



FORMATION

Matière

Cours

CHOIR BLASON
fichiers_multimedia

Prénom NOM

CHOIR BLASON
fichiers_multimedia

Prénom NOM

Édition 2021.4



Association ouvrière des Compagnons du Devoir et du Tour de France
constituée selon la loi de 1901, reconnue d'utilité publique
82, rue de l'Hôtel-de-Ville - 75180 Paris Cedex 04 - Téléphone : 01 44 78 22 50
www.compagnons-du-devoir.com

Table des matières

Préface	vii
1 Principes de base de la rédaction à l'aide de L^AT_EX	1
1.1 Valeurs ajoutées de L ^A T _E X	1
1.2 Paramètre du code <code>master.tex</code>	3
1.3 Division du document	3
1.4 Agencement du document	3
1.5 Compilation du document	5
2 Rédaction de texte	9
2.1 Structuration du texte	9
2.2 Mise en forme	12
3 Référencement et bibliographie	17
3.1 Introduction	17
3.2 Renvois	17
3.3 Index et glossaire	20
3.4 Environnements référencés	23
3.5 Bibliographie	25
Annexes	27
Bibliographie	29
Index	31
Index des termes du thème exemple	33
Glossaire	35
Liste des acronymes	37



Liste des tableaux



Liste des figures

3.1	Image pour le premier exemple d'intralien	18
3.2	Image pour le deuxième exemple d'intralien	19
3.3	Image pour le troisième exemple d'intralien	19
3.4	Image pour le quatrième exemple d'intralien	20



Liste des formules

3.1 Exemple d'environnement formule	25
---	----



Liste des définitions

3.1	Exemple d'environnement definition	24
-----	--	----



Liste des exemples

1.1	Notation des siècles	1
1.2	Style du chapitre	2
1.3	Minipage	4
1.4	Type de média	6
2.1	Texte en trois colonnes	10
2.2	Liste et énumération	11
2.3	Citation	15
3.1	Intralien	18
3.2	Hyperlien	20
3.3	Index	21
3.4	Glossaire	22
3.5	Environnement exemple	23
3.6	Exemple d'environnement exemple	23
3.7	Environnement définition	24
3.8	Environnement formule	24



Préface

Ce *template* contient un recueil neutre des programmations *types* que les programmeurs seront amenés à coder pour aboutir à des documents unifiés graphiquement et à la présentation irréprochable.

Celui-ci se veut exhaustif quant aux nombreux cas de figures auxquels les programmeurs feront face, comme l'insertion de figures avec toutes options que cela comporte, ou encore le dessin sous \LaTeX ... La liste est longue et je renvoie vers la table des matières pour avoir une vue d'ensemble face aux différentes programmations type. La liste des exemples donne également un aperçu rapide des différents exemples de codes. Il s'agit là d'un outil permettant de faciliter grandement la programmation de nouveaux documents, à conserver en toile de fond lors de la prise en main de \LaTeX ...

L'outil \LaTeX étant bien conçu (et bien amélioré par le package AOCDTF), toutes les références listées ainsi que toutes les divisions ou encore l'entièreté de la table des matières sont référencées – bien que ça ne soit mis en évidence par des couleurs pour éviter un arc-en-ciel illisible – et il suffit de cliquer sur ces éléments pour être renvoyé directement sur sa localisation dans le document. Il s'agit là d'une des nombreuses valeurs ajoutées de \LaTeX .



1 Principes de base de la rédaction à l'aide de L^AT_EX

1.1 Valeurs ajoutées de L^AT_EX

L^AT_EX se charge de structurer les textes rédigés et des éléments dits *flottants* (image, dessins...), qu'on désignera comme étant le *fond* du document. Tous les processus permettant de modifier le style du texte (gras, italique, police...), sa disposition (doubles colonnes, alignement à droite...) ou encore les informations récurrentes sont exécutés sous forme d'instructions incluses dans le langage de programmation T_EX.

La première valeur ajoutée de L^AT_EX est sa capacité à gérer la *forme* du document selon les normes en vigueur et sa classe, et de pouvoir le laisser gérer cet aspect-là, souvent chronophage, durant la rédaction.

Une autre valeur ajoutée de L^AT_EX est d'automatiser certains processus de rédaction par la création de nouvelles instructions (ou l'utilisation d'instructions existantes). Le package AOCDTF comporte ainsi une multitude de *macro-commandes* aux buts aussi variés qu'utiles, je vous conseille d'aller consulter le [code](#) du package AOCDTF pour les décrypter. Toutes les nouvelles macro-commandes (ainsi que les instructions existantes nécessaires à la bonne rédaction des documents) seront détaillées dans ce recueil.

Sachant que chaque caractéristique de ce qui constitue le *fond* du document est *a priori* encadré par une instruction précise, chacune de celle-ci peut être modifiée sur tous les documents rédigés en une seule mise à jour. Il s'agit dès lors d'encadrer un maximum de caractéristiques du document qui peuvent l'être par des instructions. Et le faire avant la production de cours, afin d'éviter de devoir rajouter de nouvelles instructions après coup, mais aussi dans un but d'uniformisation des documents produits.

Cela peut être illustré par deux exemples (un *exemple* étant déjà caractérisé par un lot d'instructions, cela est détaillé dans l'[exemple 3.5](#)) :

Exemple 1.1: Notation des siècles

Cela sera abordé dans la [sous-sous-section 2.2.5.1](#), mais afin d'automatiser la rédaction de cas particuliers de la langue française comme la notation des siècles (et afin que chacun n'en fasse pas qu'à sa manière), le package AOCDTF contient la nouvelle macro-commande définie comme telle :

```
\newcommand*{\siegcle}[1]{
\ifnum#1=1
\bsc{\romannumeral #1}\textsuperscript{er}~siècle
\else
\bsc{\romannumeral #1}\textsuperscript{e}~siècle
\fi}
```



Lorsqu'on appelle l'instruction `\sieclet{1}`, cela produira 1^{er} siècle. Si l'on appelle l'instruction `\sieclet{2}`, cela produira 2^e siècle. Je vous laisse décrypter la définition de cette instruction ci-dessus pour essayer de comprendre son fonctionnement.

Code

Lorsqu'on appelle l'instruction `\Verb{\sieclet{1}}`, cela produira `\sieclet{1}`. Si l'on appelle l'instruction `\Verb{\sieclet{2}}`, cela produira `\sieclet{2}`. Je vous laisse décrypter la définition de cette instruction ci-dessus pour essayer de comprendre son fonctionnement.

Exemple 1.2: Style du chapitre

La définition du style d'un chapitre a demandé une quinzaine de jour pour le personnaliser. Son bloc de code se situe en fin du [package AOCDF](#), sous le titre `%Mise en page du document` et le sous-titre `%Chapitre`, les plus téméraires peuvent donc essayer de le décrypter.

Résultat, l'instruction redéfinie `\chapter{Visualition du style d'un chapitre}` donnera donc :

CHAPITRE

1 Visualition du style d'un chapitre

Ce style du titre d'un chapitre pourra être modifié ultérieurement d'un simple mise à jour pour *tous* les documents rédigés avec le package AOCDF car tout ce qui caractérise ce style est défini dans ce package qui constitue donc le tronc commun de tous ces documents.

Dans les principes de base, une dernière valeur ajoutée de \LaTeX – il y en aura d'autres sur une multitude de sujets – se concrétise dans la *relativité* et l'*exactitude* des unités de mesure et de la disposition des éléments sur la page. Les unités sont multiples :

millimètre : mm ;

centimètre : cm ;

point anglo-saxon pt ;

point Didot dd ;

hauteur de la lettre x : ex ;

cadratin (largeur de la lettre M) : em.

Les unités de mesures énumérées ci-dessus sont soit absolues, soit relatives. Cela permet aux éléments définis par ces mesures d'être dimensionnés très précisément et de se restructurer en cas de changement de police ou de taille d'écriture pour conserver une même échelle.

L'*exactitude* de ces mesures permet au rédacteur de disposer d'une grande précision dans l'agencement des pages ou encore lors du codage d'un schéma en instructions \LaTeX (cela sera abordé dans le ??).

L'autre atout est de pouvoir indiquer des valeurs *relatives* aux espaces de rédaction paramétrées en préambules ou en rédaction. Cela est explicité dans l'[exemple 1.3](#).

