquet **marginnote**. Elle est rappelée ici au regard de son importance dans la structure de la classe **handbook**. Son grand intérêt est de pouvoir afficher en marge des contenus qui ne sont pas « flottants ». En effet, la commande **\marginpar** du noyau **LaTeX** est un environnement flottant, au même titre que les environnements **figure** ou **table**. Ors, l'appel d'un élément flottant depuis un autre flottant est impossible. La commande **\marginnote** s'appuie sur les positions absolues sur une page du document PDF, obtenues par les primitives **\pdfsavepos** et **\savepos** du moteur **LaTeX** utilisé (ici **LuaLaTeX**). Pour stabiliser les références il faut deux compilations. Le premier argument optionnel est le contenu de la note de marge dans le cas de documents à deux faces et pour distinguer un contenu en marge gauche du contenu en marge de droite. L'argument obligatoire est pour la marge de droite ou pour la marge des documents à une seule face. Enfin le dernier argument permet d'ajuster la note de marge, soit vers le haut avec une longueur négative, soit inversement vers le bas avec une longueur positive. On remarque que le corps de la police de caractères est laissé à la discrétion de l'utilisateur et que, par défaut, se présente avec la taille **\normalsize**.

Les environnements **figure** et **table** issus des classes **article** ou **book** sur lesquelles est bâtie la classe **handbook** sont laissés intacts. En revanche pour s'affranchir de leur nature d'environnement flottant, plusieurs environnements de pleine page ou de marge sont définis.

```
\begin{jazzfigure}
\begin{jazzfigure*}
\begin{jazztable}
\begin{jazztable*}
```

Les équivalents non flottants des environnements **figure** et **table** sont justes des coquilles vides, prêtes à recevoir des figures dans un cas et des tableaux dans l'autre. Les environnements étoilés sont définis pour proposer un affichage pleine page, couvrant la largeur du texte et celle de la marge. Ces environnements détectent la première figure ou table d'un chapitre afin de réinitialiser les compteurs et de mettre une entrée correcte dans les listes de figures, respectivement de tables. Enfin, les légendes s'adaptent automatiquement aux types d'environnement.

```
\begin{jazzfigure*}
\includegraphics[width=\linewidth]{example-image-a}
\end{jazzfigure*}
```

```
\begin{jazzgraphic}
```

```
\begin{jazzgraphic*}
```

À l'instar des environnements de ce type, ils sont non flottants, de pleine largeur de texte ou de page et possèdent une déclinaison pour les marges. Qu'ont-ils alors de si différent des environnements « **jazzfigure** » ? Simplement, ils sont dévolus aux illustrations ou images sans comptabilisation de leurs occurrences, ni dans le flux du texte, ni dans une liste de « graphiques ». Leurs légendes sont harmonisées avec les autres, mais plus à voir comme des commentaires du support visuel.

```
\begin{jazzcode}
```

```
\begin{jazzcode*}
```

```
\begin{jazzlisting}
```

```
\begin{jazzlisting*}
```

De la même manière et avec les mêmes propriétés, sont définis des environnements destinés à envelopper des listings de codes (voir § 2.4). Ils sont un peu redondants au regard des environnements de code de programmation, mais leur intérêt est de pouvoir mener une comparaison côte à côte de deux codes différents en y adjoignant éventuellement une légende (voir exemple ci dessous).

```
\begin{jazzcode*}
\caption{\label{code:1}Comparaison de code en \textsc{Java} et en \textsc{Python}..}
\hfill
\begin{codebox}[width=0.45\linewidth, nobeforeafter]{java}
public class HelloWorld {
    public static void main (string[] args) {
        System.out.println("Hello world!");
    }
}
\end{codebox}
\hfill
\begin{codebox}[width=0.45\linewidth, nobeforeafter]{python}
print("Hello world!")
\end{codebox}
\end{jazzcode*}
```

CODE 1 — Comparaison de code en *JAVA* et en *PYTHON*.



```
1 public class HelloWorld {
2     public static void main (string[] args) {
3         System.out.println("Hello world!");
4     }
5 }
```



```
1 print("Hello world!")
```

Pour ce qui concerne les éléments de marge, chaque environnement de pleine page présenté dans cette section en possède un ou plusieurs, soit comme commande, soit en tant qu'environnement.

`\sidefigure`[*{optional caption}*]{*{mandatory contents}*}

`\sidetable`[*{optional caption}*]{*{mandatory contents}*}

Ce type de commandes est traditionnel en \LaTeX . Il s'agit d'afficher en marge un contenu, avec ou sans légende. Les dites légendes, si elles existent, sont comptabilisées dans la liste des figures ou des tables. De même, si elles sont les premières d'un chapitre, elles déclenchent automatiquement une mise à jour de leurs compteurs et des entrées de leurs listes respectives de figures et de tables.

`\sidegraphic`[*][*{optional caption}*]{*{mandatory contents}*}{*{optional copyright text}*}


`\sideimage`[*][*{optional caption}*]{*{mandatory contents}*}{*{optional copyright}*} *text*[*{voffset}*]

`\onesidegraphic`[*][*{optional caption}*]{*{mandatory contents}*}{*{optional copyright text}*}

`\onesideimage`[*][*{optional caption}*]{*{mandatory contents}*}{*{optional copyright text}*}[*{voffset}*]

Assignées aux éléments graphiques de marge autres que les environnements de figures, avec ces commandes on peut indiquer le copyright ou la licence de publication et, éventuellement la source de l'illustration exposée. L'option étoilée rajoute le symbole de copyright sur le coté de la marge de l'illustration et le quatrième argument en complète l'identification des sources. La commande `\sidegraphic` est faite pour être appelée du texte principal et s'avère construite avec la commande `\marginpar` du noyau \LaTeX , alors que la commande `\sideimage` peut être appelée depuis un environnement flottant et, bâtie sur la commande `\marginnote`, bénéficie d'un paramètre d'ajustement vertical en cas de chevauchements mal aisés de contenus de marge. Une illustration du propos est donnée ci-contre. Les commandes `\sidegraphic` et `\sideimage` sont destinées aux documents à deux faces avec une bascule entre les pages impaires et paires (marge à droite et à gauche). Les annotations de copyright et de sources figurent alors du coté des marges internes. Quant à elles, les commandes `\onesidegraphic` et `\onesideimage`, affichent ces informations en marge externes, soit à droite pour les documents avec une face et marge à droite, En revanche, l'affichage se fait à gauche de la marge pour les documents à marge inversée (à gauche) (voir la commande \LaTeX `\reversemarginpar`).

`\sidevideo`[*{optional caption}*]{*{PATH/URL link filename}*}{*{image}*}<*{pictogram}*>

Comme son nom l'indique, cette commande permet de placer en marge un espace réservé à une vidéo à l'aide d'une image (cf. infra). Le pictogramme associé dans la liste des vidéos est le symbole . Ces deux objets graphiques sont modifiables. Cette commande est un enrobage de la commande `\href`{*{PATH/URL}*}{*{default image}*} du paquet `hyperref`. Cela signifie que le lancement du visionnage se fait via le lecteur multimédia installé par défaut sur la station de travail et non au sein du document PDF. Il est préférable d'utiliser l'environnement `marginvideo`, plus souple et permettant de visionner la vidéo dans le corps du document (voir infra).

`\launchvideo`{*{PATH/URL link filename}*}

Tout comme la commande précédente, `\launchvideo` se fonde sur `\href`, c'est-à-dire qu'un lecteur externe va lire la vidéo, mais cette fois seul le chemin d'accès à



Image affichée avec la commande (`\onesidegraphic`).

la vidéo est à saisir car en interne le pictogramme est figé (🖥️) en marge. Certes, la commande peut s'utiliser telle quelle, mais elle est surtout prévue pour s'intégrer aux environnements de vidéos.

```
\movie[⟨options⟩][⟨poster text⟩][⟨movie filename⟩]
```

La commande `\movie` provient du paquet `multimedia`, lui-même partie prenante de l'ensemble BEAMER. Renvoi est fait à la documentation de BEAMER pour le détail des options. Là aussi, cette commande s'utilise en autonomie, mais ici elle est plutôt à intégrer dans le corps de l'environnement `marginvideo` (cf. infra). Il faut garder à l'esprit que c'est le chemin entier jusqu'au nom du fichier qu'il faut saisir. L'URL ou le chemin seuls ne suffisent pas avec cette commande.

```
\begin{marginfigure}[⟨optional caption⟩][⟨optional voffset⟩]
```

```
\begin{marginfigure}[⟨optional caption⟩][⟨optional voffset⟩]
```

```
\begin{marginfigure}[⟨optional caption⟩][⟨optional voffset⟩]
```

```
\begin{marginfigure}[⟨optional caption⟩][⟨voffset⟩]<{pictogram}>
```

Ces environnements sont les pendants des macros présentées plus haut. Là encore ce sont des coquilles vides de contenu, mais spécifiant la nature des environnements pour avoir des légendes appropriées à chaque type. Ainsi, il est possible de rajouter une légende en option ou de manière explicite et traditionnelle dans l'environnement lui-même. Bien entendu, il ne faut pas mettre des doubles légendes, mais d'aucun s'apercevra de sa méprise à la relecture du résultat final. Des exemples d'utilisation des environnements `marginfigure` et `marginvideo` sont donnés ci-dessous.



