

Matière

Cours

CHOISIR BLASON
/fichiers_multimedia



FORMATION

Matière

Cours

CHOIR BLASON
fichiers_multimedia

Prénom NOM

CHOIR BLASON
fichiers_multimedia

Prénom NOM

Édition 2021.7



Patrimoine
Culturel
Immatériel
en France



CONFÉRENCE DES
GRANDES
ÉCOLES



PS652018



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	6
2 Rédaction de texte	9
2.1 Structuration du texte	9
2.2 Mise en forme	13
3 Référencement et bibliographie	17
3.1 Introduction	17
3.2 Renvois	17
3.3 Index et glossaire	21
3.4 Environnements référencés	23
3.5 Bibliographie	25
4 Écriture scientifique	31
4.1 Introduction	31
4.2 Notation mathématique	31
4.3 Grandeurs et unité de mesure	34
4.4 Environnement scientifique	35
5 Tableau	39
5.1 Introduction	39
5.2 Préceptes typographiques	39
5.3 Structure de base	40
5.4 Tableau avec largeur relative	41
Annexes	43
Bibliographie	45
Index	47
Index des termes du thème exemple	49
Glossaire	51
Liste des acronymes	53



Liste des tableaux

4.1	Alphabet grec	34
5.1	Un premier tableau simple avec <code>tabular</code>	40
5.2	Un deuxième tableau avec <code>tabularx</code>	42



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
4.1	Probabilité d'électrocution	37
4.2	Valeur expérimentale de l'eV	38



Liste des définitions

3.1	Exemple d'environnement definition	24
4.1	Unité de mesure	34
4.2	Grandeur physique	34
4.3	Dimension	34



Liste des exemples

1.1	Notation des siècles	1
1.2	Style du chapitre	2
1.3	Minipage	4
1.4	Type de média	7
2.1	Texte en trois colonnes	10
2.2	Note en pied de page	11
2.3	Liste et énumération	11
2.4	Citation	15
3.1	Intralien	18
3.2	Hyperlien	20
3.3	Index	22
3.4	Glossaire	23
3.5	Environnement exemple	24
3.6	Exemple d'environnement exemple	24
3.7	Environnement définition	24
3.8	Environnement formule	25
3.9	Entrée bibliographique	27
3.10	Citation bibliographique	28
4.1	Notation mathématique	31
4.2	Notation scientifique	35
4.3	Formule avec détails	36
5.1	Tableau simple	39
5.2	Tableau à la largeur relative	41



Préface

Ce *template* contient un recueil neutre des programmations *types* dont les programmeurs devront fortement s'inspirer 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 types. La liste des exemples donne également une navigation rapide parmi les 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 leur 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 \LaTeX

1.1 Valeurs ajoutées de \LaTeX

\LaTeX 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 \TeX .

La première valeur ajoutée de \LaTeX 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 \LaTeX est d'automatiser certains processus de rédaction par la création de nouvelles instructions (ou l'utilisation d'instructions existantes). Le package AOCDFTF comporte ainsi une multitude de *macro-commandes* aux buts aussi variés qu'utiles, je vous conseille d'aller consulter le code (<https://github.com/aocdf-tf-mta/AOCDFTF-package/blob/main/AOCDFTF.sty>) du package AOCDFTF 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 tout ce qui constitue le *fond* du document est *à priori* encadrée par une instruction précise, chacune d'entre elles 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 page 24) :

Exemple 1.1: Notation des siècles

Cela sera abordé dans la sous-sous-section 2.2.5.1 page 14, 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 AOCDFTF contient une nouvelle macro-commande définie comme telle :

Code

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



Lorsqu'on appelle l'instruction `\sicle{1}`, cela produira I^{er} siècle. Si l'on appelle l'instruction `\sicle{2}`, cela produira II^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 `\mintinline{latex}{\sicle{1}}`, cela produira `\sicle{1}`. Si l'on appelle l'instruction `\mintinline{latex}{\sicle{2}}`, cela produira `\sicle{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 personnalisation du style de chapitre a demandé une quinzaine de jour. Son bloc de code se situe en fin du package AOCDTF (<https://github.com/aocdtf-mta/AOCDTF-package/blob/main/AOCDTF.sty>), 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

Le style du titre de chapitre pourra être modifié ultérieurement d'un simple mise à jour pour tous les documents rédigés avec le package AOCDTF 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 ?? page ??).

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 page 4.

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 (<https://github.com/aocdtf-mta/AOCDTF-template/wiki/Décomposition-du-code-master.tex>) 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, selon l'instruction-type suivante `\division*{<titre de la division non référencée>}` .

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

À la compilation du code, il se peut que les flottants et autres éléments occupant l'espace contraignent 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

Une minipage est appelée avec l'environnement `\begin{minipage}[<position>]{<largeur>}`. Les instructions suivantes produiront donc deux « minipages », qui s'aligneront selon leurs paramètres d'alignement. Celle de gauche fera 8cm et va aligner le bas de son environnement – variable [b] – avec le haut de l'environnement – variable [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.

L'alignement vertical de cet environnement sur la page est paramétré selon la variable définissant l'argument facultatif [<position>] :

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 ou l'environnement sera appelé dans le code correspondant.

Il se paramètre également sur la largeur de la minipage selon la variable définissant l'argument obligatoire `{<largeur>}` :

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 L^AT_EX).

1.4.4 Espace de rédaction

Dans le package AOCDF 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. Selon les instructions entrées dans le code `master.tex` sur le type de média, les marges verticales 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.

1.4.5 Flottants

Les tableaux et les figures peuvent être traités comme des élément *flottants*, c'est-à-dire qu'il sont considérés hors du corps de texte et voient leurs emplacements non précisément défini par le rédacteur. Celui-ci va définir les paramètres d'emplacement généraux et c'est L^AT_EX qui se chargera du placement automatique des éléments flottants à la compilation. Cela permet d'éviter les problèmes d'insertions d'éléments qui mettent en désordre l'intégralité d'un document. C'est également aux éléments flottants que l'on associe des légendes référencées et des listes les compilant dans le *frontmatter*.

Pour un tableau ou une figure que l'on souhaite définir en tant qu'élément *flottant*, on fait appel aux environnements `\begin{table}[<emplacement>]` et `\begin{figure}[<emplacement>]`. L'argument `[<emplacement>]` donne des indications concernant le placement sur page selon les variables suivantes :

haut de la page (top) : `t` ;

milieu de la page (middle) : `m` ;

bas de la page (bottom) : `b` ;

sur une page dédiée (page) : `p` ;

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

force l'emplacement défini par le code (HERE) : `H`.

Horizontalement, les éléments flottants peuvent être alignés avec les instructions suivantes insérées en début des environnements `table` ou `figure` :

aligné à gauche : `\raggedright` ;

centré : `\centering` ;

aligné à droite : `\raggedleft` .

La légende de l'élément flottant est insérée dans ces environnements avec l'instruction `\caption{<légende de l'élément flottant>}`, qui sera toujours située en dessous de l'élément flottant. La version étoilée `\caption*{<légende de l'élément flottant non référencé>}` supprime la numérotation et le référencement de cet élément flottant.

Cette légende sera référencée dans une liste après la table des matières si le rédacteur le souhaite en activant les instructions situées dans le code `master.tex` :



liste des tableaux : `\hypersetup{linkcolor=black}\listoftables*` ;

liste des figures : `\hypersetup{linkcolor=black}\listoffigures*` .

Il s'agit donc d'éléments *indépendants* qui ne sont donc pas prévus être insérés dans d'autres environnements (exemple, minipage...) sous peine probable d'erreur de compilation. Cela dit, s'il est nécessaire d'insérer un tableau ou figure avec une légende dans un environnement ou une minipage, il existe deux solutions selon les erreurs de compilation :

sans environnement flottant : utilisation l'instruction `\captionof{<type d'élément flottant>}{<légende de l'élément flottant>}` ;

avec environnement flottant : mention la variable H dans l'argument [`<emplacement>`] .

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 emplacements spécifiés, ou encore pour que les référencements (table des matières, bibliographie...) soient correctement exécutés.