Il convient toutefois de prêter attention à l'écriture des nombres, qui se fait au format américain. Le séparateur décimal sera donc . et non , , sous peine d'erreur de compilation.



1.2 Paramètre du code master.tex

Tous les paramètres nécessaires pour bien démarrer la rédaction d'un document sont détaillés à cette [page](#) du wiki.

1.3 Division du document

Avant de rédiger du texte, il convient de structurer le document en plusieurs divisions. Chaque style de chaque titre de division est défini dans le package AOCDTF, et pourra donc être modifié – après consultation – d'une seule intervention de mise à jour du code.

Selon les instructions suivantes, elles seront automatiquement numérotées dans l'ordre ou elles sont appelées dans le code :

1. partie : `\part{<Titre de la partie>}` (à appeler dans le code `master.tex`) ;
2. chapitre : `\chapter{<Titre du chapitre>}` (à rédiger à partir d'un copier/coller du code `MWE.tex`) ;
3. section : `\section{<Titre de la section>}` ;
4. sous-section : `\chapter{<Titre de la sous-section>}` ;
5. sous-sous-section : `\chapter{<Titre de la sous-sous-section>}` ;
6. paragraphe : `\paragraph{<Titre du paragraphe>}` .

Les mêmes instructions auxquelles sont rajoutées une astérisque produiront le même résultat à la différence que ces divisions-là ne seront pas numérotées ni référencées dans la table des matières :

1. partie : `\part*{<Titre de la partie non numérotée>}` (à appeler dans `master.tex`) ;
2. chapitre : `\chapter*{<Titre du chapitre non numéroté>}` ;
3. section : `\chapter*{<Titre de la section non numérotée>}` ;
4. sous-section : `\chapter*{<Titre de la sous-section non numérotée>}` ;
5. sous-sous-section : `\chapter*{<Titre de la sous-sous-section non numérotée>}` ;
6. paragraphe : `\paragraph*{<Titre du paragraphe>}` .

Il convient également d'indiquer si le chapitre que l'on rédige présente des signets et une pagination à la couleur variable selon le chapitre avec la macro-commande : `\ChapFrame` .

1.4 Agencement du document

1.4.1 Saut de page

Ala compilation du code, il se peut que les flottants et autres éléments occupant l'espace peuvent contraindre le contenu textuel à se répartir de façon peu harmonieuse. Pour y remédier – à la toute fin de la rédaction – il existe deux instructions permettant de « jouer » avec cette répartition en forçant les sauts de pages :

- saut de page sans répartition du contenu sur la page précédente : `\newpage` ;
- saut de page avec répartition du contenu sur la page précédente : `\pagebreak` .

1.4.2 En-tête, pied de page

Par souci de légèreté visuelle, les documents à destination de l'[Association Ouvrière des Compagnons du Devoir et du Tour de France \(AOCDTF\)](#) ne comportent pas d'en-tête et le pied de page comprend le logo sans texte de l'[AOCDTF](#) ainsi que le numéro de page. Celui-ci voit son format changer selon l'emplacement dans le document (frontmatter, mainmatter et backmatter) et si l'on souhaite indiquer le marqueur de chapitre, la pagination sera incluse dans une boîte de la même couleur que celle du marqueur de chapitre (seulement dans le mainmatter).



1.4.3 Minipage

Une page peut contenir des « pages » additionnelles à l'intérieur de la page initiale nommées « minipage », qui sont des espaces qui se comporteront comme une « page dans la page ». Les « minipages » peuvent se montrer utiles pour aligner deux éléments sur un même axe horizontal, comme une figure avec une liste descriptive ou encore deux tableaux. . .

Exemple 1.3: Minipage

Les instructions suivantes produiront deux « minipages », qui s'aligneront selon leurs paramètres d'alignement. Celle de gauche fera 8cm et va aligner le bas de son environnement – paramètre [b] – avec le haut de l'environnement – paramètre [t] – de celle de droite, qui fera 30% de la largeur de l'espace de rédaction. L'instruction `\hfill` remplira l'espace horizontal disponible entre les deux « minipages » :

Code

```
\begin{minipage}[b]{8cm}
\lipsum[66]
\end{minipage}
\hfill
\begin{minipage}[t]{0.30\linewidth}
\lipsum[75]
\end{minipage}
```

Cela produira :

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Pellentesque interdum sapien sed nulla. Proin tincidunt. Aliquam volutpat est vel massa. Sed dolor lacus, imperdiet non, ornare non, commodo eu, neque. Integer pretium semper justo. Proin risus. Nullam id quam. Nam neque. Duis vitae wisi ullamcorper diam congue ultricies. Quisque ligula. Mauris vehicula.

Cet environnement se paramètre en alignement selon les variables facultatives suivantes :

haut de la page (top) : t ;

milieu de la page (middle) : m ;

bas de la page (bottom) : b ;

à l'emplacement défini par le code (here) : h.

Ces variables sont facultatives pour une minipage, qui s'intégrera par défaut dans le texte à l'emplacement du texte ou l'environnement sera appelé dans le code correspondant.

Il se paramètre également sur la largeur de la minipage :



largeur relative à la largeur du texte : `<x\linewidth>` ;
largeur relative à la largeur de rédaction : `<x\textwidth>` ;
largeur absolue : `<xcm>` (ou autre unité de mesure propre à \LaTeX).

1.4.4 Espace de rédaction

Dans le package AOCDTF sont paramétrés les différents espaces de rédaction. Les marges horizontales et verticales sont toutes de 20mm à l'exception de celle qui n'est pas du côté de la reliure, mesurant quant à elle 25mm. L'espace de rédaction mesure donc 165mm. Les marges verticales, selon les instructions entrées dans le code `master.tex` sur le type de média peuvent soit :

- être décalées entre les pages paires et impaires si le paramétrage est prévu pour un document format papier, afin de pouvoir imprimer et relier correctement le document rédigé ;
- ne pas être décalées entre les pages paires et impaires si le paramétrage est prévu pour un document format écran.

1.5 Compilation du document

La compilation du code `master.tex` ou des sous-programmations peuvent nécessiter plusieurs essais pour que les différents éléments se positionnent correctement sur leur emplacement spécifié, ou encore pour que les référencements (table des matières, bibliographie...). Pour la compilation finale, il est nécessaire d'effectuer une « compilation rapide » plutôt que de compiler à l'aide du moteur PDFLaTeX. Sur le logiciel Texmaker, les instructions pour paramétrer cette « compilation rapide » sont détaillées sur cette [page](#) du wiki.

1.5.1 Sous-programmation

Lorsqu'on rédige un document conséquent, comme abordé sur cette [page](#) du wiki, on fait appel à une architecture de document de type *master/slave*. Cette architecture s'articule autour de la *subdivision* du code et l'insertion de *sous-programmations* au format `.tex` dans des programmations parentes. Pour alléger le code `master.tex`, les chapitres sont donc des sous-programmations que l'on insère dans le code `master.tex` à l'aide de l'instruction `\include{chap_mot-clé_mot-clé_plus_précis}` avec pour but de l'aérer.

Les sous-programmations peuvent également être des tableaux ou des figures au nombre de lignes de code trop important pour être insérées directement dans le corps de texte. Elles sont injectées dans la programmation parente – généralement un chapitre – avec l'instruction `\input{objet_mot-clé_mot-clé_plus_précis}`, toujours dans le but d'aérer les différents codes et de faciliter la navigation entre ceux-ci durant la programmation.

Comme explicité à cette [page](#) du wiki, ces sous-programmations doivent donc présenter des noms de fichiers de codes qui suivent une nomenclature bien précise, avec leur *objet* à préciser selon la liste suivante :

`chap_mot-clé-général_mot-clé-précis(__mot-clé-plus-précis)` : nom de code pour un chapitre ;

`ann_mot-clé-général_mot-clé-précis(__mot-clé-plus-précis)` : nom de code pour une annexe ;

`sec_mot-clé-général_mot-clé-précis(__mot-clé-plus-précis)` : nom de code pour une section (rare) ;

`tab_mot-clé-général_mot-clé-précis(__mot-clé-plus-précis)` : nom de code pour un tableau ;

`fig_mot-clé-général_mot-clé-précis(__mot-clé-plus-précis)` : nom de code pour une figure ;



eq_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) : nom de code pour une équation ;

graph_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) : nom de code pour un graphique.