FIGURE 1 — Fauve expectatif.

```
\begin{marginfigure}{-6.25cm}
\includegraphics[width=\marginparwidth]{fauve.jpg}
\caption{Fauve expectatif.}
\end{marginfigure}
```



VIDÉO 1 — Légende en action.






```
\begin{marginvideo}[\\label{vid:1}Légende en action.]{-4.5cm}
\movie[width=\marginparwidth,showcontrols]%
{\\includegraphics[width=\marginparwidth]%
{./film-strip-dark-electric-blue.png}}%
{https://www.learningcontainer.com/wp-content/uploads/2020/05/sample-mp4-
file.mp4}%
\launchvideo{https://www.learningcontainer.com/wp-content/uploads/2020/05/
sample-mp4-file.mp4}
\end{marginvideo}
```

2.3 Mise en exergue et signalétique

Pour mettre en valeurs certaines parties importantes, remarques ou compléments, la classe `handbook` dispose de plusieurs environnements et commandes. Là encore, deux logiques sont à considérer : dans le corps de texte ou en marge.



Ceci est un encart de page pour une remarque additionnelle importante au regard du propos principal.






`\begin{remark}[\langle key-value options \rangle][\langle optional title \rangle]\langle symbol \rangle` Par défaut : **Remarque**, 
Ainsi, en catégorie de pleine page, ces compléments sont relatifs à l'environnement polyvalent **remark** qui indique différentes emphases au moyen d'un des pictogrammes optionnels (cf. supra) :  par défaut (`\faEye`) pour une note indépendante,  (`\faExclamationTriangle`) pour une alerte,  (`\faQuestion`) dans le cas d'une interrogation ou  (`\faInfoCircle`) pour une information. Les options sont celles introduites par le paquet **tcolorbox**, ni plus, ni moins.

`\sideremark[\langle optional title \rangle][\langle text/stuff \rangle][\langle optional pictogram \rangle]` Par défaut : **Remarque**, 



NOTE

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris.

Une commande de marge polyvalente avec signalétique est aussi disponible. Elle correspond à la petite sœur de l'environnement de plein texte déjà mentionné ci-dessus. Sa syntaxe est relativement explicite : elle permet de choisir le titre (par défaut « Remarque ») et le symbole de signalétique à prendre en compte (par défaut « , `\faEye`). Comme pour sa grande sœur et au-delà d'une simple remarque, on peut notifier un avertissement (« , `\faExclamationTriangle`), une interrogation (« , `\faQuestion`), une information (, `\faInfoCircle`), une définition (, `\faTag`) ou tout autre signalement, même sans titre, ni symbole d'appel.

`\cautionleft[\langle position \rangle]\langle frame color \rangle\langle text \rangle[\langle optional title \rangle]`

`\cautionright[\langle position \rangle]\langle frame color \rangle\langle text \rangle[\langle optional title \rangle]`

`\caution[\langle position \rangle]\langle frame color \rangle\langle text \rangle[\langle optional title \rangle]`

De même, les notes de rédaction et certains avertissements sont exprimés en marge de manière explicite dans leur intitulé (voir ci-après). La commande `\cautionleft` se destine à la marge de gauche et `\cautionright` est dévolue à la marge de droite. Enfin, la commande `\caution` est à employer dans les documents où la présentation des pages paires diffère des pages impaires (option de classe *twoside* dans le jargon \LaTeX) essentiellement par inversion des marges, des entêtes et des pieds de page. Dans tous les cas de figure, la syntaxe de ces commandes permet de choisir la position de l'encart, la couleur de son cadre et, bien entendu son intitulé (cf. options ci-dessous). L'intérêt de cette macro est d'être polyvalente et, par construction — en sous-jacent le paquet **marginnote** —, qu'il est possible de l'appeler dans n'importe quel contexte, y compris depuis des environnements flottants.

position = `[b|c|t]` Par défaut : `[c]`
Positionnement optionnel de l'encart : en bas `[b]` pour *bottom*, au centre (par défaut) `[c]` pour *centered* ou en haut `[t]` pour *top*.

frame color = `<valid color>` Par défaut : `<firstcolor>`
Couleur du cadre entourant l'encart.

text = `{\langle mandatory text \rangle}` (initialement vide)
Texte de l'encart.

optional title = `{\langle optional title \rangle}` Par défaut : **Avertissement** !
Titre représentant l'encart.

NOTE DE LA RÉDACTION

Ceci est un encart de marge, à utiliser pour indiquer une note de la rédaction ou un avertissement spécifique.

1

Des notes numérotées et de pleine page sont également proposées pour faire état d'un conseil important ou d'une procédure à suivre particulièrement significative. On pense notamment à des directives de programmation.

`\begin{linewidthnote}`

L'environnement ci-dessus ne comporte pas d'option et se contente d'incrémenter son compteur à chaque utilisation. De ce point de vue et compte tenu que son emploi reste sporadique, cet environnement ne se réinitialise pas à chaque chapitre, mais se comptabilise tout au long du document.

`\begin{marker}[\langle key-value options \rangle]<\langle line width \rangle>\{\langle optional title \rangle\}`

L'environnement à venir permet lui aussi une mise en exergue d'une note. La largeur de ligne et le titre sont optionnels. Pour les autres options se référer au manuel de **tcolorbox** (par exemple pour changer de couleur).

TITRE OPTIONNEL

De manière similaire, une définition ou une note à retenir peut s'inscrire dans cet environnement particulier (en mode « *Post-It* »).

POUR ALLER PLUS LOIN...

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Des environnements de pleine page adressant résumés ou commentaires spécifiques sont disponibles. Comme pour la commande d'avertissement `\caution`, il sont polyvalents, car toutes les options sont contrôlables parce que le paquet ayant servi leur conception est **tcolorbox**.

`\begin{gofurther}[\langle key-value options \rangle]\{\langle modified title \rangle\}`

Par défaut : **Pour aller plus loin**

`\begin{gofurther*}[\langle key-value options \rangle]\{\langle modified title \rangle\}`

Par défaut : **Pour aller plus loin**

Par défaut le titre est « *Pour aller plus loin...* » et sa couleur la seconde couleur principale définie par la classe **handbook**. Mais tout est modifiable à condition de connaître la syntaxe adoptée par le paquet **tcolorbox** dont la documentation est exhaustive. La différence de l'environnement étoilé⁸ est d'afficher son contenu sur toute la largeur de page, marge comprise. À titre d'illustration, l'environnement ci-dessous a changé de titre et de couleur par rapport aux valeurs par défaut.

8. Souvent, la syntaxe de la classe **handbook** propose des environnements étoilés qui signifient simplement que l'affichage est étendu pleine page, marge comprise.

```
%- Environnement pleine ligne de texte
\begin{gofurther}
\lipsum[2]
```

```
\end{gofurther}

%- Environnement pleine page, marge comprise
\begin{gofurther*}[colbacktitle=firstcolor]{Commentaires circonstanciés}
\lipsum[3]
\end{gofurther*}
```

COMMENTAIRES CIRCONSTANCIÉS

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

“
If it looks like a duck and quacks
like a duck, then it must be a duck.

— Attribué à James W. RILEY

Plus anecdotique, il reste néanmoins une mise en exergue supplémentaire de « type littéraire ou journalistique », destinée aux citations textuelles d'auteurs externes à la rédaction d'un manuel, quelles soient pleine page ou en marge.

“
If it looks like a duck and quacks like a duck, then it must be a duck.

— Attribué à James W. RILEY

“If it looks like a duck and quacks like a duck, then it must be a duck.”

— Attribué à James W. RILEY






`\blockquotation`[*<optional author(s)>*]{*<text>*}[*<offset>*]

`\sidequote`[*<optional author(s)>*]{*<text>*}[*<offset>*]

Ces commandes parlent d'elles-mêmes : faire appel à la saisie d'un texte à citer en prévoyant de manière optionnelle de lui attribuer un (des) auteur(s). Le décalage en largeur est fourni pour un ajustement fin (par défaut –18 mm à adapter).

`\begin{citequote}`[*<optional author(s)>*]{*<optional bibliographic reference>*}

Avec cet environnement on peut au choix l'utiliser sans citer d'auteur (pas de premier, ni de second argument), en attribuant un auteur (premier argument sans le second) et en assignant la citation à un auteur avec renvoi vers les références bibliographiques (second argument ou les deux, auquel cas le premier sert à rien).