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, la démarche pour paramétrer cette **Compilation rapide** est détaillée sur cette page (<https://github.com/aocdtf-mta/AOCDTF-template/wiki/Paramètres,-trucs-et-astuces-sur-Texmaker>) du wiki.

1.5.1 Sous-programmation

Lorsqu'on rédige un document conséquent, comme abordé sur cette page (<https://github.com/aocdtf-mta/AOCDTF-template/wiki/Structure-du-code-master.tex>) du wiki, on fait appel à une architecture de code 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{<chemin d'accès/objet_mot-clé_mot-clé_plus_précis>}` dans le but de aérer ce dernier.

Les sous-programmations peuvent également être des tableaux ou des figures aux nombres de lignes de code trop importants pour être insérées directement dans le corps de texte. Elles sont injectées dans la programmation parente – généralement un chapitre – avec l'instruction `\input{<chemin d'accès/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.

Ces commandes impliquent d'identifier les chemin d'accès des sous-programmations, *relatifs* aux programmations parentes. Pour faciliter leurs usages, il faut *appeler ces instructions avec les outils Texmaker* depuis Latex > `\input{file}` . Procéder de la sorte permet au rédacteur de sélectionner le fichier souhaité depuis un explorateur de fichier et cela fera apparaître le chemin d'accès aux fichiers de code `code.tex`.

Pour information, un chemin d'accès *absolu* indique l'emplacement exact du fichier dans l'ordinateur. Il est possible d'utiliser ce format de chemin d'accès mais il créera une erreur de compilation en cas de collaboration sur un même fichier de code. Le chemin d'accès *relatif* au code parent, tant que l'on fait appel à des fichiers situés dans le dépôt, ne produira pas d'erreur de compilation car tous les collaborateurs sont censés utiliser la même arborescence concernant les fichiers du dépôt.

Comme explicité à cette page (<https://github.com/aocdtf-mta/AOCDTF-template/wiki/Architecture-du-template>) 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 :

```

chapitre : chap_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
annexe : ann_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
section (rare) : sec_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
tableau : tab_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
figure : fig_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
équation : eq_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
graphique : graph_mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) .

```

Pour rédiger une sous-programmation, il suffit de suivre les instructions à cette page (<https://github.com/aocdtf-mta/AOCDTF-template/wiki/Structure-et-usage-du-code-MWE.tex>) 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 AOCDTF 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 (<https://github.com/aocdtf-mta/AOCDTF-template/wiki/Décomposition-du-code-master.tex>) 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}` appelée en préambule. 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 les variables `screen` ou `paper` renseignées dans l'argument `{<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 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 `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
\mintinline{latex}{\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
\mintinline{latex}{\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 dans le code) ;
- pour aborder un nouveau paragraphe, il faut appeler l'instruction `\` suivi de *deux* sauts de ligne dans le code ;
- sauter une ligne dans le code sans renseigner l'instruction `\` ne produira pas de saut de ligne à la compilation ;
- la structuration du texte suit la réglementation française :
 - un saut de ligne dans le code 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 dans le code ne changera pas le résultat. Toutefois, pour la lisibilité du code, il est conseillé de faire 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 l'instruction `\textbackslash` produira la contre-oblique.

\LaTeX gère les espaces entre les mots selon la réglementation française, il convient dès lors d'uniformiser la rédaction avec ses subtilités concernant les espaces, les signes de ponctuations utilisés... Toutes celles-ci sont compilées dans cet aide-mémoire (<http://www.ipgp.fr/~moguilyn/LaTeX/typo.pdf>).

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 effectuée même sur les lettres majuscules à l'aide des instructions `\`E`, `\^E` ou encore `\A` (ou juste È et Â 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 (https://fr.wikibooks.org/wiki/LaTeX/Éléments_de_base) ainsi que les macro-commandes correspondantes à la sous-sous-section 2.2.5.1 page 14 ;
- utilisation des points de suspensions comme ponctuation de fin de phrase – jamais après une virgule – avec l’instruction `\ldots` ;
- utilisation du symbole de l’euro € avec l’instruction `\EUR{}` .

Les espaces sont finement gérés par L^AT_EX et il n’est à priori pas nécessaire de les appeler durant la programmation. Néanmoins, voici la liste des différents types 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 les variables `\linewidth` ou `\textwidth` . Celles-ci renvoient à la même valeur dans le cas d’un texte à une colonne. Dans le cas d’un texte à plusieurs colonnes, `\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 typographie 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 droite> \end{raggedleft}` ;

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

Le texte peut se répartir également sur plusieurs colonnes avec l’environnement `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 `\\` :

Code

```
\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.

Pour préciser certains éléments du texte sans le surcharger, on peut également rédiger des notes en bas de page.

Exemple 2.2: Note en pied de page

Des notes en bas de page peuvent également être insérées depuis un intralieu en indice^a dans le corps de texte avec l'instruction `\footnote{<note en pied de page>}`.

^a. Cela produit une note en pied de page.

Code

Des notes en bas de page peuvent également être insérées depuis un intralieu en indice `\footnote{Cela produit une note en pied de page.}` dans le corps de texte avec l'instruction `\mintinline{latex}{\footnote{<note en pied de page>}}`.

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 2.3: Liste et énumération

Une liste se rédige dans un environnement `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.

Code

```
\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}
```

Les listes rédigées pour les documents destinés à l'AOCDFTF 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, sauf pour le dernier élément de la liste et les phrases clôturées 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 `enumerate`, ainsi que comporter plusieurs niveaux de listes imbriqués :

Début du paragraphe incluant une 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.

Suite du texte sans alinéa car la liste est incluse dans le paragraphe.

Code

```
Début du paragraphe incluant une liste
\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}
Suite du texte sans alinéa car la liste est
incluse dans le paragraphe.
```

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

premier élément description du premier élément ;

deuxième élément description du deuxième élément ;

troisième élément description du troisième élément.

Nouveau paragraphe (avec alinéa) suite à l'insertion d'un saut de ligne `\\` à la fin du dernier *item* et d'un saut de ligne du code après l'environnement de liste.

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}

Nouveau paragraphe (avec alinéa) suite à
l'insertion d'un saut de ligne
\mintinline{latex}{\\} \emph{à la fin} du dernier
\emph{item} et d'un saut de ligne du code après
l'environnement de liste.
```

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

Comme explicité dans l'exemple de description, selon que l'on souhaite entamer un nouveau paragraphe ou non après la liste, il faut clôturer le dernier élément avec la commande `\\` et effectuer un saut de ligne du code après l'environnement de liste. Le comportement de la liste est de fait identique au à l'agencement du texte détaillée en sous-section 2.1.1 page 9 à cette notion près donc que l'instruction du saut de ligne `\\` est incluse dans les environnements de liste et non insérée après ceux-ci comme on pourrait spontanément le faire. 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 ?? page ??.



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 caractère 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 possibles 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'AOCDTF. 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{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'AOCDTF. 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 (<http://www.ipgp.fr/~moguilyn/LaTeX/typo.pdf>) – 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 AOCDFTF contient quelques nouvelles macro-commandes listées ci-dessous :

- | | |
|--|---|
| — an 2000 : <code>\annee{2000}</code> ; | — M ^{lle} : <code>\mademoiselle{}</code> ; |
| — an 50 (avant J.-C.) : <code>\annee{-50}</code> ; | — M ^{lles} : <code>\mademoiselles{}</code> ; |
| — apr. J.-C. : <code>\apjc</code> ; | — M ^e : <code>\maitre{}</code> ; |
| — av. J.-C. : <code>\avjc</code> ; | — M ^{es} : <code>\maitres{}</code> ; |
| — D ^r : <code>\docteur{}</code> ; | — M ^{mes} : <code>\mesdames{}</code> ; |
| — D ^{rs} : <code>\docteurs{}</code> ; | — MM. : <code>\messieurs{}</code> ; |
| — € : <code>\EUR{}</code> ; | — I ^{er} millénaire : <code>\millenaire{1}</code> ; |
| — ex. : <code>\ex{}</code> ; | — I - II ^e millénaires : <code>\millenaires{1}{2}</code> ; |
| — M ^{me} : <code>\madame{}</code> ; | — M. : <code>\monsieur{}</code> ; |



— n^o : <code>\numero{}</code> ;	— Sts- : <code>\saints{}</code> ;
— n^{os} : <code>\numeros{}</code> ;	— Ste- : <code>\sainte{}</code> ;
— p. ex. : <code>\parexemple{}</code> ;	— Stes- : <code>\saintes{}</code> ;
— P^r : <code>\professeur{}</code> ;	— 1^{er} siècle : <code>\siecle{1}</code> ;
— P^{rs} : <code>\professeurs{}</code> ;	— I - II ^e siècles : <code>\siecles{1}{2}</code> .
— St- : <code>\saint{}</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 (<http://www.ipgp.fr/~moguilyn/LaTeX/typo.pdf>).