Pour rédiger une sous-programmation, il suffit de suivre les instructions à cette [page](#) du wiki. Il convient donc d'initier une nouvelle sous-programmation en copiant le code `MWE.tex`, commande Texmaker que l'on retrouve dans **Fichier > Nouveau en copiant à partir d'un document existant**. L'optimisation du code contenu dans le package AOCDFTF automatise complètement la compilation du code d'une sous-programmation, qu'elle soit compilée seule ou incluse dans une programmation parente.

En cas de rédaction d'un nouveau chapitre, il faut juste ne pas oublier d'indiquer l'instruction `\ChapFrame` juste après l'instruction `\chapter{Titre du chapitre}` si l'on souhaite voir apparaître les marqueurs sur ce chapitre.

1.5.2 Usage des documents

Comme abordé dans cette [page](#) du wiki, il est possible de conditionner la compilation du document selon l'usage qu'il en sera fait avec l'instruction `\media{type_media}`. Le type de média sera donc **screen** ou **paper**. Ce paramétrage réarrange le document selon qu'il sera diffusé sur papier ou lu à l'écran (ré-agencement des pages, liens en noir...) mais pas que.

Durant toutes la rédaction du code, le rédacteur sera amené à inclure des éléments multimédias (vues 3D, vidéo...), il conviendra donc de doubler chaque élément *interactif* de sa version papier et inversement.

Cela est réalisé à l'aide de la macro-commande conditionnelle `\media{\thetypemedia}{code pour usage papier}{code pour un usage interactif}`. Les codes rédigés dans cette instruction s'exécuteront donc selon le paramètre **screen** ou **paper** renseigné dans l'instruction `\media{<type du média>}` en début du code `master.tex`, il s'agit d'un *ou* exclusif, les deux codes ne pourront jamais s'exécuter de concert.

Exemple 1.4: Type de média

L'exemple suivant permettra, selon les paramètres **screen** ou **paper** renseignés à la compilation de ce document, d'exécuter l'un ou l'autre code :

Code destiné à un usage interactif comme par exemple une vidéo qui s'afficherait dans le document si l'instruction `\media{screen}` était appelée dans le code `master.tex`.

Code

```
\media{\thetypemedia}
{Code destiné à un usage papier comme
par exemple une capture d'écran tirée
d'une vidéo qui s'afficherait dans le
document si l'instruction
\media{screen} était appelée dans le
code \texttt{master.tex} .}
{Code destiné à un usage interactif
comme par exemple une vidéo qui
s'afficherait dans le document si
l'instruction \media{screen} était
appelée dans le code
\texttt{master.tex} .}
```



2.1 Structuration du texte

2.1.1 Paragraphes et saut de ligne

Lorsqu'on rédige des éléments textuels avec l'outil \LaTeX , il faut prendre en compte quelques subtilités d'agencement du texte :

- \LaTeX se charge de structurer le texte et passera à la ligne selon les normes typographiques françaises (division des mots en fin de ligne, espacement...) ;
- pour sauter à la ligne dans le même paragraphe, il faut appeler l'instruction `\` (avec *un* ou sans saut de ligne) ;
- pour aborder un nouveau paragraphe, il faut appeler l'instruction `\` suivi de *deux* sauts de ligne ;
- sauter une ligne sans l'instruction `\` ne produira pas de saut de ligne à la compilation ;
- la structuration du texte suit la réglementation française :
 - un saut de ligne n'induit pas *d'indentation* (petit espacement avant le premier d'un bloc de phrase dans le même paragraphe) ;
 - un changement de paragraphe produit un *interligne* suivi d'un *alinéa* (indentation suite à un changement de paragraphe).

On remarque donc que l'instruction `\`, qu'elle ne soit pas suivie ou qu'elle soit suivie d'un seul saut de ligne ne changera pas le résultat. Toutefois, pour la lisibilité du code, il est conseillé de suivre cette instruction d'un saut de ligne.

2.1.2 Ponctuation, symboles spéciaux et espaces

Lorsqu'on programme du texte avec \LaTeX , il existe une série de caractères réservés qui ne peuvent pas figurer dans le document final car ils ont une incidence dans la compilation du code. Il s'agit des caractères suivants :

`{ } % # $ ^ ~ & _ \`

S'il faut écrire ces caractères dans le texte final, il conviendra de rajouter `\` devant chacun de ces caractères et `\textbackslash` pour la contre-oblique.

\LaTeX gère les espaces entre les mots selon la réglementation française, il convient dès lors d'uniformiser la rédaction selon ses subtilités concernant les espaces, les signes de ponctuations utilisés... Toutes ces subtilités à respecter sont détaillées dans cet [aide-mémoire](#).

Au registre des subtilités récurrentes :

- différenciation les traits d'union - (mot composé) des tirets demi-cadratin `--` (incises et intervalles) et des tirets cadratin `---` (dialogue et listes) ;
- accentuation s'effectue même sur les lettres majuscules à l'aide des instructions `\`E`, `\`E` ou encore `\`A` (ou juste `Ê` et `À` selon les claviers) ;
- utilisation des guillemets français `\og mot entre guillemets \fg{}`
- utiliser les instructions pour les symboles spéciaux et abréviations que `\oe{}`, `\ier{}`... la liste est longue et peut être retrouvée sur cette [page](#) ainsi qu'à cette [sous-sous-section 2.2.5.1](#) ;



- utilisation des points de suspensions comme ponctuation de fin de phrase – jamais après une virgule – avec l’instruction `\ldots` ;
- utiliser le symbole de l’euro € avec l’instruction `\EUR{}` .

Les espaces sont finement gérés par L^AT_EX et a priori, il n’est pas nécessaire de les appeler durant la programmation. Néanmoins voici la liste des différents type d’espaces :

espace justifiant : espace obtenue avec la barre d’espace. Répéter cet espace ne produira qu’un seul espace justifié dans le document final et il peut ne pas apparaître après une instruction type « ouverte » (non clôturée par `}`) ;

espace insécable : justifie obligatoirement un saut de ligne avec l’instruction `~` ;

petit espace : utilisée surtout en fin de ligne dans les listes et descriptions avec l’instruction `\,` ;

espace fine : peut régler des problèmes de débordements avec l’instruction `\/` .

2.1.3 Disposition du texte

La disposition du texte avec L^AT_EX est gérée automatiquement. Par défaut, le texte occupe toute la largeur de l’espace de rédaction dont la mesure est appelée avec l’instruction `\linewidth` ou `\textwidth` . Ces deux instructions renvoient à la même valeur dans le cas d’un texte à une colonne. Dans le cas d’un texte à plusieurs colonnes, l’instruction `\linewidth` renvoie la valeur de la largeur d’une seule colonne tandis que `\textwidth` renvoie toujours à la largeur de l’espace de rédaction de la page.

Le texte, ainsi que les figures et autres éléments, se disposent selon les règles de typographies en vigueur selon une disposition « justifiée » (le saut de ligne est effectué au même emplacement). La disposition peut être modifiée dans quatre environnements différents :

justifié : `\begin{justify}` texte justifié sans indentation `\end{justify}` ;

aligné à gauche : `\begin{raggedright}` texte aligné à gauche `\end{raggedright}` ;

aligné à droite : `\begin{raggedleft}` texte aligné à gauche `\end{raggedleft}` ;

centré : `\begin{center}` texte centré `\end{center}` .

Le texte peut se répartir également sur plusieurs colonnes avec l’environnement `\begin{multicols}` , qui répartira son contenu en nombre de colonnes spécifié avec une répartition homogène du texte.