Question signalétique, il reste encore à mentionner les divers fichiers de documents externes au manuel mais qui lui sont liés. En effet, la classe **handbook** distingue les types de fichiers qui sont cités avec des pictogrammes qui représentent la nature du fichier, à savoir :  texte ou LIBREOFFICE WRITER,  PDF,  page Web,  vidéo ou diaporama et  pour un lien externe explicite.

`\hrefdoc`[*<optional pictogram>*]{*<optional replacement text>*}>{*<link>*}{*<text>*}

Par défaut : 

`\pdfdoc`[*<optional pictogram>*]{*<path/link>*}{*<text>*}

Par défaut : 

<code>\pdflink[⟨optional pictogram⟩]{⟨path/link⟩}{⟨text⟩}</code>	Par défaut : 
<code>\pdfwatch[⟨optional pictogram⟩]{⟨path/link⟩}{⟨text⟩}</code>	Par défaut : 
<code>\textdoc[⟨optional pictogram⟩]{⟨path/link⟩}{⟨text⟩}</code>	Par défaut : 
<code>\webdoc[⟨optional pictogram⟩]{⟨path/link⟩}{⟨text⟩}</code>	Par défaut : 

Une mosaïque de commandes est directement adressée à chaque type de document : texte, page Web, fichier PDF et ainsi de suite... Toutes ces commandes créent une entrée dans une « liste de documents », chacune avec le pictogramme de la police de caractères *FontAwesome* qui lui est associé (cf. supra). Toutes ces commandes se fondent sur la macro `\href` du paquet **hyperref**, laquelle se prête aisément à ouvrir tout document externe selon son type MIME, s'il est connu.

2.4 Code et listing

Essentiellement deux paquets permettent de régir les listings de code de programmation avec \LaTeX ; le plus ancien **listings** et le plus puissant **minted**. La classe **handbook** fait appel aux deux pour cette tâche car, même si cela fait doublon, le premier est facilement configurable pour définir ou étendre la syntaxe des langages et le second pour ses performances, au-delà du fait qu'il est aussi imposé par l'extension **pythontex**, deux paquets maintenus par le même auteur (voir infra).

Il est bon de savoir que ces deux extensions sont également bien gérées par le paquet **tcolorbox**, ce qui offre pour les listings toutes les possibilités de conception graphique qui lui sont connues.

La classe **handbook** s'appuie sur ces trois paquets pour les listings de code. Sont ainsi définis des environnements généraux qui, par option, supportent divers langages de programmation. Les langages connus sont dans l'ordre alphabétique BASH, C, JAVA, JAVASCRIPT, PYTHON et, bien entendu \LaTeX , mais il est relativement facile d'étendre ces fonctionnalités à de nouveaux langages. Dérivé du module PYGMENTS, à lui seul le paquet **minted** en supporte plus de trois cents.

Avant de passer en revue les environnements de codes, il est à connaître qu'un environnement « **listing** » dit « flottant » est automatiquement créé par l'extension **minted**. Sa nature de flottant, comme les figures et les tables (voir § 2.3), ne permet pas toujours de le placer à l'endroit voulu. Par conséquent, cet environnement a été redéfini pour être « fixe ». Renvoi est fait à la documentation de l'extension **minted** pour appréhender toutes les options possibles.

<code>\begin{listingbox}[⟨key-value options⟩]{⟨code language⟩}</code>	Par défaut : <code>numbering lines=false</code>
<code>\begin{listingbox*}[⟨key-value options⟩]{⟨code language⟩}</code>	Par défaut : <code>numbering lines=false</code>
<code>\begin{listing}[⟨key-value opt.⟩]{⟨code language⟩}[⟨optional title⟩][⟨opt. caption⟩]<⟨opt. label⟩></code>	

Pour les environnements **listingbox**, la numérotation des lignes de codes est absente par défaut (comportement inverse des environnements de code à venir) et, là encore, les options supplémentaires correspondent à celle de l'extension **tcolorbox**. Le second argument, obligatoire, spécifie un langage de programmation connu de la classe. Les environnements **listingbox** peuvent être insérés dans un autre environnement ou appelés de manière autonome. Comme déjà évoqué, la version étoilée de

l'environnement s'étend sur toute la largeur de la page, marge y comprise. La différence de ces environnements de listing de code avec les autres, est qu'il ne sont pas associés à un logotype rappelant le langage de programmation et ne sont ni titrés, ni comptabilisés dans la liste des codes. Pour cela, il doivent être insérés dans un environnement `listing`. En revanche l'environnement `listing` est la redéfinition du flottant initialement défini par l'extension `minted`. En dehors de l'affichage d'un pictogramme, sa syntaxe est strictement la même que celle des environnements `code` et `code*` présentés ci-après (voir exemples). Pour faire concis, les environnements `listingbox` et `listing` sont plutôt prévus pour des extraits de code non numérotés alors que les environnements `codebox` et `code` sont envisagés pour des codes plus longs. C'est à l'utilisateur de faire son choix.

```
\begin{listingbox}{java}
public class HelloWorld
{
    public static void main (string[] args)
    {
        System.out.println("Hello world!");
    }
}
\end{listingbox}
```

```
public class HelloWorld
{
    public static void main (string[] args)
    {
        System.out.println("Hello world!");
    }
}
```

```
\begin{listing}{java}[Premier code \textsc{Java} avec titre intégré]
public class HelloWorld
{
    public static void main (string[] args) {
        System.out.println("Hello world!"); // Mon premier programme Java
    }
}
\end{listing}
```

```
1 public class HelloWorld
2 {
3     public static void main (string[] args) {
4         System.out.println("Hello world!"); // Mon premier programme Java
5     }
6 }
```

CODE 2 — Premier code JAVA avec titre intégré

```

\begin{listing}{java}[Autre code \textsc{Java} avec légende.]
public class HelloWorld {
    public static void main (string[] args) {
        System.out.println("Hello world!");
    }
}
\end{listing}

```

CODE 3 — Autre code JAVA avec légende.

```

1 public class HelloWorld {
2     public static void main (string[] args) {
3         System.out.println("Hello world!");
4     }
5 }

```

`\begin{codebox}[(key-value options)]{<code language>}`

`\begin{codebox*}[(key-value options)]{<code language>}`

`\begin{code}[(key-value options)]{<code language>}[<optional title>][<opt. caption>][<opt. label>]`

`\begin{code*}[(key-value options)]{<code language>}[<optional title>][<opt. caption>][<opt. label>]`

La classe `handbook` offre également la possibilité d'avoir des encarts de code intégrant un logotype et un titre optionnel. Dans ce dernier cas, le titre est enregistré dans la liste des codes, si on souhaite éventuellement l'exposer dans un document, surtout de type `book`. À l'instar de `listingbox`, `codebox` peut s'employer au sein d'un environnement flottant ou en autonomie. Les environnements `code` et `code*` s'affichent comme leurs homologues soit respectivement en largeur de texte, soit en pleine largeur. Ils s'emploient en autonomie et leur label est automatique, de la forme `\label{<code:number>}`. Ainsi, pour utiliser leur labellisation, on procède traditionnellement avec la commande \LaTeX `\ref{<code:number>}` ou avec le paquet `cleveref` pour afficher automatiquement le type d'environnement, ici « code » `\cref{<code:number>}`. Par exemple le premier et le dernier codes jusqu'à présent insérés dans le document sont les codes 1 et 3.

```

\begin{code}{java}[Second code \textsc{Java} avec titre intégré]
public class HelloWorld
{
    public static void main (string[] args)
    {
        System.out.println("Hello world!"); // Mon deuxième programme Java
    }
}
\end{code}

```



```

1 public class HelloWorld
2 {
3     public static void main (string[] args)

```

CODE 4 — Second code JAVA avec titre intégré

```

4  {
5      System.out.println("Hello world!"); // Mon deuxième programme Java
6  }
7  }

```

```

\begin{code*}{java}[Autre code \textsc{Java} avec légende et en pleine page.]
public class HelloWorld
{
    public static void main (string[] args) {
        System.out.println("Hello world!");
    }
}
\end{code*}

```

CODE 5 — Autre code JAVA avec légende et en pleine page.

```

1  public class HelloWorld
2  {
3      public static void main (string[] args) {
4          System.out.println("Hello world!");
5      }
6  }

```



```

1  print("Hello world!")

```

CODE 6 — Enfin du Python : modeste mais concis

À noter que pour ce qui concerne le paquet `listings`, il propose un environnement brut (sans sur-couche de `tcolorbox`) intitulé `lstlisting`; c'est l'*alter ego* de l'environnement `listing` de `minted`. Renvoi est fait à la documentation de l'extension de style `listings` pour le détail des options.