D'autres instructions pour normaliser des abréviations sont déjà implémentées dans le code `.tex` et ont été abordées en sous-section 2.1.2 page 9 mais comme il s'agit d'une source de fautes typographiques importante, j'en fais l'inventaire ici :

— œ : <code>\oe{}</code> ;	— 1^{ers} : <code>1\iers{}</code> ;
— Œ : <code>\OE{}</code> ;	— 1^{re} : <code>1\iere{}</code> ;
— æ : <code>\ae{}</code> ;	— 1^{res} : <code>1\ieres{}</code> ;
— Æ : <code>\AE{}</code> ;	— 1^o : <code>\primo{}</code> ;
— 1^{er} : <code>1\ier{}</code> ;	— 2^o : <code>\secundo{}</code> ;
— n^o : <code>\no{}</code> ;	— 3^o : <code>\tertio{}</code> ;
— N^o : <code>\No{}</code> ;	— 4^o : <code>\quarto{}</code> .
— 2^e : <code>2\ieme{}</code> ;	

2.2.5.2 Citation

Comme pour certaines abréviations, les citations font l'objet d'un usage conventionné et leur mise en forme sont déterminées dans le package AOCDF afin – encore et toujours – d'uniformiser toutes les citations contenues dans les documents rédigés avec ce package.

Exemple 2.4: 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 `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 la section 3.5 page 25.



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 grosses 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 pédagogique.

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 les 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 de l'appeler avec un intralien plus loin dans le document, il conviendra de le *labelliser*, c'est-à-dire de lui attribuer un *label* – où *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 abordé dans la section 3.4 page 23.



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 page 6) :

chapitre : chap:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
annexe : ann:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
section : sec:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
sous-section : subsec:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
sous-sous-section : subsubsec:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
figure : fig:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
tableau : tab:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
exemple : ex:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
définition : def:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) ;
formule : form:mot-clé-général_mot-clé-précis(_mot-clé-plus-précis) .

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.

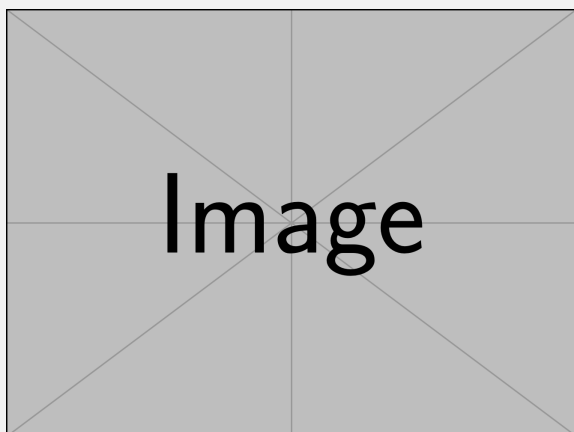


FIG. 3.1 – Image pour le premier exemple d'intralien

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}
```

On remarquera que l'instruction `\ref{<label de l'élément>}` 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 :

Ce texte comporte une figure 3.2 labellisée avec une référence.

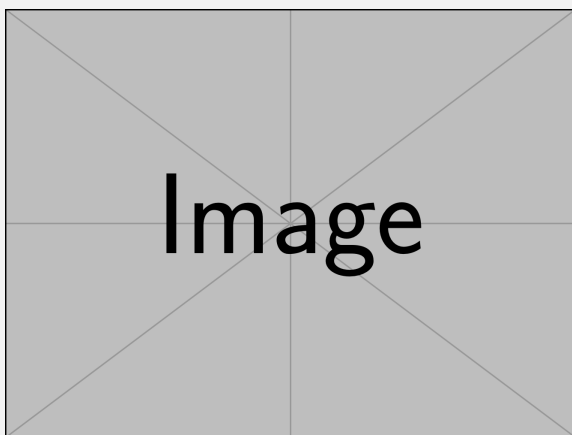


FIG. 3.2 – Image pour le deuxième exemple d'intralien

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.

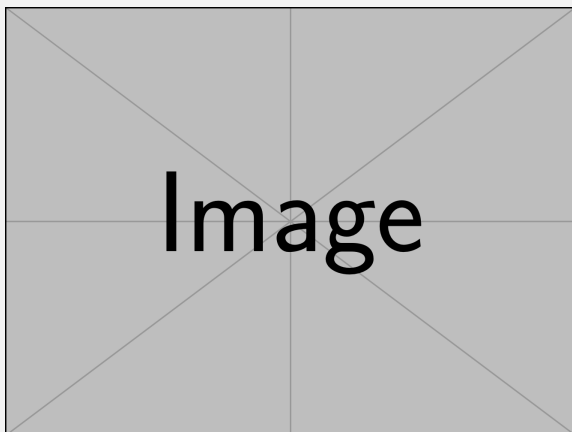


FIG. 3.3 – Image pour le troisième exemple d'intralien

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 page 20 labellisée avec une référence à la mise en forme conditionnée au type de document.

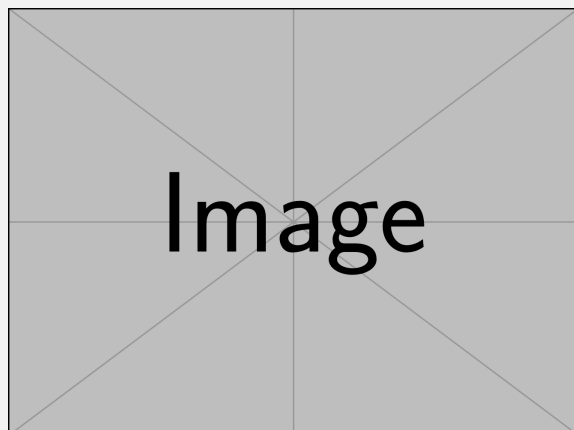


FIG. 3.4 – Image pour le quatrième exemple d'intralien

Code

Ce texte comporte une

```
\superref{fig:image_exemple_intralien_4}
labellisée avec une référence à la mise en
forme conditionnée au type de document.
\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 de couleur noire 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 (<https://github.com/aocdtf-mta/AOCDTF-template>) de ce template.

Code

Ce texte comporte un hyperlien qui renvoie vers le

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

Autre possibilité, la création de liens vers des fichiers locaux ou stockés sur un serveur. Pour cela, on fait également appel à la macro-commande `\Href{run:<chemin d'accès relatif au fichier/fichier.extension>}{<texte cliquable descriptif du lien>}`, en ajoutant donc `run:` devant le chemin d'accès relatif menant au fichier vers lequel on souhaite renvoyer le lecteur. Tout comme avec la commande `\input{file}`, il faut donc identifier le chemin d'accès vers le fichier voulu. Il s'agit d'un chemin d'accès *relatif* à l'emplacement du fichier de code dans lequel il est inclus, et non *absolu*.

Pour cela, on va jouer avec le fichier `INDICATEUR_ARBORESCENCE.tex` situé dans le même dossier que le code `master.tex`.

Après l'avoir préalablement déplacé dans le dossier contenant le fichier vers lequel on



veut effectuer un renvoi, il faut ensuite l'inclure dans le code. Cela est réalisé à l'aide de l'instruction `\input{file}` appelée depuis le menu 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 relatif pour le fichiers de code `INDICATEUR_ARBORESCENCE.tex` qu'il suffira de copier/coller dans l'instruction `\Href{run:<chemin d'accès copié relatif au fichier/fichier.extension>}{<texte cliquable descriptif du lien>}` pour inclure le lien vers le fichier voulu. Il ne faut pas oublier de préciser l'extension du fichier sous peine de rendre le lien inopérant.