Exemple 2.1: Texte en trois colonnes

Les instructions suivantes disposeront le texte en trois colonnes selon le paramétrage choisi entre accolade. Pour forcer un passage à la colonne suivante, il faudra indiquer l’instruction `\columnbreak\\` sans oublier le retour à la ligne `\\` :

```
\begin{multicols}{3}
\lipsum[45]
\columnbreak\\
\lipsum[75]
\end{multicols}
```

Cela produira :



Cras egestas ipsum a nisl. Vivamus varius dolor ut dolor. Fusce vel enim. Pellentesque accumsan ligula et eros. Cras id lacus non tortor facilisis facilisis. Etiam nisl elit, cursus sed, fringilla in, congue nec, urna. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Integer at turpis. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis fringilla, ligula sed porta fringilla, ligula wisi commodo felis, ut adipiscing felis dui in enim. Suspendisse malesuada ultrices ante. Pellentesque scelerisque augue sit amet urna. Nulla volutpat aliquet tortor. Cras aliquam, tellus at aliquet pellentesque, justo sapien commodo leo, id rhoncus sapien quam at erat. Nulla commodo, wisi eget sollicitudin pretium, orci orci aliquam orci, ut cursus turpis justo et lacus. Nulla vel tortor. Quisque erat elit, viverra sit amet, sagittis eget, porta sit amet, lacus. Pellentesque interdum sapien sed nulla. Proin tincidunt. Aliquam volutpat est vel massa. Sed dolor lacus, imperdiet non, ornare non, commodo eu, neque. Integer pretium semper justo. Proin risus. Nullam id quam. Nam neque. Duis vitae wisi ullamcorper diam congue ultricies. Quisque ligula. Mauris vehicula.

2.1.4 Listes

L^AT_EX permet de créer facilement des listes qui présenteront toujours la même mise en page, encore une fois paramétrable à tous les documents d’un coup.

Exemple 2.2: Liste et énumération

Une liste se rédige dans un environnement `\begin{itemize}` , qui est explicité selon les instructions suivantes :

— un premier élément de la liste ;	
— un deuxième élément de la liste ;	
— un troisième élément de la liste.	
	<div>Code</div>
	<pre> \begin{itemize} \item un premier élément de la liste\,; \item un deuxième élément de la liste\,; \item un troisième élément de la liste. \end{itemize} </pre>

Les listes rédigées pour les documents destinés à l’**AOC****DTF** doivent respecter les normes de typographies françaises. Cela concerne deux consignes, d’une part l’instruction `\,;` en fin de ligne pour les éléments de la liste, à l’exception du dernier qui clôture la liste par un `.` .Et d’autre part, l’absence de majuscule pour le premier mot de chaque élément de *tous* les types de liste.

Les éléments peuvent également être énumérés avec l’environnement `\begin{enumerate}`, ainsi que comporte plusieurs niveaux de liste :



1. un premier élément de la liste ;
2. un deuxième élément de la liste ;
 - un premier élément du deuxième niveau de la liste ;
 - un deuxième élément du deuxième niveau de la liste.
3. un troisième élément de la liste.

Code

```
\begin{enumerate}
\item un premier élément de la liste\,;
\item un deuxième élément de la liste\,;
    \begin{itemize}
    \item un premier élément du
    deuxième niveau de la liste\,;
    \item un deuxième élément du
    deuxième niveau de la liste\,;
    \end{itemize}
\item un troisième élément de la liste.
\end{enumerate}
```

La liste peut également contenir des descriptions d'éléments avec l'environnement `\begin{description}` :

premier élément description du premier élément ;

deuxième élément description du deuxième élément ;

troisième élément description du troisième élément.

Code

```
\begin{description}
\item [premier élément] description du
premier élément\,;
\item [deuxième élément] description du
deuxième élément\,;
\item [troisième élément] description du
troisième élément.
\end{description}
```

Selon le contexte, le terme en gras peut être suivi d'un double point :, qui fera *toujours* partie intégrante du terme entre crochet [] .

Également, selon que l'on insère un *double* saut de ligne dans le code (sans l'instruction `\\+`) juste après une liste ou no, cela produit ou non un nouveau paragraphe, il conviendra d'adapter cette spécificité pour conserver une cohérence dans la rédaction.

D'autres formats de listes plus compactes ont été créés pour être intégrées aux tableaux, ils seront abordés dans ??.

2.2 Mise en forme

Le style du texte sous L^AT_EX est prédéfini selon sa fonction (texte, titre de section...) mais il existe des instructions pour définir la taille de la fonte, sa forme et sa graisse. La plupart de ces instructions se retrouvent en raccourci à gauche de l'**Environnement de Développement (Integrated Development Environment) (IDE)** de Texmaker.

La terminologie précise indique que la *police* de caractères désigne le dessin général des lettres tandis que la combinaison taille/forme/graisse d'une police sera désignée en tant que *fonte*. L'ensemble de toutes les fontes possible forme la police.

Les instructions suivantes permettent de paramétrer les *fontes* et sont combinables les unes avec les autres.

2.2.1 Choix de la police

Le choix de la police peut être modifié les instructions suivantes :

- police du document : `\textnormal{<caractères dans la police du document>}` ;
- romain : `\textrm{<caractères en police romaine>}` ;



- sans sérif/linéale : `\textsf{<caractères sans sérif>}` ;
- machine à écrire : `\texttt{<caractères de machine à écrire>}` .

En limitant le choix de police, on s'assure ainsi de conserver une unité graphique au sein de tous les documents produits pour l'AOCDF. Le changement de jeu de polices de caractères peut être effectué d'une seule mise à jour si besoin est.

2.2.2 Taille de la police

La taille de la police peut être modifiée avec un jeu d'instructions détaillées dans la liste ci-dessous, de la plus petite à la plus grande :

- `texte : \begin{tiny} texte \end{tiny}` ;
- `texte : \begin{scriptsize} texte \end{scriptsize}` ;
- `texte : \begin{footnotesize} texte \end{footnotesize}` ;
- `texte : \begin{small} texte \end{small}` ;
- `texte : \begin{small} texte \end{small}` ;
- `texte : \begin{normal} texte \end{normal}` ;
- `texte : \begin{large} texte \end{large}` ;
- `texte : \begin{Large} texte \end{Large}` ;
- `texte : \begin{LARGE} texte \end{LARGE}` ;
- `texte : \begin{huge} texte \end{huge}` ;
- `texte : \begin{Huge} texte \end{Huge}` ;
- `texte : \begin{HUGE} texte \end{HUGE}` .

En attribuant des tailles *relatives*, L^AT_EX garde la main-mise sur un choix défini de tailles et conserve ainsi l'unité graphique au sein de tous les documents produits pour l'AOCDF. Ces tailles peuvent être modifiées d'une seule mise à jour si besoin est.

2.2.3 Forme de la police

La forme de la police peut être modifiée selon les instructions suivantes :

- droit : `\textup{<caractères droits>}` ;
- italique : `\textit{<caractères en italique>}` ;
- oblique : `\textsl{<caractères obliques>}` ;
- PETITE CAPITALE : `\textsc{<caractères en petites capitales>}` ;
- NOM PROPRE : `\textmd{<caractères pour noms propres>}` ;
- *emphase* : `\emph{<caractères en emphase>}` ;
- souligné : `\underline{<caractères soulignés>}` ;
- en exposant : `<caractères en exposant>` ;
- en indice : `\textsubscript{<caractères en indice>}` .

Le texte en emphase aura généralement le même effet que le texte italique, l'instruction `\emph{}` mettra la police des caractères contenus dans l'instruction dans la forme opposée du texte les englobant. Ainsi, un texte en italique comprenant une portion de texte dans l'instruction `\emph{}` verra cette portion en forme droite. Si l'on change ce texte pour du droit, la forme de la portion de texte s'inversera pour de l'italique. On préférera donc cette instruction relative à l'instruction absolue `\textit{caractères en italique}` pour mettre en évidence un mot dans un texte.



2.2.4 Graisse de la police

La graisse de la police peut être modifiée selon les instructions suivantes :

- **gras** : `\textbf{<caractères gras>}`
- moyennement gras : `\textmd{<caractères moyennement gras>}`.

2.2.5 Utilisation conventionnelle

Les fontes permettent de différencier des éléments spécifiques dans le texte tel que des noms propres, des dates. ... Cet [aide-mémoire](#) – déjà référencé ci-dessus – indique clairement les différentes fontes à utiliser selon l’usage en français, la liste est longue et ne sera pas détaillée dans le présent texte.