CODE 7 — PYTHON clôture la séance avec un code en pleine page et chevauchant trois pages différentes.

```

1  from scipy.io import wavfile
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import csv
5  from itertools import zip_longest
6  import math
7
8  # Discrete Fourier Transform
9  import scipy.fftpack
10 from numpy.fft import fft
11
12 # Windows filtering DSP
13 from scipy.signal import get_window
14 from numpy.lib import stride_tricks
15

```

```

16 # CSV file manipulation
17 import csv
18 from itertools import zip_longest
19
20 fs, signal = wavfile.read('guitar_F2.wav')
21
22 # Waveform
23 Nt = len(signal) # Number of samples
24 duration = Nt / fs
25 time = np.linspace(0, duration, Nt)
26
27 stored_signal = signal[0:8191]
28 stored_time = time[0:8191]
29
30 plt.plot(stored_time, stored_signal)
31 plt.title('Waveform')
32 plt.savefig('waveform.svg', bbox_inches='tight')
33 plt.close()
34
35 waveform_data = [time, stored_signal]
36 export_data = zip_longest(*waveform_data, fillvalue = '')
37 with open('waveform.csv', 'w', newline='') as waveform_file:
38     wr = csv.writer(waveform_file, delimiter=',') # comma delimiter (default)
39     wr.writerow(["Time", "Amplitude"])
40     wr.writerows(export_data)
41 waveform_file.close()
42
43 # Spectrum
44 def plotspectrum(data, rate, start, stop):
45     spectrum = scipy.fftpack.fft(data[start:stop])
46     n = spectrum.size
47     freq = np.arange(n)*1.0/n*rate
48     spectrum = (1.0/n)*np.abs(spectrum)
49     spectrum = spectrum/spectrum.max()
50     n = round(n/2)
51     print ('N/2 = ', n)
52     freq = freq[n:]
53     spectrum = spectrum[n:]
54     print(spectrum.size)
55     plt.title('FFT')
56     plt.plot(freq, spectrum)
57     plt.savefig('fft.svg', bbox_inches='tight')
58     plt.close()
59
60 plotspectrum(signal, fs, 0, 4096)
61
62 # Windowed and sliced spectrum
63 def windowslicehamming(data, rate, fftbins, databins):
64     data = data[:databins-1]

```

```

65     datasize = data.size
66     wdata = np.fft.rfft(data * get_window('hamming', datasize), n=fftbins)
67     wdata = np.abs(wdata)
68     wdatamax = wdata.max()
69     wdata = 20*np.log10(wdata) - 20*math.log10(wdatamax)
70     return wdata
71
72 # 3D view
73 def plot3D(data, rate):
74     wdata = windowslicehamming(data, rate, 2**13, 2**14)
75     fig = plt.figure()
76     ax = fig.gca(projection='3d')
77     ax.set_xlim(0, 10)
78     #ax.set_ylim(0, 10)
79     ax.set_ylim(10, 0)
80     ax.set_zlim(0, 10)
81     X = np.arange(0, 5, 0.25)
82     Y = np.arange(0, 5, 0.25)
83     Z = np.arange(0, 5, 0.25)
84     #X, Y = np.meshgrid(X, Y)
85     ax.plot(X,Y,Z)
86     #ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=plt.cm.hot)
87     ax.set_xlabel('X')
88     ax.set_ylabel('Y')
89     ax.set_zlabel('Z')
90     #ax.set_zlim(-100, 100)
91     plt.savefig('window_3D.svg')
92     plt.close()
93
94 plot3D(signal, fs)

```

Si la classe **handbook** utilise **minted** pour les langages de programmation, **listings** est employé pour le pseudo-code d'algorithmique ou \LaTeX en lui-même. La raison provient du fait qu'il est facile d'ajuster les mots-clefs et la syntaxe d'un langage ou encore de simuler des consoles de terminal (cf.infra) avec le paquet **listings**.

2.5 Terminaux et consoles

L'extension **handbook** bénéficie également d'environnements qui simulent un terminal (*shell*), un environnement IPYTHON ou bien *notebook* JUPYTER et une console PYTHON IDLE.

`\begin{ubuntu}[<shell name>]`

Par défaut : **Terminal**

`\begin{ubuntu*}[<shell name>]`

Par défaut : **Terminal**

`\setuser[<shuser=user name, shhost=host, shcolor= color, shprompt char=char, shdirectory=dir.>]`

L'environnement de terminal est celui d'une distribution Ubuntu MATE 20.04 installée pour le français. Par défaut, le titre de la fenêtre du *shell* est « **Terminal** », mais il peut être changé. L'environnement étoilé est défini pour afficher le terminal en

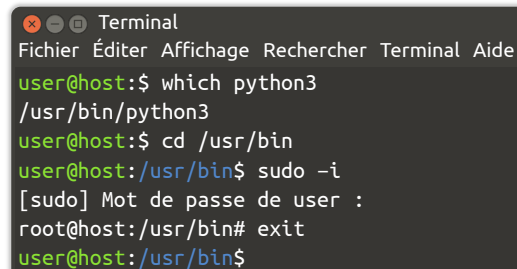
pleine page; c'est parfois utile si on veut simuler une console IPYTHON au sein du SHELL. La commande `\setuser` permet de configurer l'invite de commande ou *prompt* par un système de clefs-valeurs exposé ci-après.

<code>shuser</code>	= <code><user name></code>	Par défaut : <code>user</code>
<code>shhost</code>	= <code><host name></code>	Par défaut : <code>host</code>
<code>shcolor</code>	= <code><shell color></code>	Par défaut : <code>shellusercolor</code>
<code>shprompt char</code>	= <code><character></code>	Par défaut : <code>\$</code>
<code>shdirectory</code>	= <code><directory></code>	Par défaut : <code>~</code>
<code>root</code>	= <code>root (style)</code>	Par défaut : <code>shuser=user, shhost=host, shcolor= white, shprompt char=#</code>
<code>user</code>	= <code>user (style)</code>	Par défaut : <code>user, host, shellusercolor, \$</code>

`\startconsole`

Cette dernière commande permet de forcer l'abandon de l'invite de commande pour simuler la réponse du système à une instruction (voir ci-dessous le code \LaTeX et le résultat obtenu — le caractère d'échappement est le sigle « `$` »). En effet, en interne, chaque ligne de terminal commence par l'invite de commande précédemment introduite. La réponse de la console doit de même être saisie entre deux caractères d'échappement pour qu'elle soit prise en compte sans s'afficher.

```
\setuser{user}
\begin{ubuntu}
which python3 $\startconsole$
/usr/bin/python3 $\setuser{user}$
cd /usr/bin $\setuser{user, shdirectory=/usr/bin}$
sudo -i $\startconsole$
[sudo] Mot de passe de user : $\setuser{root, shdirectory={\color{white}/usr/bin}}$
exit $\setuser{user, shdirectory=/usr/bin}$
$\startconsole$
\end{ubuntu}
```



```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ which python3
/usr/bin/python3
user@host:$ cd /usr/bin
user@host:/usr/bin$ sudo -i
[sudo] Mot de passe de user :
root@host:/usr/bin# exit
user@host:/usr/bin$
```

```
\setuser{user, shdirectory=}
\begin{ubuntu*}
mkdir programming
cd programming $\setuser{user, shdirectory=/programming}$
python $\startconsole$
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
```

```

[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 20*30
600
>>> a=10
>>> print(a)
10
>>> exit() $ \setuser{user, shdirectory=/programming}$
$ \startconsole$
\end{ubuntu*}

```

```

Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ mkdir programming
user@host:$ cd programming
user@host:/programming$ python
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 20*30
600
>>> a=10
>>> print(a)
10
>>> exit()
user@host:/programming$

```

La simulation d'un interpréteur PYTHON au sein d'un terminal Ubuntu n'est qu'une déclinaison particulière de l'environnement déjà vu. Il est juste présenté pleine page pour éviter les retours à la ligne, de la même manière qu'un terminal réel si la fenêtre n'est pas assez large.

`\begin{ipythonshell}[\langle shell name \rangle]` Par défaut : **Terminal**

`\begin{ipythonshell*}[\langle shell name \rangle]` Par défaut : **Terminal**

Comme précédemment, un interpréteur IPYTHON peut se simuler en reprenant les réels codes couleur de la station de travail.