Cette fonction ne fonctionne que sur certains lecteurs PDF, dont la référence Adobe Acrobat Reader (<https://get.adobe.com/fr/reader/>). Le lien vers le fichier peut également ne pas aboutir selon le format, toutefois cela fonctionne pour renvoyer vers d'autres documents au format `.pdf`.

Ce texte comporte un hyperlien qui renvoie vers l'indicateur d'arborescence (`run:../../bibliographies/INDICATEUR_ARBORESCENCE.tex`) du dossier contenant les bibliographies de ce template.

Code

Ce texte comporte un hyperlien qui renvoie vers l'`\Href{run:../../bibliographies/INDICATEUR_ARBORESCENCE.tex}{indicateur d'arborescence}` du dossier contenant les bibliographies de ce template.

Il semblerai que pour des raisons de sécurité, Adobe Acrobat Reader (<https://get.adobe.com/fr/reader/>) ne supporte les liens vers les dossiers en tant que tel.

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 page 23. 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 référencés `\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 où 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'hypothèse 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 l'`\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 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 référencés 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 page 24.

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 AOCDF. 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 de-
finition non référencé**

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

Code

```
\begin{definition*}{Exemple
d'environnement definition non
référéncé}{}
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.

Code

```
\begin{formule}{Exemple
d'environnement formule}{exemple_
environnement_formule}
Cet environnement comporte une
formule.
\end{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 for-
mule non référencé**

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

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 L^AT_EX. Son moteur bibliographique présente un principe de fonctionnement un peu spécifique par rapport au reste de la programmation T_EX, qui nécessite une prise en main avant la rédaction de documents. Ce référentiel (<http://bertrandmasson.free.fr/?telechargement/Li4vLi4vZGFOYS9kb2N1bWVudHMvbGF0ZXgvYmliZGF0ZXhtaWNodS5wZGYqZWQyMTRh>) peut être d'une grande aide pour faire face au plus grand nombre de cas de figure.

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 utilisera pour une citation spécifique dans le corps de texte ou en fin de document.

3.5.1 Code bibliographique

Il y a donc un fichier de code `matiere.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 `matiere.bib` sont appelés dans le code de la même manière que l'inclusion de fichiers `code.tex`, mais avec l'instruction `\addbibresource{<matiere.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 `matiere.bib`. Il s'agit d'un chemin d'accès *relatif* à l'emplacement du fichier de code dans lequel il est inclus, et non *absolu*.

Pour cela, on va utiliser un fichier `.tex` 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 depuis le menu `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/matiere.bib>}` pour inclure les fichiers de codes bibliographiques.

3.5.1.2 Structure de codes bibliographiques

Les codes `matiere.bib` présentent 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'utilisent pas exactement la même nomenclature que les codes `.tex`, et il convient de s'inspirer du code `exemple.bib` (`run:../bibliographies/exemple.bib`) pour la rédaction de nouveaux codes bibliographiques ainsi que l'entrée de nouvelles références.

Le code au format `.bib` contient donc des *entrées* bibliographiques qui sont en quelques sortes les « cartes d'identités » des sources utilisées pour la rédaction de documents. Ces entrées présentent trois caractéristiques majeures :

- `@<format>` : elles présentent plusieurs formats selon le type de source utilisée (site internet, thèse, livre dans une collection...) ;
- `@<format>{<clé d'appel>}` : elles sont identifiées par une clé d'appel définie selon des critères précis ;
- `@<format>{<clé d'appel>}{<méta-données de l'entrée>}` : elles contiennent toutes les informations – les *méta-données* – relatives à la source (titre, auteur, année de publication...).



Exemple 3.9: Entrée bibliographique

Le code suivant produit les trois entrées bibliographiques les plus fréquemment utilisées – un livre dans une collection, un site internet et un document pédagogique – pour citer des sources :

Code

```
@incollection{AuteurAnnee,
author = {Auteur},
title = {Title de l'ouvrage},
booktitle = {Collection},
year = {Année},
editor = {Editeur},
publisher = {{Maison d'édition}},
note = {\Href{run:../dossier/ouvrage.pdf}{Nom de fichier de l'ouvrage}},
}

@online{Site:abreviation,
author = {{Auteur}},
title = {Titre de la page},
url = {Lien de la page},
date = {Date de consultation de la page (format 1900-12-01)},
organization = {Organisation},
date = {Date de rédaction de la page (format 1900-12-01)},
}

@report{Organisme:abreviationAnnee,
author = {Auteur},
title = {Titre du cours},
institution = {Institut},
year = {1900},
url = {Lien de la page},
note = {\Href{run:../dossier/cours.pdf}{Nom de fichier du cours}},
}
```

Ces entrées seront classées par l'ordre alphabétique de leur clés d'appel. Il convient d'absolument respecter la structure ainsi que les caractères utilisés pour coder ces entrées, un oubli de , produira une erreur.

On remarquera également que l'on peut insérer un hyperliens vers le fichier de la source si celui-ci existe et est stocké dans le dépôt. Dans la pratique, on préférera l'usage d'une base de donnée des sources utilisées stockées dans un cloud dont on peut extraire des liens que l'on renseignera au champ `url = {Lien de la page}`, . Cela permet aussi d'éviter les erreurs de chemin d'accès, qui peuvent être en conflit si l'arborescence des dossiers d'un dépôt n'est pas respectée à la lettre et qu'une clé d'appel avec un hyperlien vers un fichier local est appelé dans deux documents distincts.

Le *format* de la source est défini par l'instruction appelée par le caractère @. Il en existe une quantité considérable pour caractériser au mieux un maximum de sources diverses et variées. À chaque format d'entrée est associé une nomenclature de la clé d'appel qu'il conviendra de respecter scrupuleusement, pour conserver une cohérence sur l'ensemble des fichiers de code bibliographiques, mais également pour éviter les doublons et faciliter leur appel dans le corps de texte.

La liste suivante donne quelques formats ainsi que les clés d'appel correspondantes, elle ne se veut pour autant pas exhaustive mais couvrira une grande partie des usages :

livre : @book{AuteurAnnee} ;

brochure : @booklet{Organisme:titreenattache} ;

livre dans une collection : @incollection{AuteurAnnee} ;



conférence/colloque : `@InProceedings{AuteurAnnee}` ;
norme : `@Manual{Organisme:Numeronorme-Annee}` ;
autre : `@misc{AuteurAnnee}` ;
site : `@online{Site:abreviation}` ;
cours : `@report{Organisme:abreviationAnnee}` .

Les clés d'appel doivent respecter au mieux la nomenclature proposée dans la liste ci-dessus. Toutefois, si les données de la source ne le permettent pas totalement, on peut appliquer une certaine souplesse, surtout concernant **abreviation**, pouvant être décliné en **titreenattache** si besoin est. L'essentiel, c'est que :

- la clé d'appel permette une identification rapide de la source dans les codes `.bib` et `.tex` ;
- il n'y aie pas de doublons de clés d'appel ;
- la clé d'appel ne comporte pas d'accent ni de caractères spéciaux.

Dans le cas à priori rarissime d'un doublon inévitable, il suffira de numéroter les clés d'appel comme suit `@<format>{<clé d'appel/A>}` et `@<format>{<clé d'appel/B>}` .

Si l'un des rédacteur fait face à un nouveau cas de figure nécessitant un nouveau format d'entrée, il faut l'ajouter – après consultation – dans le code `exemple.bib` (`run:../../bibliographies/exemple.bib`) ainsi que dans la liste ci-dessus, après lui avoir défini une clé d'appel structurée.

Les méta-données qui suivent la clé d'appel d'une entrée sont détaillées non exhaustivement dans le code `exemple.bib` (`run:../../bibliographies/exemple.bib`). Ceux qui souhaiteraient plus de cas de figures et d'explication peuvent consulter ce référentiel (<http://bertrandmasson.free.fr/?telechargement/Li4vLi4vZGF0YS9kb2N1bWVudHMvbGF0ZXgYmliGF0ZXhtaWNodS5wZGYqZWQyMTRh>) très complet et didactique.