2.2.5.1 Abréviations et macro-commandes

Pour uniformiser certaines notations normalisées, le package AOCDTF contient quelques nouvelles macro-commandes listées ci-dessous :

- | | |
|---|---|
| — an 2000 : <code>\annee{2000}</code> ; | — 1 ^{er} millénaire : <code>\millenaires{1}</code> ; |
| — an 50 (avant J.-C.) : <code>\annee{-50}</code> ; | — I - II ^e millénaires : <code>\millenaires{1}{2}</code> ; |
| — apr. J.-C. : <code>\apjc</code> ; | — M. : <code>\monsieur{}</code> ; |
| — av. J.-C. : <code>\avjc</code> ; | — n ^o : <code>\numero{}</code> ; |
| — D ^r : <code>\docteur{}</code> ; | — n ^{os} : <code>\numeros{}</code> ; |
| — D ^{rs} : <code>\docteurs{}</code> ; | — p. ex. : <code>\parexemple{}</code> ; |
| — € : <code>\EUR{}</code> ; | — P ^r : <code>\professeur{}</code> ; |
| — ex. : <code>\ex{}</code> ; | — P ^{rs} : <code>\professeurs{}</code> ; |
| — M ^{me} : <code>\madame{}</code> ; | — St- : <code>\saint{}</code> ; |
| — M ^{lle} : <code>\mademoiselle{}</code> ; | — Sts- : <code>\saints{}</code> ; |
| — M ^{lles} : <code>\mademoiselles{}</code> ; | — Ste- : <code>\sainte{}</code> ; |
| — M ^e : <code>\maitre{}</code> ; | — Stes- : <code>\saintes{}</code> ; |
| — M ^{es} : <code>\maitres{}</code> ; | — 1 ^{er} siècle : <code>\siecle{1}</code> ; |
| — M ^{mes} : <code>\mesdames{}</code> ; | — I - II ^e siècles : <code>\siecles{1}{2}</code> . |
| — MM. : <code>\messieurs{}</code> ; | |

D’autres abréviations seront encadrées par des macro-commandes au fil des productions, cela permet de conserver une cohérence de rédaction sur l’ensemble des documents d’une part et d’autre part de pouvoir modifier d’une seule mise à jour toutes les abréviations encadrées par des macro-commandes.

Il est donc impératif de privilégier ces nouvelles instructions plutôt que de rédiger « en brut » ces abréviations. Pour les spécificités française, je renvoie encore une fois à cet [aide-mémoire](#).

D’autres instructions pour normaliser des abréviations sont déjà implémentées dans le code `.tex` et ont été abordées en ?? mais comme il s’agit d’une source de fautes typographiques importante, j’en refais l’inventaire ici :

- | | |
|----------------------------|--|
| — œ : <code>\oe{}</code> ; | — 1 ^{er} : <code>1\ier{}</code> ; |
| — Œ : <code>\OE{}</code> ; | — 2 ^e : <code>2\ieme{}</code> ; |
| — æ : <code>\ae{}</code> ; | — 1 ^{ers} : <code>1\iers{}</code> ; |
| — Æ : <code>\AE{}</code> ; | — 1 ^{re} : <code>1\iere{}</code> ; |



- 1^{res} : `1\ieres{}` ;
- 1^o : `\primo{}` ;
- 2^o : `\secundo{}` ;

- 3^o : `\tertio{}` ;
- 4^o : `\quarto{}` .

2.2.5.2 Citation

Comme pour certaines abréviations, les citations font l'objet d'un usage conventionné et sa mise en forme est déterminée dans le package AOCDF pour uniformiser toutes les citations contenues dans les documents rédigés avec ce package.

Exemple 2.3: Citation

Selon les normes en vigueur, une citation de moins de quarante mots s'intègre dans la phrase. Elle se rédige dans l'instruction `\say{citation}`, selon l'exemple suivant :

Un illustre individu a dit un jour : «une citation de moins de quarante mots qui s'intègre dans la phrase», alors que le texte initial reprend après.

Code

```
Un illustre individu a dit un jour :  
\say{une citation de moins de  
quarante mots qui s'intègre dans la  
phrase}, alors que le texte initial  
reprend après.
```

Pour les citation de plus de quarante mots, elle se sépare du corps de texte pour former un bloc. Elle se rédige dans l'instruction `\begin{displayquote}` , selon l'exemple suivant :

Un illustre individu a dit un jour :

Pellentesque interdum sapien sed nulla. Proin tincidunt. Aliquam volutpat est vel massa. Sed dolor lacus, imperdiet non, ornare non, commodo eu, neque. Integer pretium semper justo. Proin risus. Nullam id quam. Nam neque. Duis vitae wisi ullamcorper diam congue ultricies. Quisque ligula. Mauris vehicula.

Code

```
Un illustre individu a dit un jour :  
\begin{displayquote}  
Pellentesque interdum sapien sed  
nulla. Proin tincidunt. Ali- quam  
volutpat est vel massa. Sed dolor  
lacus, imperdiet non, ornare non,  
commodo eu, neque. Integer pretium  
semper justo. Proin risus. Nullam id  
quam. Nam neque. Duis vitae wisi  
ullamcorper diam congue ultricies.  
Quisque ligula. Mauris vehicula.  
\end{displayquote}
```

Il conviendra également de citer les sources de la citation à sa fin, ainsi que de la référencer dans la bibliographie du document. Cela sera abordé dans le ??.



3.1 Introduction

Avant de rédiger des documents d'une longueur conséquente, il convient impérativement de prendre en main les notions de référencement qui régissent les documents rédigés avec \LaTeX . Effectivement, plus on référence les éléments importants du document durant la rédaction, plus il sera aisé de les compiler en liste, en glossaire, en index. . .

Ces référencements permettront également de naviguer plus aisément dans les documents, par l'implantation d'*intraliens* et d'*hyperliens* qui prendront tous leurs sens lors d'une lecture sur un outil informatique type tablette.

Au rayon des grosse valeurs ajoutées de \LaTeX vient également se greffer ses outils permettant de rédiger des bibliographies normalisées à la présentation irréprochable. La prise en main de ces outils diffère quelque peu de l'utilisation classique de \LaTeX , mais elle reste indispensable – avant d'entamer toute rédaction – car il est absolument nécessaire de référencer et citer toutes les sources utilisées dans la cadre de la rédaction de documents à visée pédagogiques.

3.2 Renvois

\LaTeX présente des outils de référencement interne et externe au document pouvant faciliter grandement la navigation dans les documents par renvois. Cela permet également de communiquer avec une base de données ou avec Internet. Par exemple, dans le cadre de la rédaction de documents à visée pédagogique, cela permet de renvoyer le lecteur vers une base de données privées contenant des ouvrages de références ainsi que les sources ayant servi à rédiger le document, pour plus de précisions.

3.2.1 Intraliens

Les intraliens permettent de faciliter la navigation dans le documents. Ceux-ci font appel à un *label* et à des instructions d'insertion de label produisant des *intraliens* de couleur verte lorsque le document est paramétré pour le format écran et noir lorsque le document est paramétré pour le format papier.

3.2.1.1 Labellisation

Pour référencer un élément en vue d'être appelé en intralien plus loin dans le document, il conviendra de le *labelliser*, c'est-à-dire de lui attribuer un *label* – ou *étiquette* – spécifique à l'aide de l'instruction `\label{label de l'élément}`. Afin que celui-ci soit unique et n'entre pas en conflit avec un autre label, il y a donc quelques règles de rédaction à respecter :

- le label prend la forme générale suivante : *objet:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis)* avec des objets définis selon la liste suivante ;
- les caractères spéciaux français (accent, cédille. . .), les espaces et les signes de ponctuations sont à bannir ;
- les labels pour les exemples, définitions et formules sont inclus dans l'instruction les appelant, cela est abordés dans la [section 3.4](#).



Voici la liste des objets auxquels les rédacteurs vont faire face, il convient d'impérativement respecter cette nomenclature (similaire aux noms des fichiers des sous-programmations détaillés dans la [sous-section 1.5.1](#)) :

chap:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour un chapitre ;
ann:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour une annexe ;
sec:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour une section ;
subsec:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour une sous-section ;
subsubsec:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour une sous-sous-section ;
fig:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour une figure ;
tab:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour un tableau ;
ex:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour un exemple (intra-lien seulement) ;
def:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour une définition (intra-lien seulement) ;
form:mot-clé-général_mot-clé-précis(__mot-clé-plus-précis) : label pour une formule (intra-lien seulement).

3.2.1.2 Insertion intraliens

Il existe plusieurs instructions permettant d'insérer des intraliens dans un texte, selon le format que le rédacteur souhaite lui donner.

Exemple 3.1: Intralien

Il existe une série d'instructions pour mettre en forme l'intralien selon la situation.