`\ipythonuserprompt{\langle user \rangle}{\langle host \rangle}{\langle directory \rangle}{\langle prompt symbol \rangle}`

`\ipythontext{\langle text \rangle}`

`\ipythonpromptin{\langle prompt in number \rangle}`

`\ipythonpromptout{\langle prompt out number \rangle}`

`\ipythonpromptdots`

Associées à l'environnement `ipythonshell`, il existe cinq commandes de configuration de la console IPYTHON qui correspondent aux différents cas d'invite de commande de l'interpréteur. Le sigle « \$ » est celui du caractère d'échappement.

```

\begin{ipythonshell*}
$ \ipythonuserprompt{user}{host}{~/programming}{\$} \textcolor{white}{ipython}$

```



```

\ipythontext{Python 3.8.2 (default, Apr 27 2020, 15:53:34)}$
\ipythontext{Type `copyright`, `credits` or `license` for more information}$
\ipythontext{IPython 7.13.0 -- An enhanced Interactive Python. Type `?` for help.}$

\ipythonpromptin{1}$ 20*30
\ipythonpromptout{1}$ \ipythontext{600}$
\ipythonpromptin{2}$ a=10
\ipythonpromptin{3}$ print(a)
\ipythontext{10}$
\ipythonpromptin{4}$ exit()
\ipythonuserprompt{user}{host}{~/programming}{\}$
\end{ipythonshell*}

```

```

Terminal
Fichier Éditer Affichage Recherche Terminal Aide
user@host:~/programming$ ipython
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
Type `copyright`, `credits` or `license` for more information
IPython 7.13.0 -- An enhanced Interactive Python. Type `?` for help.

In [1]: 20*30
Out[1]: 600
In [2]: a=10
In [3]: print(a)
10
In [4]: exit()
user@host:~/programming$

```

`\begin{nbjupyterin}[<key-value options>]{<entry number>}`

`\begin{nbjupyterout}[<key-value options>]{<entry number>}`

Dans la droite ligne des autres environnements présentés, il s'agit juste ici de rendre compte des entrées/sorties similaire à la forme des *notebooks* JUPYTER.

```

\begin{nbjupyterin}{1}
20 * 30
\end{nbjupyterin}
\begin{nbjupyterout}{1}
600
\end{nbjupyterout}
\begin{nbjupyterin}{2}
a=10
print(a)
\end{nbjupyterin}
\begin{nbjupyterout}{2}
10
\end{nbjupyterout}
```

```

In[1] 20 * 30
Out[1] 600

```

```
In[2] a=10
      print(a)
```

```
Out[2] 10
```

```
\begin{idleshell}[\langle shell name \rangle]
```

Par défaut : **Python 3.8.2 Shell**

```
\begin{idleshell*}[\langle shell name \rangle]
```

Par défaut : **Python 3.8.2 Shell**

Pour poursuivre la simulation des interpréteurs `PYTHON` tels qu'ils se présentent dans la réalité, cet environnement est sans équivoque dédié à IDLE — *Integrated Development and Learning Environment*. La version étoilée est là encore l'extension de pleine largeur de l'environnement. En revanche, la particularité de cet environnement est de pouvoir embarquer une console IDLE avec l'extension `PYTHONTEX` qui, *via* son environnement `pyconsole` rend les choses très agréables et relativement puissantes à utiliser puisqu'on délègue à `PYTHON` la restitution de la programmation depuis `LATEX`.

```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (default, Apr 27 2020, 15:53:34)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> 20*30
600
>>> a=10
>>> print(a)
10
```

```
\begin{idleshell*}[before skip=2pt, after skip=8pt]
Python 3.8.2 (default, Apr 27 2020, 15:53:34)\par
[GCC 9.3.0] on linux\par
Type "help", "copyright", "credits" or "license()" for more information.
\begin{pyconsole}
20*30
a=10
print(a)
\end{pyconsole}
\end{idleshell*}
```

Les divers types de simulation de console exposés s'avèrent pratiques dans un contexte de découverte de l'environnement de travail et de propos didactiques sur celui-ci. À la longue, il s'avère relativement fastidieux de procéder ainsi pour la rédaction d'un document au volume conséquent, en saisissant chaque ligne une à une, source d'erreur et de coquilles.

C'est donc tout l'intérêt de l'extension `PYTHONTEX` qui offre l'automatisation d'une console IDLE. Notons que l'utilisation qui est faite de `PYTHONTEX` est ici élémentaire et se borne essentiellement à n'utiliser que l'environnement `pyconsole`. La puissance de cet outil n'est pas actuellement exploitée à sa juste valeur. On pense notamment à la possibilité d'avoir recours aux fonctions graphiques *via* `MATPLOTLIB` et la console `PYTHONTEX` `pylab`.

`\begin{idleconsole}[<key-value options>]`

`\begin{idleconsole*}[<key-value options>]`

Pour des raisons de redondance et de place, il est inutile d'intégrer l'environnement `pyconsole` dans une console complète avec `idleshell`. En revanche, conserver une mise en page propre à `pyconsole` se définit par l'environnement `idleconsole`. Cela permet un affichage seyant et sans superflu (cf. infra).

```
\begin{idleconsole}[before skip=2pt, after skip=8pt]
\begin{pyconsole}
20*30
a=10
print(a)
\end{pyconsole}
\end{idleconsole}
```

```
>>> 20*30
600
>>> a=10
>>> print(a)
10
```

3 Autres fonctionnalités

De multiples fonctionnalités sont passées sous silence. Toutefois, certaines commandes peuvent d'être pertinentes pour l'utilisateur final. Il en va d'abord des commandes de mise en forme du texte, typographie et graisse des polices de caractères.

`\lightboldfont`

Par défaut : **Fira Sans Regular**

`\titlefont`

Par défaut : **XCanter Bold**

`\subtitlefont`

Par défaut : **XCanter Light**

`\titlingfont`

Par défaut : **Fira Sans Book**

`\titlingspacedfont`

Par défaut : **Fira Sans Book**

`\tocfont`

Par défaut : **Fira Sans Regular**

`\boxtitlefont`

Par défaut : **Fira Sans Book**

`\glossaryheaderfont`

Par défaut : **Quaerite Regnum Dei**

Un certain nombre de familles de fontes supplémentaires sont définies et attribuées à des tâches spécifiques. Leurs différents noms sont assez explicite pour ne pas les détailler. D'autant plus que pour l'utilisateur final, elles sont accessibles par des commandes d'appel (voir infra) ou employées en contextes exposés précédemment.

`\lightbf{<text>}`

`\spacedlowsmallcaps{<text>}`

`\spacedsmallcaps{<text>}`

`\textttl{<text>}`

On pourra remarquer à l'usage que la fonte à caractères gras n'est pas *stricto sensu* employée au profit d'une graisse dite « *Regular* ». En effet, les caractères gras sont proscrits pour n'être pas assez « discrets ». Puisque la fonte principale est de graisse « *Light* » (choisie pour des raisons de lisibilité et de moindre fatigue à la lecture, de rendu et d'économie en cas d'impression), ils sont empruntés aux fontes « *Regular* » dont dispose la famille de fontes *Fira Sans*; d'où la commande `\lighthbf` si on veut absolument employer des caractères gras. Pour bien distinguer les titres, les fontes ne sont là encore pas en gras mais en petites capitales dont les espaces entre glyphes sont étirées. L'accès à cette possibilité dans le texte est fournie par ces commandes.

<code>\subtitle{<subtitle>}</code>	(initialement vide)
<code>\authorref{<author reference>}</code>	(initialement vide)
<code>\copyrightname{<text>}</code>	(initialement vide)
<code>\copyrightsymbol{<text>}</code>	(initialement vide)
<code>\docversion{<text>}</code>	(initialement vide)
<code>\versiondate{<text>}</code>	(initialement vide)
<code>\doclogo{<scale>}{<image file>}</code>	(initialement vide)
<code>\docaddlogo{<scale>}{<image file>}</code>	(initialement vide)

Aux champs traditionnels de référencement du document que sont `\title`, `\author` et `\date`, la classe **handbook** définit un ensemble de commandes complémentaires.

Pour générer les tableaux, la classe **handbook** utilise quasiment toutes les extensions de style existantes. Cela peut paraître redondant mais cela permet l'emploi de fonctionnalités complémentaires. Ainsi, on peut citer **array**, **booktab**, **colortbl**, **longtable**, **makecell**, **multirow**, **tabu**, **tabularx** et **threeparttable**. On renvoie à la documentation de chacun de ces paquets pour de plus amples renseignements.

Un paquet important pour la classe **handbook** est **multicol**. En effet, il est à la base de la table des matières. Si l'utilisateur veut l'employer dans son document, il est possible d'afficher le texte en plusieurs colonnes (voir documentation).

Un autre paquet charnière de la classe **handbook** est **titlesec** pour les titres de sections, accompagné sous forme d'option des extensions de style **titleps** pour la gestion des entêtes et pieds de page, ainsi que de **titletoc** pour la table des matières et les listes de figures, tables, codes, exercices, quiz, documents et vidéos.

Citons enfin les paquets de style auxquels **handbook** fait appel pour compléter ses fonctionnalités. Il s'agit de **piechart** pour les diagrammes circulaires, mais surtout de **askreply** et **assignpoints** pour les exercices et les quiz, affublés de la possibilité d'attribution de points (chaque extension possède sa propre documentation).