3.5.2 Citation des sources

Le but derrière la programmation d'un fichier de code `matiere.bib` est bien de citer les sources utilisées dans le document initial, que ça soit dans le corps du texte ou encore en liste en fin de document.

Exemple 3.10: Citation bibliographique

Pour citer une source, il existe plusieurs instructions selon le format de la citation, dont le bloc sera la clé d'appel de la source souhaitée, définie au préalable dans un fichier de code `matiere.bib`. On privilégiera les formats de citations utilisés dans les exemples suivants, dont le premier est défini par l'instruction `\supercite{<clé d'appel>}` , qui renvoie le lecteur vers la bibliographie avec le numéro de la source en indice :

L'instruction `\supercite{<clé d'appel>}` va citer la source d'un cours³ en produisant un lien en indice vers la bibliographie.

Code

L'instruction
`\mintinline{latex}{\supercite{<clé d'appel>}}`
 va citer la source d'un
 cours `\supercite{AOCDTF:RPT2021}` en produisant
 un lien en indice vers la bibliographie.

La seconde instruction `\footfullcite{<clé d'appel>}` permet d'insérer une citation en indice qui renvoie à la source en bas de page :



L'instruction `\footfullcite{<clé d'appel>}` va citer la source d'un site internet ^a en produisant un lien en indice vers la source en bas de page.

a. BRUNO DOUCHY. *Wiki du template AOCDTF*. Métier des Technologies Associées. 10 fév. 2021. URL : <https://github.com/aocdtf-mta/AOCDTF-template/wiki> (visité le 20/04/2021).

Code

```
L'instruction
\mintinline{latex}{\footfullcite{<clé
d'appel>}} va citer la source d'un site
internet
\footfullcite{Github:AOCDTFtemplatewiki} en
produisant un lien en indice vers la source en
bas de page.
```

La troisième instruction `\pprbcite{<clé d'appel>}` permet d'insérer le titre d'une source et son auteur, permettant d'indiquer la source d'une citation par exemple :

Il est cité dans « Titre d'un livre de référence issu d'une collection » de DOUCHY : « une citation de moins de quarante mots qui s'intègre dans la phrase », alors que le texte initial reprend après.

Code

```
Il est cité dans \pprbcite{Douchy2021} :
\say{une citation de moins de quarante mots
qui s'intègre dans la phrase}, alors que le
texte initial reprend après.
```

La dernière manière privilégiée de citer ses sources est de directement les appeler dans la bibliographie en fin de document avec l'instruction `\nocite{<clé d'appel A, clé d'appel B, clé d'appel X>}`. Cette instruction contient les clés d'appel non citées dans le corps du document mais dont les sources sont tout de même utilisées tout au long de la rédaction. Elle est à renseigner dans le code `master.tex`, suivie de l'instruction `\printbibliography` pour produire la bibliographie à l'endroit du code où cette instruction est appelée, généralement après la conclusion et avant l'index.

Pour plus de précisions, d'autres formats de citations sont détaillés dans le référentiel (<http://bertrandmasson.free.fr/?telechargement/Li4vLi4vZGF0YS9kb2N1bWVudHMvbGF0ZXgYmliZGF0ZXhtaWNodS5wZGYqZWQyMTRh>) suivant.

3.5.3 Compilation

Une fois le fichier de code `cours.bib` remplis d'entrées et les clés d'appel correspondantes dans les fichiers de code `.tex` correctement renseignées, il est nécessaire de compiler avec le moteur *Compilation rapide*, dont le paramétrage est renseigné à cette page (<https://github.com/aocdtf-mta/AOCDTF-template/wiki/Configuration-de-l'outil-bibliographique-Biber>) du wiki.

Avant cela, il faut configurer le Texmaker pour qu'il fonctionne avec le moteur bibliographique *Biber* dans les paramètres de Texmaker *appelés depuis le menu Texmaker > Préférences > Commandes > Bib(la)tex > biber %*.



4.1 Introduction

À la base de $\text{T}_{\text{E}}\text{X}$ et $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ était l'intention de créer un outil destiné à faciliter la rédaction de publications scientifiques. C'est donc tout naturellement que $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ intègre le meilleur outil pour rédiger des équations mathématiques complexes ou encore des formules de physiques. D'ailleurs, beaucoup ont déjà côtoyé le langage $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ sans le savoir, car cet outil de rédaction est intégré dans le logiciel de traitement de texte Microsoft Word, tant il reste pertinent depuis une quarantaine d'année maintenant.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ présente un *mode mathématique* caractérisé par une syntaxe légèrement différente du reste, mais plutôt aisée à prendre en main. En outre, toutes les joyeusetés de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ servent également l'écriture scientifique (référencement, mise en forme...).

4.2 Notation mathématique

4.2.1 Environnement mathématique

Pour les documents à destination de l'AOCDF, l'environnement de rédaction mathématique `displaymath` est appelé de deux manières :

« en ligne » dans le texte : les caractères `\(<notation mathématique>\)` intègrent des notations mathématiques directement dans le texte.

dans un environnement spécifique : l'environnement `align*` permet l'intégration de lot d'équations alignées sur un symbole défini.

Dans la pratique, on fait appel à l'environnement `align*` non numéroté plutôt que l'environnement `align` car, dans la très grande majorité des cas, ces environnements sont intégrés dans des environnements `formule` ou `exemple`. Toutefois, devant un cas de figure spécifique, on peut faire appel à l'environnement numéroté `align`, auquel on peut y ajouter un label pour l'intégrer dans des intraliens.

L'écriture mathématique modifie légèrement la mise en forme du texte :

— suppression de tous les espaces qui doivent donc être appelés à l'aide des instructions suivantes :

espace normal : `\` ;

espace grande : `\;` ;

espace fine : `\,` ;

espace moyenne : `\:` ;

espace fine négative : `\!` .

— les fontes sont automatiquement gérées et respectent les normes de notations scientifiques en vigueur.

Exemple 4.1: Notation mathématique

L'environnement `align*` permet l'insertion d'équations alignées sur un symbole défini – généralement le `=` – par le caractère `&` le précédant. Chaque ligne de l'équation doit présenter un saut de ligne `\\` :



Dans ce texte est inséré « en ligne » une notation mathématique $x = 2^2$, ainsi qu'une équation alignée sur le symbole $=$:

$$\begin{aligned} x &= 2^2 \times 5 \\ &= 4 \times 5 \\ &= 20 \end{aligned}$$

Dans ce texte est inséré `\og en ligne \fg{}` une notation mathématique `\(x=2^{2}\)`, ainsi qu'une équation alignée sur le symbole `\(=\)` :

```
\begin{align*}
x   &= 2^{2} \cdot 5 \\
&= 4 \cdot 5 \\
&= 20
\end{align*}
```

4.2.2 Syntaxe et instructions

La syntaxe utilisée pour écrire dans l'environnement `displaymath` est un peu différente, la liste suivante donne les grands principes de rédaction :

instruction : utilisation *au maximum* d'instructions pour l'insertion des symboles¹ ;

exposant : utilisation de l'instruction `^{<terme en exposant>}` ;

indice : utilisation de l'instruction `_{<terme en indice>}` :

variable ou grandeur physique : indice en italique ;

abréviation : indice en normal.

opération : utilisation d'instructions spécifiques et non du clavier pour les divisions et multiplications.