Ce premier exemple de code met en évidence simplement l'interaction entre un label produit par l'instruction `\label{label de l'élément}` et l'instruction `\ref{label de l'élément}` produisant l'intralien :

Ce texte comporte une figure [3.1](#) labellisée avec une référence.

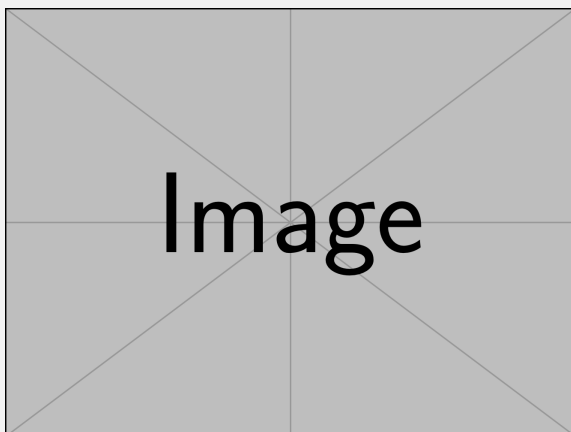


FIG. 3.1 – Image pour le premier exemple d'intralien

On remarquera que l'instruction `\ref` ne transforme que le seul numéro de la figure en intralien. Pour améliorer cela, il existe l'instruction `\autoref{label de l'élément}` qui va nommer automatiquement l'objet du lien et étendre la zone disponible pour cliquer, cela sera très pratique dans le cadre d'un usage interactif :

```
Code
Ce texte comporte une figure
\ref{fig:image_exemple_intralien_1}
labellisée avec une référence.
\begin{figure}[H]
\includegraphics[width=
\linewidth]{fig_image.png}
\caption{Image pour le premier
exemple d'intralien\label{fig:image_
exemple_intralien_1}}
\end{figure}
```



Ce texte comporte une [figure 3.2](#) labellisée avec une référence.

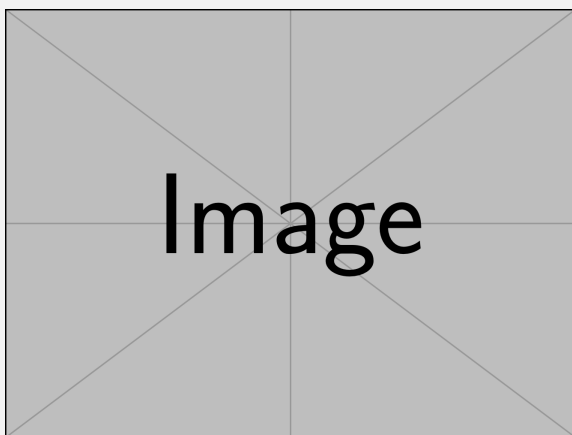


FIG. 3.2 – Image pour le deuxième exemple d'intralien

Pour les documents à usage papier, les intraliens ne produiront aucun effet interactif. Néanmoins, les renvois peuvent être spécifiés à l'aide de l'instruction `\autopageref{label de l'élément}`, qui va renseigner la page sur laquelle se situe l'élément labellisé :

Ce texte comporte une figure [page 19](#) labellisée avec une référence.

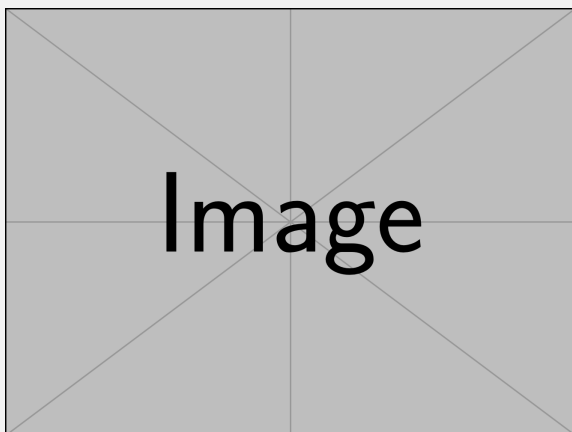


FIG. 3.3 – Image pour le troisième exemple d'intralien

Et enfin, la macro-commande `\superref{label de l'élément}` permettra, selon les paramètres `media` ou `paper` renseigné dans `master.tex`, d'appeler soit l'instruction `\autoref{label de l'élément}` pour les documents à usage interactifs, soit la combinaison des instructions `\autoref{label de l'élément}` et `\autopageref{label de l'élément}` pour les documents à usage papier.

Il conviendra de privilégier cette solution pour un maximum de flexibilité sur l'ensemble des documents rédigés :

Code

```
Ce texte comporte une
\autoref{fig:image_exemple_intralien_
2} labellisée avec une
référence.
\begin{figure}[H]
\includegraphics[width=
\linewidth]{fig_image.png}
\caption{Image pour le deuxième
exemple d'intralien\label{fig:image_
exemple_intralien_2}}
\end{figure}
```

Code

```
Ce texte comporte une figure
\autopageref{fig:image_exemple_
intralien_3} labellisée avec une
référence.
\begin{figure}[H]
\includegraphics[width=
\linewidth]{fig_image.png}
\caption{Image pour le troisième
exemple d'intralien\label{fig:image_
exemple_intralien_3}}
\end{figure}
```



Ce texte comporte une [figure 3.4](#) labellisée avec une référence.

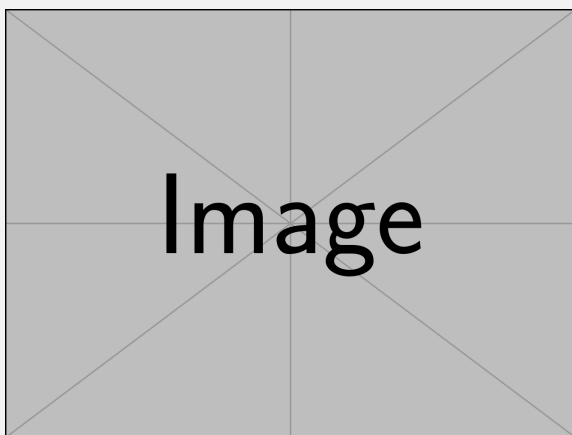


FIG. 3.4 – Image pour le quatrième exemple d'intralien

Code

Ce texte comporte une [figure 3.4](#) labellisée avec une référence.

```
\begin{figure}[H]
\includegraphics[width=
\linewidth]{fig_image.png}
\caption{Image pour le quatrième
exemple d'intralien\label{fig:image_
exemple_intralien_4}}
\end{figure}
```

3.2.2 Hyperliens

Les *hyperliens* sont des liens permettant de renvoyer le lecteur vers une adresse située en dehors du document, que ça soit sur un serveur privé, une base de données ou encore vers un lien internet. Ces liens sont de couleur bleue lorsque le document est paramétré pour le format écran et noir lorsque le document est paramétré pour le format papier.

Exemple 3.2: Hyperlien

Les hyperliens sont appelés avec deux instructions. La première instruction `\url{adresse du lien}` affiche simplement l'adresse du lien référencé. La deuxième instruction `\href{adresse du lien}{texte cliquable}` renvoie également à l'adresse du lien référencée mais en remplacement cette adresse par un texte cliquable. Encore une fois, ces instructions sont conditionnées par l'usage du document – interactif ou papier – et une macro-commande `\Href{adresse du lien}{texte cliquable descriptif du lien}` a été créée pour respecter cette condition :

Ce texte comporte un hyperlien qui renvoie vers le [portail Github](#) de ce template.

Code

Ce texte comporte un hyperlien qui renvoie vers le [portail Github](#) de ce template.

```
\Href{https://github.com/aocdtf-mta
/AOCDTF-template}{portail Github} de
ce template.
```

3.3 Index et glossaire

La création d'index et de glossaires sont également des démarches à appréhender *avant* d'entamer la rédaction de documents.

Pour les index, cela permet de référencer par ordre alphabétique dans une liste d'intraliens tous les mots-clés avec leurs pages respectives pour lesquels cela est jugé nécessaire. On peut également produire des index suivant des thématiques spécifiques.

Pour les glossaires, cela permet de référencer en liste d'intraliens tous les mots-clés et leur définitions respectives pour lesquels cela est jugé nécessaire. Cet outil permet aussi de faciliter la rédaction



d'acronymes et de définitions avec la rédaction d'*entrées de glossaire* dans des fichiers de code `.tex` séparés, qui se comportent comme des macro-commandes pouvant être utilisées durant la rédaction du corps de texte.