4 Installation de T_EX Live

Pour pouvoir utiliser la classe **handbook** et ses extensions, il faut faire appel à la dernière version existante de la distribution⁹ T_EX Live. On peut récupérer son script d'installation sur le site officiel : <https://www.tug.org/texlive/>.

⁹. Pour MacOS, il existe la distribution MacT_EX et pour Windows MikT_EX.

10. Il est supposé que l'installation est réalisée par quelqu'un qui a les droits d'administration et que l'outil puisse être accessible à tous les utilisateurs du système, même s'il n'y en a qu'un seul.

Pour une installation *via* Internet — vivement conseillée car le dépôt le plus proche sera automatiquement choisi —, il faut ensuite aller dans la page de téléchargement et récupérer le fichier d'archive de l'installation de la dernière version à jour qui correspond au système d'exploitation voulu : **install-tl-unx.tar.gz** pour les UNIX y compris LINUX, ici DEBIAN/UBUNTU.

Une fois le fichier décompressé, il faut aller dans le répertoire fraîchement créé et nommé **install-tl-<date>** puis lancer¹⁰ l'utilitaire d'installation en PERL **perl ./install-tl -gui**, pour avoir une interface plein texte (voir simulation ci-après).

Ensuite, il faut naviguer dans les différentes options pour choisir les paquets à installer. En effet, une distribution T_EX LIVE représente de l'ordre de sept gigabits et tout n'est pas à installer. L'économie est à accomplir sur les paquets linguistiques. Ainsi, il suffit amplement de se concentrer sur le français et l'anglais au profit d'autres options de styles plus utiles en contexte professionnel.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ cd Téléchargements
user@host:/Téléchargements$ wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
user@host:/Téléchargements$ tar zxvf install-tl-unx.tar.gz
user@host:/Téléchargements$ cd install-tl-20210212
user@host:/Téléchargements/install-tl-20210212$ sudo perl install-tl -gui
===== TeX Live installation procedure =====

===== Letters/digits in <angle brackets> indicate <=====
===== menu items for actions or customizations <=====
= help> https://tug.org/texlive/doc/install-tl.html <=====

Detected platform: GNU/Linux on x86_64

<B> set binary platforms: 1 out of 16

<S> set installation scheme: scheme-full

<C> set installation collections:
    40 collections out of 41, disk space required: 7164 MB

<D> set directories:
    TEXDIR (the main TeX directory):
        /usr/local/texlive/2020
    TEXMFLOCAL (directory for site-wide local files):
        /usr/local/texlive/texmf-local
    TEXMFSYSVAR (directory for variable and automatically generated data):
        /usr/local/texlive/2020/texmf-var
    TEXMFSYSCONFIG (directory for local config):
        /usr/local/texlive/2020/texmf-config
    TEXMFVAR (personal directory for variable and automatically generated data):
        ~/.texlive2020/texmf-var
    TEXMFCONFIG (personal directory for local config):
        ~/.texlive2020/texmf-config
```

```

TEXMFHOME (directory for user-specific files):
~/texmf

<O> options:
[ ] use letter size instead of A4 by default
[X] allow execution of restricted list of programs via \write18
[X] create all format files
[X] install macro/font doc tree
[X] install macro/font source tree
[ ] create symlinks to standard directories

<V> set up for portable installation

Actions:
<I> start installation to hard disk
<P> save installation profile to 'texlive.profile' and exit
<Q> quit

Enter command:

```

L'interface présente alors plusieurs options :

- l'item **** pour la plateforme (**binary**) n'est pas à modifier (sauf administrateur système), car le système d'exploitation est automatiquement détecté ;
- par défaut, l'option **<S>** pour « schéma » propose une installation complète, laquelle ne se justifie pas sauf besoin explicite et représente un gain notable en place mémoire. Pour pouvoir sélectionner finement ce que l'on installe on se déplace dans la section **<S>** pour cocher l'option **<j>** (*custom scheme*), puis revenir au menu principal (option **<R>**) ;
- en sélectionnant l'option **<C>** pour collections, on détermine la réelle politique d'installation. Comme précisé plus haut, l'essentiel des paquets linguistiques peuvent être désélectionnés pour ne garder que les collections répertoriées sous les appellations suivantes : 'Essential programs and files', 'BibTeX additional styles', 'TeX auxiliary programs', 'Additional fonts', 'Recommended fonts', 'Graphics and font utilities', 'Additional formats', 'Humanities packages', 'US and UK English', 'Other European languages', 'French', 'Mathematics, natural sciences, computer science packages', 'LaTeX fundamental packages', 'LaTeX additional packages', 'LaTeX recommended packages', 'LuaTeX packages', 'Graphics, pictures, diagrams', 'Plain (La)TeX packages', 'PSTricks', 'Publisher styles, theses, etc.', 'XeTeX and packages'. La place occupée en mémoire passe ainsi de 7 164 MB à 5 255 MB ;
- le troisième item a trait aux répertoires d'installation. Par défaut, il s'adresse aux administrateurs système en mettant T_EX Live dans les répertoires de l'arborescence **usr ▶ local**. C'est une question de choix, mais on peut également tout installer dans une partition **opt ▶ texlive**. De plus, on peut configurer directement son arborescence personnelle dès l'installation, au sein du répertoire caché **▶ .texmf** et non pas visible par défaut. On reprend cette configuration dans la simulation ci-dessus ;
- enfin le dernier item concerne les options. Il faut bien s'assurer que la deuxième et la troisième option sont bien cochées, sinon la classe **handbook** ne pourra

pas fonctionner (interaction autorisée avec un terminal). Les autres options sont relatives aux documentations et aux sources des paquets. Elle sont utiles notamment comme exemples pour apprendre \LaTeX ;

```

Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
=====
Directories customization:

<1> TEXDIR:      /opt/texlive/2020
    main tree:   /opt/texlive/2020/texmf-dist

<2> TEXMFLOCAL:  /opt/texlive/texmf-local
<3> TEXMFSYSVAR: /opt/texlive/2020/texmf-var
<4> TEXMFSYSCONFIG: /opt/texlive/2020/texmf-config

<5> TEXMFVAR:    ~/.texlive2020/texmf-var
<6> TEXMFCONFIG: ~/.texlive2020/texmf-config
<7> TEXMFHOME:   ~/.texmf

Note: ~ will expand to $HOME (or to %USERPROFILE% on Windows)

Actions:
<R> return to main menu
<Q> quit

Enter command:

```

En retournant au menu principal, on peut alors lancer l'installation (commande `<I>`), et prendre une pause café le temps que la foultitude des fichiers se téléchargent et se configurent.

Pour avoir une distribution $T_{\text{E}}X$ Live opérationnelle, il faut à ce stade introduire son accès en complétant la variable d'environnement `PATH`, chemin de recherche des programmes sur la station de travail. Cette inclusion peut se faire à la fin du fichier caché `~/.bashrc.`, situé à la racine du répertoire utilisateur.

```

</> export PATH=${PATH}:/opt/texlive/2020/bin/x86_64-linux
      export MANPATH=${MANPATH}:/opt/texlive/2020/texmf-dist/doc/man
      export INFOPATH=${INFOPATH}:/opt/texlive/2020/texmf-dist/doc/info