La liste suivante, non exhaustive, détaille les principales instructions des signes mathématiques utilisés dans l'environnement `displaymath`, qui couvriront la majorité des usages :

- | | |
|---|---|
| — $a = b$: <code>a = b</code> ; | — $3a$: <code>3 a</code> (algèbre avec chiffre) ; |
| — $a \neq b$: <code>a \neq b</code> ; | — a/b : <code>a/b</code> ; |
| — $a \triangleq b$: <code>a \triangleq b</code> ; | — $\frac{a}{b}$: <code>\frac{a}{b}</code> ; |
| — $a \simeq b$: <code>a \simeq b</code> ; | — $\frac{a}{b}$: <code>\dfrac{a}{b}</code> ; |
| — $a < b$: <code>a < b</code> ; | — $\sum_{i=1}^n a_i$: <code>{\displaystyle\sum_{i=1}^n a_{i}}</code> ; |
| — $a > b$: <code>a > b</code> ; | — $\prod_{i=1}^n a_i$: <code>{\displaystyle\prod_{i=1}^n a_{i}}</code> ; |
| — $a \leq b$: <code>a \leq b</code> ; | — $n!$: <code>n!</code> ; |
| — $a \geq b$: <code>a \geq b</code> ; | — a^n : <code>a^{n}</code> ; |
| — $a \ll b$: <code>a \ll b</code> ; | — \sqrt{a} : <code>\sqrt{a}</code> ; |
| — $a \gg b$: <code>a \gg b</code> ; | — $\sqrt[n]{a}$: <code>\sqrt[n]{a}</code> ; |
| — ∞ : <code>\infty</code> ; | — $a^{1/n}$: <code>a^{1/n}</code> ; |
| — $a \pm b$: <code>a \pm b</code> ; | — $ a $: <code>\vert a \vert</code> ; |
| — $x \in b$: <code>x \in b</code> ; | — \overrightarrow{AB} : <code>\overrightarrow{AB}</code> . |
| — $x \notin b$: <code>x \notin b</code> ; | |
| — $a + b$: <code>a + b</code> ; | |
| — $a - b$: <code>a - b</code> ; | |
| — $a \times b$: <code>a \times b</code> (arithmétique) ; | |
| — $a \cdot b$: <code>a \cdot b</code> (algèbre littéral) ; | |

1. L^AT_EX peut prendre en charge les caractères de symboles « clé en main » qui peuvent être copier/coller depuis d'autres documents mais ceux-ci présentent des propriétés différentes des caractères mathématiques appelés avec l'instruction correspondante. Cela peut dérégler la mise en forme des écritures mathématiques.



La liste suivante, non exhaustive, détaille les principales instructions des symboles mathématiques utilisés dans l'environnement `displaymath`, qui couvriront la majorité des usages :

— $f : f$;	— $\mathbb{Q} : \texttt{\textbackslash mathbb{Q}}$;
— $f(x) : f(x)$;	— $\mathbb{R} : \texttt{\textbackslash mathbb{R}}$;
— $[f(x)]_b^a : \texttt{\textbackslash left[f(x) \textbackslash right] ^{a} _{b}}$;	— $\mathbb{C} : \texttt{\textbackslash mathbb{C}}$;
— $f(x) _b^a : f(x) \textbackslash ^{a} _{b}$;	— $\mathbb{P} : \texttt{\textbackslash mathbb{P}}$;
— $\lim_{x \rightarrow a} f(x) : \texttt{\textbackslash lim\limits_{x \rightarrow a}}$;	— $\cos x : \texttt{\textbackslash cos x}$;
— $f' : f\texttt{\textbackslash prime}$;	— $\sin x : \texttt{\textbackslash sin x}$;
— $f^{(k)}(x) : f^{\{k\}}(x)$;	— $\tan x : \texttt{\textbackslash tan x}$;
— $\Delta f : \texttt{\textbackslash Delta f}$;	— $\cot x : \texttt{\textbackslash cot x}$;
— $\frac{df}{dx} : \texttt{\textbackslash frac{df}{dx}}$;	— $\arccos x : \texttt{\textbackslash arccos x}$;
— $\frac{\partial f}{\partial x} : \texttt{\textbackslash frac{\partial f}{\partial x}}$;	— $\arcsin x : \texttt{\textbackslash arcsin x}$;
— $\frac{\delta f}{\delta x} : \texttt{\textbackslash frac{\delta f}{\delta x}}$;	— $\arctan x : \texttt{\textbackslash arctan x}$;
— $\int_a^b f(x)dx : \texttt{\textbackslash int_{a}^b f(x)dx}$;	— $\exp x : \texttt{\textbackslash exp x}$;
— $\bar{f} : \texttt{\textbackslash bar{f}}$;	— $\ln x : \texttt{\textbackslash ln x}$;
— $\mathbb{N} : \texttt{\textbackslash mathbb{N}}$;	— $i, j : \texttt{\textbackslash mathrm{i}}, \texttt{\textbackslash mathrm{j}}$;
— $\mathbb{Z} : \texttt{\textbackslash mathbb{Z}}$;	— $\arg : \texttt{\textbackslash arg}$.

La notation mathématique fait également souvent appel à l'alphabet grec, dont les instructions sont listées ci-dessous :

Caractère romain		Caractère italique	
A : A	$\alpha : \texttt{\textbackslash alphaup}$	$A : \texttt{\textbackslash textit {A}}$	$\alpha : \texttt{\textbackslash alpha}$
B : B	$\beta : \texttt{\textbackslash betaup}$	$B : \texttt{\textbackslash textit {B}}$	$\beta : \texttt{\textbackslash beta}$
$\Gamma : \texttt{\textbackslash Gamma}$	$\gamma : \texttt{\textbackslash gammaup}$	$\Gamma : \texttt{\textbackslash mathit {\Gamma}}$	$\gamma : \texttt{\textbackslash gamma}$
$\Delta : \texttt{\textbackslash Delta}$	$\delta : \texttt{\textbackslash deltaup}$	$\Delta : \texttt{\textbackslash mathit {\Delta}}$	$\delta : \texttt{\textbackslash delta}$
E : E	$\epsilon : \texttt{\textbackslash epsilonup}$	$E : \texttt{\textbackslash textit {E}}$	$\epsilon : \texttt{\textbackslash epsilon}$
	$\varepsilon : \texttt{\textbackslash varepsilonup}$		$\varepsilon : \texttt{\textbackslash varepsilon}$
Z : Z	$\zeta : \texttt{\textbackslash zetaup}$	$Z : \texttt{\textbackslash textit {Z}}$	$\zeta : \texttt{\textbackslash zeta}$
H : H	$\eta : \texttt{\textbackslash etaup}$	$H : \texttt{\textbackslash textit {H}}$	$\eta : \texttt{\textbackslash eta}$
$\Theta : \texttt{\textbackslash Theta}$	$\theta : \texttt{\textbackslash thetaup}$	$\Theta : \texttt{\textbackslash mathit {\Theta}}$	$\theta : \texttt{\textbackslash mathit {\theta}}$
I : I	$\iota : \texttt{\textbackslash iotaup}$	$I : \texttt{\textbackslash textit {I}}$	$\iota : \texttt{\textbackslash iota}$
K : K	$\kappa : \texttt{\textbackslash kappaup}$	$K : \texttt{\textbackslash textit {K}}$	$\kappa : \texttt{\textbackslash kappa}$
	$\varkappa : \texttt{\textbackslash varkappaup}$		$\varkappa : \texttt{\textbackslash varkappa}$
$\Lambda : \texttt{\textbackslash Lambda}$	$\lambda : \texttt{\textbackslash lambdaup}$	$\Lambda : \texttt{\textbackslash mathit {\Lambda}}$	$\lambda : \texttt{\textbackslash lambda}$
M : M	$\mu : \texttt{\textbackslash muup}$	$M : \texttt{\textbackslash textit {M}}$	$\mu : \texttt{\textbackslash mu}$
N : N	$\nu : \texttt{\textbackslash nuup}$	$N : \texttt{\textbackslash textit {N}}$	$\nu : \texttt{\textbackslash nu}$
$\Xi : \texttt{\textbackslash Xi}$	$\xi : \texttt{\textbackslash xiup}$	$\Xi : \texttt{\textbackslash mathit {\Xi}}$	$\xi : \texttt{\textbackslash xi}$
O : O	$\omicron : \texttt{\textbackslash omicronup}$	$O : \texttt{\textbackslash textit {O}}$	$\omicron : \texttt{\textbackslash textit {o}}$
$\Pi : \texttt{\textbackslash Pi}$	$\pi : \texttt{\textbackslash piup}$	$\Pi : \texttt{\textbackslash mathit {\Pi}}$	$\pi : \texttt{\textbackslash pi}$
	$\varpi : \texttt{\textbackslash varpiup}$		$\varpi : \texttt{\textbackslash varpi}$
P : P	$\rho : \texttt{\textbackslash rhoup}$	$P : \texttt{\textbackslash textit {P}}$	$\rho : \texttt{\textbackslash rho}$
	$\varrho : \texttt{\textbackslash varrhoup}$		$\varrho : \texttt{\textbackslash varrho}$
$\Sigma : \texttt{\textbackslash Sigma}$	$\sigma : \texttt{\textbackslash sigmaup}$	$\Sigma : \texttt{\textbackslash mathit {\Sigma}}$	$\sigma : \texttt{\textbackslash sigma}$
T : T	$\tau : \texttt{\textbackslash tauup}$	$T : \texttt{\textbackslash textit {T}}$	$\tau : \texttt{\textbackslash tau}$
Y : Y	$\upsilon : \texttt{\textbackslash upsilonup}$	$Y : \texttt{\textbackslash textit {Y}}$	$\upsilon : \texttt{\textbackslash upsilon}$