Tout comme pour les index, on peut également produire des glossaires suivant des thématiques spécifiques.

Le package AOCDFTF prévoit déjà des environnements thématiques référencés en plusieurs listes après la table des matières, cela est abordé dans la [section 3.4](#). À l'usage, l'index et le glossaire peuvent leur faire office de doublon et il ne seront donc pas forcément pertinent dans les documents rédigés pour l'AOCDFTF.

Toutefois, il me semble important d'aborder ces aspects-là de la rédaction car cela peut être une exigence sur l'un ou l'autre document, qu'il vaut mieux assurer dès le début de la rédaction. Effectivement, il s'agit d'indexer et de remplacer tous les acronymes et mots-clés définis à l'aide d'instructions spécifiques tout au long du document, et réaliser ces tâches après la rédaction sur une petite centaine de pages peut nécessiter un temps conséquent et être source d'oubli.

Exemple 3.3: Index

Pour produire un index standard, il est nécessaire de renseigner l'instruction `\makeindex` en préambule dans le code `master.tex`, ainsi que l'instruction `\printindex` dans le corps de document du code `master.tex`, à l'endroit du code où l'on souhaitera voir apparaître l'index, a priori en fin de document. Dans le texte, les mots-clé sont indexés avec l'instruction `\index{mot-clé à indexer}` :

Ce texte comporte un mot-clé indexé qui sera référencé dans une liste.

Code

Ce texte comporte un mot-clé `\index{<mot-clé>}` indexé qui sera référencé dans une liste.

Les entrées dans l'index peuvent se faire sur plusieurs niveaux, si certains mots-clé ne sont que des précisions de mots-clé plus globaux avec l'instruction `\index{mot-clé!sous_mot-clé}`. On peut également au renvoi de terme précis vers un terme plus généraliste avec l'instruction `\index{sous_mot-clé|see{mot-clé}}` :

Ce texte comporte des mots-clés indexés qui seront référencés dans une liste. Ce répertoire liste par ordre alphabétique tous les mots-clés indexés avec leurs pages respectives, cela permet au lecteur d'aisément retrouver ces termes jugés significatifs dans le document.

Code

Ce texte comporte des mots-clés indexés qui seront référencés dans une liste `\index{liste}`. Ce répertoire `\index{liste!répertoire}` liste par ordre alphabétique tous les mots-clés `\index{<mot-clé>}` indexés avec leurs pages respectives, cela permet au lecteur d'aisément retrouver ces termes `\index{<terme|see{mot-clé}>}` jugés significatifs dans le document.

On peut également produire des index selon des thématiques spécifiques qui regrouperont donc par ordre alphabétique une série de mots-clés se référant à ce thème. L'instruction pour initier la création d'un index thématique est la suivante :

```
\makeindex[program=makeindex, columns=<3>, intoc=<true>, options={-s index_style.ist}, name=<nom_index>, title=<Titre de l'index thématique>]
```



Avec le paramètre `name=<nom_index>` définissant l'appellation du thème qui sera à utiliser dans les instructions d'indexation `\index[<nom du thème>]{<mot-clé se référant au thème>}`, et le paramètre `title=<Titre de l'index thématique>` qui produira le titre de l'index une fois appelé dans le code `master.tex`.

Et dans le texte, l'indexation des mots-clés se référant à ce thème s'effectue donc avec l'instruction `\index[<nom du thème>]{<mot-clé se référant au thème>}`. L'exemple suivant explicite les index thématique selon un thème à deviner :

Ce texte comporte des mots-clés indexés qui seront référencés dans une liste. Ce répertoire situé généralement en fin de document liste par ordre alphabétique tous les mots-clés indexés avec leurs pages respectives, cela permet au lecteur d'aisément retrouver ces termes jugés significatifs dans le document.

Code

```
Ce texte comporte des mots-clés
indexés\index[exemple]{indexé} qui
seront
référéncés\index[exemple]{référéncé}
dans une liste. Ce répertoire
situé\index[exemple]{situé}
généralement en fin de document liste
par ordre alphabétique tous les
mots-clés indexés avec leurs pages
respectives, cela permet au lecteur
d'aisément retrouver ces termes
jugés\index[exemple]{jugé}
significatifs dans le document.
```

Les index sont à retrouver en fin du code `master.tex`, avec les instructions `\printindex` et `\printindex[exemple]`, qu'il conviendra de mettre en commentaire selon le contexte de rédaction.

Exemple 3.4: Glossaire

Pour produire un glossaire standard, la démarche est similaire que pour produire un index, il s'agit d'indiquer les instructions `\makenoidxglossaries` en préambule et `\printnoidxglossary` dans le corps de document du code `master.tex`, à l'endroit du code ou l'on souhaitera voir apparaître le glossaire, à priori en fin de document.

Il conviendra de renseigner également des entrées du glossaire non plus dans le corps du texte mais dans deux codes séparés, `glossary_entry.tex` et `acronym_entry.tex`, un peu dans la même démarche que pour les entrées de clés bibliographiques. Effectivement, dans l'hypthèse où le document rédigé est conséquent, le nombre d'entrées le sera aussi et il s'agit de ne pas encombrer le code `master.tex` avec celles-ci.

Ces entrées peuvent prendre deux formats, avec les instructions :

```
acronyme : \newacronym{<label_acronyme>}{<Acronyme>}{<description de
l'acronyme>} ;
```

```
entrée « classique » : \newglossaryentry{<label_entree_glossaire>}{type=<nom_
glossaire>, name={<caractères à afficher dans le texte>},
description={<définition dans le glossaire>}, sort={<nom de l'entrée
dans le glossaire>}} ;
```

Les entrées du glossaire définies dans les fichiers correspondant sont appelées dans le corps avec l'instruction `\gls{label}`, avec le `label` valable autant pour les acronymes que pour les entrées « classiques » :



Ce texte comporte un acronyme tel que l'**AOCDF** ou encore un **exemple** comportant sa définition dans le glossaire.

Code

Ce texte comporte un acronyme tel que `\gls{aocdf}` ou encore un `\gls{ex}` comportant sa définition dans le glossaire.

Il existe plusieurs options – un exemple est à retrouver dans le code `master.tex` – permettant de produire des glossaires thématiques (notation, acronyme...) mais elles ne seront pas abordées ici car il est peu probable que les documents à destination de l'**AOCDF** nécessitent un glossaire (à moins de rédiger un livre de référence). Le package AOCDF fait le choix de produire des listes de différents éléments par thématiques plutôt que des index et glossaires, sujet de la section suivante.

Toutefois, référencer les acronymes permet de conserver une main-mise sur des éléments du texte qui peuvent voir leur mise en forme varier d'un rédacteur à l'autre, cela peut se montrer pertinent de les répertorier si le rédacteur constate qu'il apparaît un nombre conséquent d'acronymes dans son document.

3.4 Environnements référencés

Pour accompagner la visée pédagogique des documents destinés à l'**AOCDF**, il fut créé trois environnements spécifiques – pour le moment – destinés à mettre en évidence les formules, les définitions et les exemples dans le texte. Cela permet également de conserver la main-mise sur la mise en forme de tous ces environnements d'une simple mise-à-jour.

Ceux-ci sont compilés dans trois listes, auxquelles s'ajoutent également les listes des figures et des tableaux, présentes dans le *frontmatter* après la table des matières. Toutes celles-ci sont composées d'intraliens renvoyant vers leurs emplacements respectifs dans le document, pour faciliter la navigation dans celui-ci en usage interactif.

Exemple 3.5: Environnement exemple

Cet environnement est mis en évidence en gris et bleu dans le corps de texte. Il est appelé avec l'instruction `\begin{exemple}{<titre de l'exemple>}{<label (sans l'objet)>}`.

Il est donc plus aisément référencé mais il n'est pas nécessaire de lui indiquer son objet, qui est `ex:`, car cela est déjà paramétré dans le package AOCDF. Il n'est donc pas nécessaire de renseigner l'instruction `\label{<ex:mot-clé_mot-clé_plus_précis>}`.