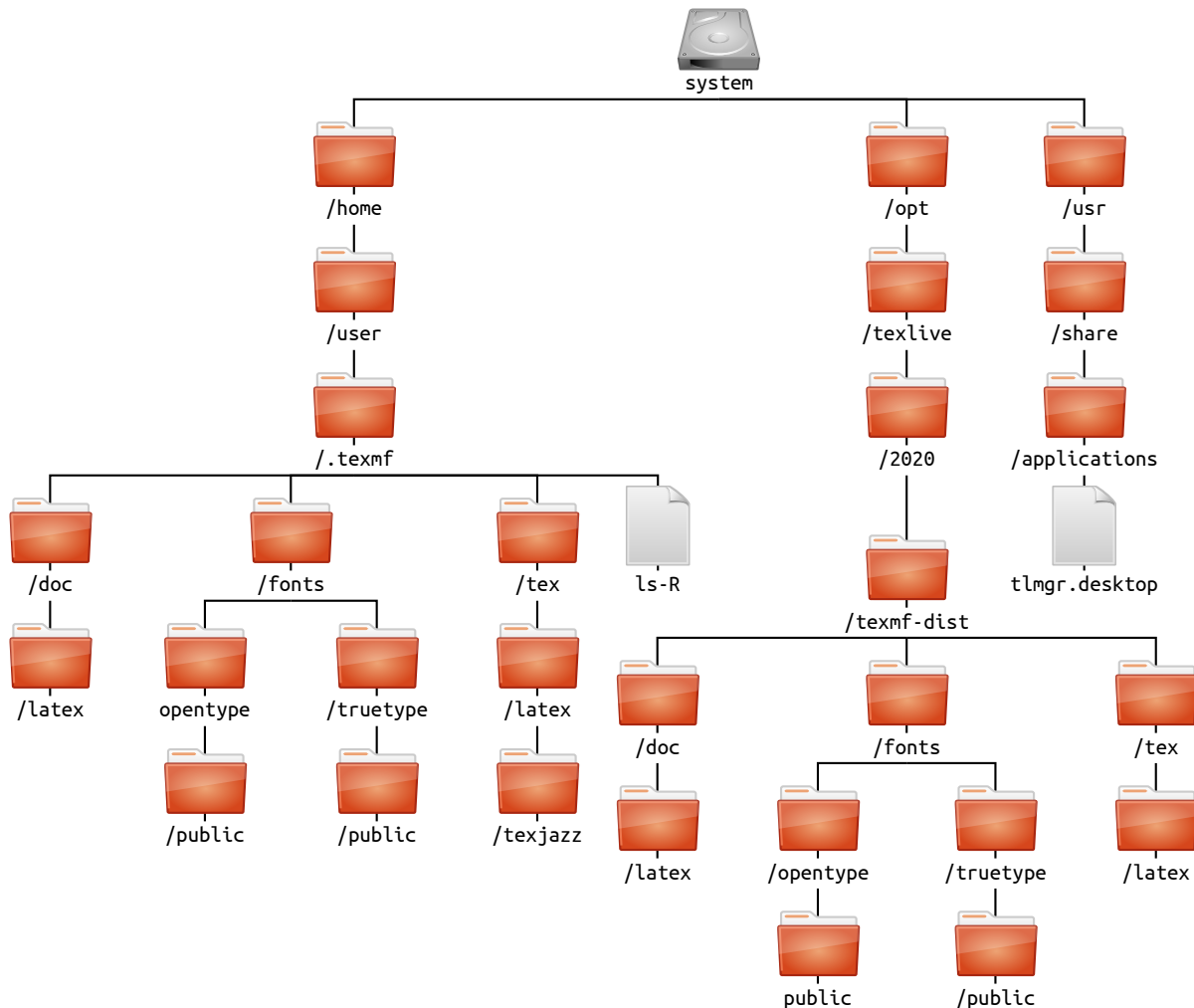
```

Enfin, on peut configurer son dépôt \LaTeX personnel situé sous `~/.texmf`. En respectant la TDS — *T_EX Directory Structure* —, il faut créer une suite de répertoires singeant la même arborescence¹¹ que la distribution $T_{\text{E}}X$ Live elle-même (voir arborescence personnelle et du système en page 3, de même qu'en page suivante pour mémoire). Cela permet de retrouver les fichiers rapidement, tant pour le système que pour l'utilisateur averti.

- `~/.texmf/doc/<package-name>;`
- `~/.texmf/fonts/truetype/public/<font-name>;`
- `~/.texmf/fonts/opentype/public/<font-name>;`
- `~/.texmf/tex/latex/<package-name>;`

11. Voir sous `opt/texlive/2020/texmf-dist`.

Concrètement, renvoi est fait à la section § 1.1 pour avoir le détail des fichiers de la classe **handbook** à installer. Ce faisant, la distribution T_EX LIVE est désormais opérationnelle en ligne de commande.



Une dernière étape importante est relative à l'installation d'un IDE — *Integrated Development Environment* pour L^AT_EX, comme T_EXMAKER ou L^AT_EXILA/GNOME L^AT_EX. En effet, si l'on tente d'installer un programme de ce type ou tout autre ayant des dépendances avec T_EX LIVE, la distribution LINUX va vouloir installer *ses propres paquets* pour satisfaire *ses dépendances*. La solution consiste à proposer au système d'exploitation un paquet factice — en anglais *dummy* ou *fake package* — qui ne fait rien d'autre que répondre aux dépendances et contourne la T_EX LIVE des dépôts LINUX au profit de celle venant d'être installée.

La démarche pour réaliser cet objectif provient d'une adaptation actualisée de l'explication donnée dans les liens qui suivent : <https://tex.stackexchange.com/>

questions/1092/how-to-install-vanilla-texlive-on-debian-or-ubuntu et <http://www.tug.org/texlive/debian.html#vanilla>).

Il faut avant tout installer les paquets de la distribution LINUX qui sont nécessaires à la création d'un paquet factice avec **equivs**, créer un répertoire temporaire puis télécharger ou construire le fichier de renseignements (cf. infra) et de constitution du paquet T_EX Live local. S'il est téléchargé, on peut l'éditer pour supprimer la dépendance avec l'extension **freelut3**, qui n'intervient pas dans le processus (implémentation d'un « OpenGL » libre).

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ sudo apt install equivs
user@host:$ mkdir -p ~/tmp/tl-equivs && cd ~/tmp/tl-equivs
user@host:~/tmp/tl-equivs$ wget -O texlive-local http://www.tug.org/texlive/files/debian-equivs-2020-ex.txt
```

</>

CODE 8 — Fichier de construction du paquet **texlive-local.deb**

```
Section: misc
Priority: optional
Standards-Version: 4.5.0

Package: texlive-local
Version: 2020-jazz
Maintainer: ejazz <ejazz.fr@gmail.com>
Provides: chktex, biblatex, biblatex-dw, cm-super, cm-super-minimal, context,
dvidvi, dvipng, feynmf, fragmaster, jadetex, lacheck, latex-beamer,
latex-cjk-all, latex-cjk-chinese, latex-cjk-chinese-arphic-bkai00mp,
latex-cjk-chinese-arphic-bsmi00lp, latex-cjk-chinese-arphic-gbsn00lp,
latex-cjk-chinese-arphic-gkai00mp, latex-cjk-common, latex-cjk-japanese,
latex-cjk-japanese-wadalab, latex-cjk-korean, latex-cjk-thai, latexdiff,
latexmk, latex-sanskrit, latex-xcolor, lcdf-typetools, lmodern, luatex,
musixtex, passivetex, pgf, preview-latex-style, prosper, ps2eps, psutils,
purifyeps, t1utils, tex4ht, tex4ht-common, tex-gyre, texlive, texlive-base,
texlive-bibtex-extra, texlive-binaries, texlive-common, texlive-extra-utils,
texlive-fonts-extra, texlive-fonts-extra-doc, texlive-fonts-recommended,
texlive-fonts-recommended-doc, texlive-font-utils, texlive-formats-extra,
texlive-games, texlive-generic-extra, texlive-generic-recommended,
texlive-humanities, texlive-humanities-doc, texlive-lang-african,
texlive-lang-all, texlive-lang-arabic, texlive-lang-cjk, texlive-lang-cyrillic,
texlive-lang-czechslovak, texlive-lang-english, texlive-lang-european,
texlive-lang-japanese, texlive-lang-chinese, texlive-lang-korean,
texlive-lang-french, texlive-lang-german, texlive-lang-greek,
texlive-lang-indic, texlive-lang-italian, texlive-lang-other,
texlive-lang-polish, texlive-lang-portuguese, texlive-lang-spanish,
texlive-latex-base, texlive-latex-base-doc, texlive-latex-extra,
texlive-latex-extra-doc, texlive-latex-recommended,
texlive-latex-recommended-doc, texlive-luatex, texlive-math-extra,
texlive-metapost, texlive-metapost-doc, texlive-music,
texlive-omega, texlive-pictures, texlive-pictures-doc, texlive-plain-extra,
texlive-plain-generic,
texlive-pstricks, texlive-pstricks-doc, texlive-publishers,
```

```
texlive-publishers-doc, texlive-science, texlive-science-doc, texlive-xetex,
thailatex, tipa, tipa-doc, xindy, xindy-rules, xmltex, asymptote, texinfo
```

Architecture: all

Description: Local fake installation of TeX Live 2020.

A full "vanilla" TeX Live 2020 customized for French/English languages.