Page suivante



Caractère romain		Caractère italique	
Φ : <code>\Phi</code>	ϕ : <code>\phi</code>	Φ : <code>\mathit{\Phi}</code>	ϕ : <code>\phi</code>
X : <code>x</code>	χ : <code>\chi</code>	X : <code>\textit{X}</code>	χ : <code>\chi</code>
Ψ : <code>\Psi</code>	ψ : <code>\psi</code>	Ψ : <code>\mathit{\Psi}</code>	ψ : <code>\psi</code>
Ω : <code>\Omega</code>	ω : <code>\omega</code>	Ω : <code>\mathit{\Omega}</code>	Ω : <code>\Omega</code>

TAB. 4.1 – Alphabet grec

4.3 Grandeurs et unité de mesure

L^AT_EX offre également la possibilité de mettre en forme des grandeurs physiques et unités de mesures à l'aide d'instructions spécifiques qu'elles soient intégrées au texte ou dans un environnement mathématique.

Le package AOCDFTF contient également des macro-commandes automatisant la description de formules avec un rendu clair et limpide.

4.3.1 Généralités

4.3.1.1 Différenciation

Avant de détailler les codes pour écrire des grandeurs physique et unités de mesures, il convient de bien identifier la terminologie les concernant.

Définition 4.1: Unité de mesure

Étalon de mesure nécessaire pour la mesure d'une grandeur physique dont le fondement est l'exacte reproductibilité expérimentale de l'étalon.

Définition 4.2: Grandeur physique

Toute propriété des sciences de la nature qui peut être mesurée ou calculée et dont les différentes valeurs s'expriment à l'aide d'un nombre réel ou complexe. Une grandeur physique peut s'exprimer sans unité de mesure, ce sont des *grandeurs sans dimension*. L'inverse n'est pas vraie, toute unité de mesure est associée une grandeur physique.

Définition 4.3: Dimension

Expression de la dépendance d'une grandeur par rapport aux grandeurs de base d'un système de grandeurs sous la forme d'un produit de puissance de facteurs correspondant aux grandeurs de base, en omettant tout facteur numérique.

4.3.1.2 Principes de rédaction

Pour obtenir des notations de grandeurs physiques et unités de mesures respectant au mieux les normes en vigueur et pour uniformiser l'ensemble des documents produits à destination de l'AOCDFTF, il convient de respecter quelques principes de rédaction :

symboles des grandeurs les symboles usuels des grandeurs physique prennent généralement la forme d'une seule lettre (alphabet grec ou latin), toujours en italique, et peuvent être précisés par des indices.

indice un indice permet de différencier des grandeurs présentant le même symbole usuel ou, pour une même grandeur, différentes applications de celle-ci.



- symbole d'une grandeur physique ou d'une variable mathématique ;
- mots ou nombres fixes.

symboles des unités Les symboles des unités prennent généralement la forme d'une seule lettre (alphabet grec ou latin), toujours en caractère droit, ce qui permet de les différencier des symboles des grandeurs.

Une unité composée d'une multiplication de deux unités ou plus peut être indiquée de deux manières :

$$N \cdot m$$

$$Nm$$

Il convient de faire attention lorsque le symbole d'une unité est le même que celui d'un préfixe.

4.3.1.3 Terminologie

Afin d'être précis dans la terminologie scientifique, voici quelques précisions sur des termes qui se ressemblent et qui peuvent être source d'imprécisions :

Coefficient dans une équation type $A = k \cdot B$, k est le coefficient/facteur et A est une grandeur proportionnelle à B . Usage du terme *coefficient* (ou *module*) lorsque les grandeurs A et B présentent des *dimensions* différentes.

Facteur dans une équation type $A = k \cdot B$, k est le coefficient/facteur et A est une grandeur proportionnelle à B . Usage du terme *facteur* lorsque les grandeurs A et B sont de même *dimension*.

Paramètre combinaison de grandeurs qui apparaissent sous une telle forme dans les équations, pouvant être considérée comme constituant de nouvelles grandeurs.

Nombre combinaison de grandeurs sans dimension.

Rapport quotient sans dimension de deux grandeurs.

Constante grandeur qui présente la même valeur en toutes circonstances.

Massique adjectif apposé à une grandeur caractérisant le quotient de cette grandeur par la masse.

Volumique adjectif apposé à une grandeur caractérisant le quotient de cette grandeur par le volume.

Surfacique adjectif apposé à une grandeur caractérisant le quotient de cette grandeur par l'aire.

Densité adjectif apposé à une grandeur exprimant un flux ou un courant, qui caractérise le quotient de cette grandeur par l'aire.

Linéique adjectif apposé à une grandeur caractérisant le quotient de cette grandeur par la longueur.

Molaire adjectif apposé à une grandeur caractérisant le quotient de cette grandeur par la quantité de matière.

Concentration adjectif apposé à une grandeur, spécifiquement dans le cas d'un mélange, caractérisant le quotient de cette grandeur par le volume total.

4.4 Environnement scientifique

Les unités du Système International (SI) sont donc *systématiquement* définies par leur instruction correspondante, issues du package SIunitx (<http://mirrors.ibiblio.org/CTAN/macros/latex/contrib/siunitx/siunitx.pdf>).

Exemple 4.2: Notation scientifique

Pour rédiger des notations scientifiques – incluse ou non dans un environnement `displaymath` – on fait appel aux diverses instructions définissant précisément la mise en forme de symboles,



de listes ou encore de nombre. L'instruction `\SI{<terme>}{<préfixe et unité>}` sera la plus utilisée tout au long de la rédaction :

Dans ce texte est abordé la vente d'un appartement d'une surface de 45m^2 , ainsi que la vitesse la lumière, qui est de $3 \times 10^5 \text{km s}^{-1}$.

Code

```
Dans ce texte est abordé la vente d'un
appartement d'une surface de
\SI{45}{\square\meter}, ainsi que la vitesse
la lumière, qui est de
\SI{3e5}{\kilo\meter\per\second}.
```

Les variables de l'instruction `\SI{<terme>}{<préfixe et unité>}` sont donc les instructions pour les différentes unités et préfixes sont à retrouver dans la ?? page ??, ainsi qu'en ?? page ??. Dans le premier argument, si l'on souhaite produire une notation scientifique des nombres, il faut rédiger `\SI{<significande>e<exposant>}{<préfixe et unité>}`.

Dans le deuxième argument, l'écriture des unités est assez littérale, avec l'instruction `\per` pour mettre une unité au dénominateur et les instructions `\square` et `\cubic` pour mettre des unités respectivement au carré et au cube.

On peut n'afficher que l'unité d'une grandeur avec l'instruction `\si{<préfixe et unité>}` ou qu'un nombre avec l'instruction `\num{<terme>}`, pratique pour la notation scientifique.

Il existe également la possibilité de produire des listes de nombres sans unité avec l'instruction `\numlist{<terme1;terme2;terme3>}` ou avec unité avec l'instruction `\SIlist{<terme1;terme2;terme3>}{<préfixe et unité>}`, avec un séparateur de terme défini avec le caractère ; . Ou encore des plages de nombres avec les instructions `\numrange{<terme1>}{<terme2>}` et `\SIrange{<terme1>}{<terme2>}{<préfixe et unité>}`.