Par contre, son label pour produire un intralieu selon l'instruction `\superref{<ex:mot-clé_mot-clé_plus_précis>}` devra quant à lui indiquer cet objet `ex:`, comme pour tous les autres objets labellisés :

Exemple 3.6: Exemple d'environnement exemple

Cet environnement comporte un exemple (calcul, mise en situation...) se situant dans l'**exemple 3.6**.

Code

```
\begin{exemple}{Exemple
d'environnement
exemple}{exemple_
environnement_exemple}
Cet environnement comporte un
exemple (calcul, mise en
situation\ldots) se situant
dans l'\superref{ex:exemple_
environnement_exemple}.
\end{exemple}
```



Cet environnement peut également ne pas être numéroté ni listé avec l'instruction `\begin{exemple*}{<titre de l'exemple>}{<label (sans l'objet)>}` :

Exemple: Exemple d'environnement exemple non référencé

Cet environnement comporte un exemple (calcul, mise en situation...) qui n'est pas numéroté ni listé.

Code

```
\begin{exemple*}{Exemple
d'environnement exemple non
référéncé}{}
Cet environnement comporte un
exemple (calcul, mise en
situation\ldots) qui n'est
pas numéroté ni listé.
\end{exemple*}
```

Exemple 3.7: Environnement définition

Cet environnement est mis évidence en gris et rose dans le corps de texte. Il est appelé avec l'instruction `\begin{formule}{<titre de la définition>}{<label (sans l'objet)>}`. Il est donc plus aisément référencé mais il n'est pas nécessaire de lui indiquer son objet, qui est `def:`, car cela est déjà paramétré dans le package AOCDTF. Il n'est donc pas nécessaire de renseigner l'instruction `\label{<def:mot-clé_mot-clé_plus_précis>}`. Par contre, son label pour produire un intralieu selon l'instruction `\superref{<def:mot-clé_mot-clé_plus_précis>}` devra quant à lui indiquer cet objet `def:`, comme pour tous les autres objets labellisés :

Définition 3.1: Exemple d'environnement definition

Cet environnement comporte une définition.

Code

```
\begin{definition}{Exemple
d'environnement
definition}{exemple_
environnement_definition}
Cet environnement comporte
une définition.
\end{definition}
```

Cet environnement peut également ne pas être numéroté ni listé avec l'instruction `\begin{definition*}{<titre de la définition>}{<label (sans l'objet)>}` :

Définition: Exemple d'environnement definition non référencé

Cet environnement comporte une définition qui n'est pas numérotée ni listée.

Code

```
\begin{definition*}{Exemple
d'environnement definition
non référencé}{}
Cet environnement comporte
une définition qui n'est pas
numérotée ni listée.
\end{definition*}
```

Exemple 3.8: Environnement formule

Cet environnement est mis évidence en gris et vert dans le corps de texte. Il est appelé avec l'instruction `\begin{formule}{<titre de la formule>}{<label (sans l'objet)>}`. Il est donc plus aisément référencé mais il n'est pas nécessaire de lui indiquer son objet, qui est `form:`, car cela est déjà paramétré dans le package AOCDTF. Il n'est donc pas nécessaire de renseigner l'instruction `\label{<form:mot-clé_mot-clé_plus_précis>}`. Par contre, son label pour produire un intralieu selon l'instruction



`\superref{<form:mot-clé_mot-clé_plus_précis>}` devra quant à lui indiquer cet objet `form:`, comme pour tous les autres objets labellisés :

Formule 3.1: Exemple d'environnement formule

Cet environnement comporte une formule.

Cet environnement peut également ne pas être numéroté ni listé avec l'instruction `\begin{formule*}{<titre de la formule>}{<label (sans l'objet)>}` :

Formule: Exemple d'environnement formule non référencé

Cet environnement comporte une formule qui n'est pas numérotée ni listée.

Code

```
\begin{formule}{Exemple
d'environnement
formule}{exemple_
environnement_formule}
Cet environnement comporte
une formule.
\end{formule}
```

Code

```
\begin{formule*}{Exemple
d'environnement formule non
référéncé}{}
Cet environnement comporte
une formule qui n'est pas
numérotée ni listée.
\end{formule*}
```

3.5 Bibliographie

Monter une bibliographie est primordial dans le cadre de la diffusion de documents à visée pédagogique. Effectivement, il est indispensable de citer correctement ses sources et c'est là une des grandes forces de \LaTeX . Son moteur bibliographique présente un principe de fonctionnement un peu spécifique par rapport au reste de la programmation \TeX , qui nécessite une prise en main bien acquise avant la rédaction de documents.

Ce principe se rapproche de celui utilisé pour les glossaires, c'est-à-dire que l'on va *externaliser* les entrées bibliographiques – les *sources* utilisées – dans des fichiers de codes spécifiques. Ceux-ci ne sont plus sous format `.tex` mais seront sous format `.bib`, prévu pour les contenus bibliographiques. Une fois ces codes remplis d'entrées bibliographiques, on va appeler les *clés d'identifications* de ces entrées que l'on utilise pour la rédaction du document, pour une citation spécifique dans le corps de ou encore en fin de document.

3.5.1 Code bibliographique

Il y a donc un fichier de code `cours.bib` par *matière*, c'est la classification thématique voulue pour éviter une multiplication de ces fichiers de code. Ils sont *communs* à toutes les matières et sont donc situés au plus bas niveau dans l'arborescence de fichiers des différents dépôts.

3.5.1.1 Inclusion de codes bibliographiques

Ces fichiers `cours.bib` sont appelés dans le code de la même manière que l'inclusion de fichiers `code.tex`, mais avec l'instruction `\addbibresource{<cours.bib>}` située dans le préambule du code `master.tex`.

Cette démarche est déjà exécutée dans le code `master.tex` du template, et si l'arborescence initiale



des fichiers est respectée pour la production de documents, elle ne devrait plus à être exécutée. Toutefois, dans l'optique de couvrir un maximum de cas de figures dans ce recueil, la méthode est ici détaillée.

Tout comme avec la commande `\input{file}`, il faut identifier le chemin d'accès aux fichiers `cours.bib`. Pour cela, on va utiliser un fichier nommé `INDICATEUR_ARBORESCENCE.tex` qui est situé dans le même dossier que les codes bibliographiques.

Il faut l'inclure dans le code `master.tex`, à l'aide de l'instruction `\input{file}` appelée avec les outils *Texmaker* depuis `Latex > \input{file}`. Procéder de la sorte permet au rédacteur de sélectionner le fichier `INDICATEUR_ARBORESCENCE.tex` depuis un explorateur de fichier. Cela fera apparaître le chemin d'accès pour les fichiers de code `code.tex` qu'il suffira de copier/coller dans l'instruction `\addbibresource{<chemin d'accès copié/cours.bib>}` pour inclure les fichiers de codes bibliographiques.

3.5.1.2 Structure de codes bibliographiques

Les codes `cours.bib` présente une structure de code très simple comparée aux fichiers de code `.tex`, car ils ne comportent que les entrées bibliographiques de *toutes* les sources utilisées pour la rédaction de *tous* les documents produits sur un dépôt (typiquement tous les documents à destination d'une formation). Ces entrées sont identifiées par des *clés*, qui sont similaire aux labels des références et qui doivent suivre une nomenclature bien précise.

Les codes au format `.bib` n'utilise pas exactement la même nomenclature que les codes `.tex`, et il convient de s'inspirer du code `exemple.bib` pour la rédaction de nouveaux codes bibliographiques ainsi que l'entrée de nouvelles références.



Annexes



Bibliographie

- [1] AUTEUR. « Title de l'ouvrage ». Dans : *Collection*. Dir. EDITEUR. Maison d'édition, Année.
- [2] AUTEUR. *Titre de la page*. Organisation. 1^{er} jan. 1900. URL : [Lien%20de%20la%20page](#).
- [3] *Titre de la norme*. Organisme de normalisation. 1900.



Index

L

liste 18
répertoire 18

M

mot-clé 18

T

terme *voir* mot-clé



Index des termes du thème exemple

I

indexé 19

J

jugé 19

R

référéncé 19

S

situé 19



Glossaire

exemple définition de l'exemple d'entrée classique dans le glossaire 23



Liste des acronymes

AOCDTF Association Ouvrière des Compagnons du Devoir et du Tour de France 4, 11, 13, 21, 23

IDE Environnement de Développement (Integrated Development Environment) 12



Temporary page!

L^AT_EX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because L^AT_EX now knows how many pages to expect for this document.