Dummy/fake package to allow others packages installation which depend
of TeX-related Debian/Ubuntu packages, like TeXmaker or LaTeXila for example
<http://tug.org/texlive/debian#vanilla>

On peut alors construire et installer le paquet **texlive-local**, puis, par exemple installer l'IDE que l'on souhaite : T_EXmaker, T_EXstudio, T_EXworks, etc.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:/tmp/tl-equivs$ equivs-build texlive-local
user@host:/tmp/tl-equivs$ sudo dpkg -i texlive-local_2020-1_all.deb
```

12. Il est fortement déconseillé de lancer une mise à jour au moment où une tâche importante est à faire à courte échéance.

La mise à jour de T_EX Live peut se faire régulièrement au moyen de l'outil **tlmgr** pour *T_EX Live Manager* qui, lui-même se remet également à jour périodiquement. Cette action peut aussi bien se réaliser¹² en ligne de commande qu'à l'aide d'une interface graphique en T_CL-T_K.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ sudo tlmgr update --self
user@host:$ sudo tlmgr update --all
user@host:$ sudo tlmgr --gui
```

Enfin, on peut souhaiter disposer dans le menu des applications d'un lanceur de l'interface graphique pour **tlmgr**. Pour ce faire, il faut créer un nouveau fichier **tlmgr.desktop** donné ci-dessous puis placer ce dernier dans le répertoire **usr ▶ share ▶ applications** pour que tous les utilisateurs ayant les droits d'administration puissent y accéder.

```
Terminal
Fichier Éditer Affichage Rechercher Terminal Aide
user@host:$ cat > tlmgr.desktop <<EOF
[Desktop Entry]
Version=2020-jazz
Name=TeX Live Manager
Name[fr]=Outil d'administration TeX Live
Comment=Manage TeX Live packages
Comment[fr]=Administration des paquets TeX Live
GenericName=Package Manager
GenericName[fr]=Outil d'administration de paquets
Exec=pkexec env DISPLAY=$DISPLAY XAUTHORITY=$XAUTHORITY /opt/texlive/2020/bin/x86_64-linux/tlmgr --gui
Icon=system-software-update
Terminal=false
Type=Application
Categories=Office
```

```
Keywords=TeXLive;package;install;update
EOF
user@host:$ sudo cp tlmgr.desktop /usr/share/applications
```

5 Copyright et licence

Copyright © 2020-2021 ejazz.

This work is part of the T_EXJazz bundle and may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is available at <http://www.latex-project.org/lppl.txt> and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is ejazz.

This work consists of the files :

‘**texjazz-handbook.cls**’ and ‘**texjazz-handbook[-fr].tex**’

6 Historique des versions

2021/02/12 — **vo.1h**

Splitting of the main class into multiple more comprehensible files.
French documentation improving. Still experimental.

2021/01/14 — **vo.1g**

New French documentation, new version according to the **askreply**
and **assignpoints** packages version. Still experimental.

2021/01/12 — **vo.1b**

Changing the **sidenotes** package using to an adapted version of the
snotez package : **texjazz-sidenotes**.

2020/03/20 — **vo.1a**

First official release, but still very experimental.

7 Commentaires circonstanciés (pour mémoire)

8 Implémentation

Index

A

`\addbibresource`, 5
`amsmath` (paquet), 8
`\appendixchapter`, 7
`\appendixpage`, 9
`array` (paquet), 28
`article` (paquet), 10
`askreply` (paquet), 28, 35
`assignpoints` (paquet), 28, 35
`\asymmetricalpage`, 9
`\author`, 28
`\authorref`, 28

B

`\backchapter`, 7
`biblatex` (paquet), 5
`\blockquotation`, 16
`book` (paquet), 10, 19
`booktab` (paquet), 28
`\boxtitlefont`, 27

C

`caption` (paquet), 9
`\caution`, 14, 15
`\cautionleft`, 14
`\cautionright`, 14
`changepage` (paquet), 9
`\chapter`, 7
`citequote` (environnement), 16
`cleveref` (paquet), 9, 19
`code` (environnement), 18, 19
`code*` (environnement), 18, 19
`codebox` (environnement), 18, 19
`codebox*` (environnement), 19
`colortbl` (paquet), 28
`\copyrightname`, 28
`\copyrightsymbol`, 28
`\cref`, 19

D

`\date`, 28
`\docaddlogo`, 28
`\doclogo`, 28
`\documentclass`, 4

`\docversion`, 28

E

`english` (option), 5

F

`\faExclamationTriangle`, 14
`\faEye`, 14
`\faInfoCircle`, 14
`\faQuestion`, 14
`\faTag`, 14
`figure` (environnement), 10
`fontspec` (paquet), 8
`\footnote`, 8
`\footnotesize`, 10
`frame color` (option), 14
`french` (paquet), 8
`fullwidth` (environnement), 9

G

`geometry` (paquet), 9
`glossaries` (paquet), 5
`glossaries-extra` (paquet), 5
`\glossaryheaderfont`, 27
`gofurther` (environnement), 15
`gofurther*` (environnement), 15
`graphicx` (paquet), 8

H

`hanbook` (paquet), 17
`handbook` (paquet), 110, 13, 1517, 19, 22, 28, 30, 32
`\href`, 12, 17
`\hrefdoc`, 16
`hyperref` (paquet), 9, 12, 17

I

`idleconsole` (environnement), 27
`idleconsole*` (environnement), 27
`idleshell` (environnement), 26, 27
`idleshell*` (environnement), 26
`ifoddpage` (paquet), 9
`\ipythonpromptdots`, 24
`\ipythonpromptin`, 24
`\ipythonpromptout`, 24

`ipythonshell` (environnement), 24
`ipythonshell*` (environnement), 24
`\ipythontext`, 24
`\ipythonuserprompt`, 24

J

`jazzcode` (environnement), 11
`jazzcode*` (environnement), 11
`jazzfigure` (environnement), 10, 11
`jazzfigure*` (environnement), 10
`jazzgraphic` (environnement), 11
`jazzgraphic*` (environnement), 11
`jazzlisting` (environnement), 11
`jazzlisting*` (environnement), 11
`jazztable` (environnement), 10
`jazztable*` (environnement), 10

L

`\label`, 19
`\launchvideo`, 12
`lettrine` (paquet), 8
`\lightbf`, 27
`\lightboldfont`, 27
`\lighthbf`, 28
`linewidthnote` (environnement), 15
`listing` (environnement), 17, 18, 20
`listingbox` (environnement), 17, 19
`listingbox*` (environnement), 17
`listings` (paquet), 17, 20, 22
`longtable` (paquet), 28
`lstlisting` (environnement), 20

M

`makecell` (paquet), 28
`\makeglossaries`, 6
`\margincontents`, 10
`\marginelement`, 10
`marginfigure` (environnement), 13
`marginfix` (paquet), 9
`margingraphic` (environnement), 13
`marginnote` (paquet), 9, 10, 12, 14
`\marginnote.`, 10
`\marginpar`, 9, 10, 12
`marginable` (environnement), 13
`marginvideo` (environnement), 12, 13
`marker` (environnement), 15
`minted` (paquet), 46, 17, 18, 20, 22

`modelling` (option), 5
`\movie`, 13
`multicol` (paquet), 28
`multimedia` (paquet), 13
`multirow` (paquet), 28

N

`nbjupyterin` (environnement), 25
`nbjupyterout` (environnement), 25
`\normalsize`, 10

O

`\onesidegraphic`, 12
`\onesideimage`, 12
`optional title` (option), 14
`\overparagraph`, 7

P

`\paragraph`, 7
`\parnote`, 8
`parnotes` (paquet), 8
`\pdfdoc`, 16
`\pdflink`, 17
`\pdfsavepos`, 10
`\pdfwatch`, 17
`pgfplots` (paquet), 8
`piechart` (paquet), 28
`position` (option), 14
`printfiles` (option), 5
`pyconsole` (environnement), 26, 27
`pylab` (environnement), 26
`pythontex` (paquet), 5, 6, 17

R

`\ref`, 19
`\referencepage`, 9
`remark` (environnement), 14
`\reversemarginpar`, 12

S

`\savepos`, 10
`\section`, 7
`\setuser`, 22, 23
`shcolor` (option), 23
`shdirectory` (option), 23
`shhost` (option), 23
`showframe` (option), 5

shprompt char (option), 23
shuser (option), 23
\sidefigure, 12
\sidegraphic, 12
\sideimage, 12
\sidenote, 8
sidenotes (paquet), 35
\sidequote, 16
\sideremark, 14
\sidetable, 12
\sidevideo, 12
snotex (paquet), 35
\spacedlowsmallcaps, 27
\spacedsmallcaps, 27
\startconsole, 23
subcaption (paquet), 9
\subsection, 7
\subsubsection, 7
\subtitle, 28
\subtitlefont, 27
\symmetricalpage, 9

T

table (environnement), 10
tabu (paquet), 28
tabularx (paquet), 28
tcolorbox (paquet), 8, 10, 14, 15, 17, 20
texjazz-askreply (paquet), 3
texjazz-assignpoints (paquet), 3
texjazz-handbook (paquet), 3

texjazz-piechart (paquet), 3
texjazz-sidenotes (paquet), 35
text (argument), 14
\textdoc, 17
\textttl, 28
threeparttable (paquet), 28
tikz (paquet), 8
\title, 28
\titlefont, 27
titles (paquet), 28
titlesec (paquet), 9, 28
titletoc (paquet), 28
\titlingfont, 27
\titlingspacedfont, 27
\tocfont, 27

U

ubuntu (environnement), 22
ubuntu* (environnement), 22
unicode-math (paquet), 8
user (option), 23

V

\versiondate, 28

W

\webdoc, 17
workbook (option), 4

X

xcolor (paquet), 8
xfontawesome (paquet), 3