Dans ce texte est abordé les premiers paliers des sections de câbles utilisé dans les installations électriques domestiques, mesurant respectivement $1,5\text{mm}^2$, $2,5\text{mm}^2$, 4mm^2 et 6mm^2 . Ces sections permettent de faire transiter un ampérage de 0 à 32A, toujours selon les normes en vigueur.

Code

```
Dans ce texte est abordé les premiers paliers
des sections de câbles utilisé dans les
installations électriques domestiques,
mesurant respectivement
\SIlist{1,5;2,5;4;6}{\square\milli\meter}.
Ces sections permettent de faire transiter un
ampérage de \SIrange{0}{32}{\ampere},
toujours selon les normes en vigueur.
```

4.4.1 Formules

Pour rédiger des formules scientifiques, il convient de les inclure dans l'environnement `formule` abordées dans l'exemple 3.8 page 25 .

Les formules scientifiques bénéficient de deux nouveaux environnements permettant de structurer en tableau le détails des variables, selon que cela soit numérique avec `numvariables` ou textuel avec `textvariables`. Cela permet de conserver une unité graphique parmi tous les documents de l'AOCDF.

Exemple 4.3: Formule avec détails

L'environnement `textvariables` produit un tableau à cinq colonnes permettant de détailler chaque variable selon le contenu type suivant :

```
<symbole de la grandeur> & <grandeur> & <unité de la grandeur> & <symbole de l'unité (ins-
truction)> & <description et rôle de la grandeur> \\
```



Par exemple, la formule de la probabilité d'électrocution est détaillée de la manière suivante :

Code

```
\begin{formule}{Probabilité d'électrocution}{probabilite_electrocution}
\begin{align*}
I &= \frac{116}{\sqrt{t}}
\end{align*}
\begin{textvariables}
I & courant électrique & milliampère & \milli\ampere & Courant traversant le corps
\\
t & durée & seconde & \second & Durée du choc électrique d'une durée \((8\milli\second <
t \leq 5\second)\) \\
116 & constante & & / & Constante empirique déterminée statistiquement\\
\end{textvariables}
\end{formule}
```

Cela produira :

Formule 4.1: Probabilité d'électrocution

$$I = \frac{116}{\sqrt{t}}$$

Avec :

Grandeur dans l'ISQ	Unité SI de mesure	Description
I : courant électrique	milliampère (mA)	Courant traversant le corps
t : durée	seconde (s)	Durée du choc électrique d'une durée $8\text{ms} < t \leq 5\text{s}$
116 : constante	(/)	Constante empirique déterminée statistiquement

L'environnement `numvariables` produit un tableau à six colonnes permettant de détailler chaque variable selon le contenu type suivant :

```
<symbole de la grandeur> & <grandeur> & <unité de la grandeur> & <symbole de l'unité (ins-
truction)> & <symbole de l'unité (instruction)> & <description numérique> \\
```

Par exemple, la formule de la valeur expérimentale de l'eVest détaillée de la manière suivante :

Code

```
\begin{formule}{Valeur expérimentale de l'\electronvolt}{valeur_experimentale_electronvolt}
\begin{align*}
U &= \frac{W}{Q} \\
\electronvolt &= \sqrt{\frac{2h\alpha}{\mu\cleft}} \frac{W}{Q} \\
&= \SI{1,602176634e-19}{\joule}
\end{align*}

\begin{numvariables}
U & différence de potentiel & volt & \volt & \si{kilogram\square\meter\per\cubic\second\per\ampere} \\
W & énergie & joule & \joule & \joule & \si{kg.m^2/s^2} \\
Q & charge électrique & coulomb & \coulomb & \coulomb & \si{\ampere\second} \\
\electronvolt & électron-volt & joule & \joule & \electronvolt & \SI{1,602176634e-19}{\joule} \\
h & constante de Planck & joule seconde & \si{\joule\second} & h & \\
\alpha & constante de structure fine & sans dimension & & \alpha & \num{7,2973525564e-3}
\end{numvariables}
```




```
\mu & perméabilité magnétique du vide & henry par mètre & \si{\henry\per\meter} & \mu &
\SI{4\pi e-7}{\henry\per\meter} \\
\clight & vitesse de la lumière dans le vide & mètre par seconde & \si{\meter\per\second} &
\clight & \SI{2,99792458e8}{\meter\per\second}
\end{numvariables}
\end{formule}
```

Cela produira ^a :

Formule 4.2: Valeur expérimentale de l'eV

$$U = \frac{W}{Q}$$

$$\text{eV} = \sqrt{\frac{2h\alpha}{\mu c_0}} \frac{W}{Q}$$

$$= 1,602\,176\,634 \times 10^{-19} \text{ J}$$

Avec :

Grandeur dans l'ISQ	Unité SI de mesure		Valeur
U : différence de potentiel	volt	(V)	$V = \text{kg m}^2 \text{s}^{-3} \text{A}^{-1}$
W : énergie	joule	(J)	$J = \text{kg m}^2/\text{s}^2$
Q : charge électrique	coulomb	(C)	$C = \text{A s}$
eV : électron-volt	joule	(J)	$\text{eV} = 1,602\,176\,634 \times 10^{-19} \text{ J}$
h : constante de Planck	joule se- conde	(Js)	$h = 6,626\,070\,15 \times 10^{-34} \text{ J s}$
α : constante de structure fine	sans dimen- sion	()	$\alpha = 7,297\,352\,556\,4 \times 10^{-3}$
μ : perméabilité magnétique du vide	henry par mètre	(H m ⁻¹)	$\mu = 4\pi \times 10^{-7} \text{ H m}^{-1}$
c_0 : vitesse de la lumière dans le vide	mètre par se- conde	(m s ⁻¹)	$c_0 = 2,997\,924\,58 \times 10^8 \text{ m s}^{-1}$

^a. Les différentes colonnes sont mieux agencées quand l'environnement `formule` n'est pas inséré dans un environnement `exemple`.



5.1 Introduction

Avec \LaTeX , on peut produire des tableaux respectant une charte graphique favorisant la clarté des informations et contenant des fonctions d'automatisation et d'agencement permettant un rendu final irréprochable et « adaptatif ».

De prime abord, le principe de rédaction des tableaux en code peut paraître fastidieux et c'est pourquoi des macro-commandes ont été définies pour faciliter leur programmation. Néanmoins, une fois les instructions prise en main, il sera compliqué d'envisager rédiger des tableaux autrement que de la sorte.

5.2 Préceptes typographiques

Quand il s'agit de rédiger des tableaux, il existe des règles de « bonne pratique » que l'on retrouve appliquées dans la totalité des ouvrages comportant des tableaux de qualité :

- il ne doit y avoir *aucune* séparation verticale dans les tableaux, remplacée par des espaces.
- les blocs de texte d'une colonne ont une disposition *justifiée*.
- les éléments textuels seuls sont centrés horizontalement et verticalement (ou alignés verticalement en haut s'ils sont à côté d'une colonne justifiée).
- les nombres et grandeurs sont horizontalement et verticalement centrés (ou verticalement alignés en haut s'ils sont à côté d'une colonne justifiée).
- les devises sont horizontalement alignées à droite et verticalement centrées (ou verticalement alignées en haut si elles sont à côté d'une colonne justifiée).
- les unités sont insérées entre parenthèses dans le titre de la colonne.
- les titres de colonnes sont en gras et centrés horizontalement et verticalement.
- les titres de colonnes sont *toujours* au singulier.

Exemple 5.1: Tableau simple

Selon les critères listé ci-dessous, un tableau basique est produit avec l'environnement `\begin{tabular}[<position>]{<nombre et type de colonnes>}` et doit donc s'apparenter au code suivant :

Code

```
\begin{table}[H]
\caption{Un premier tableau simple avec \texttt{tabular}}
\begin{tabular}[t]{c p{5cm}}
\toprule
\thead{Titre colonne nombre (unité)} & \thead{Titre colonne texte} \\
\midrule
1 & Un premier paragraphe de \SI{5}{\centi\meter} de large \\
2 & Un deuxième paragraphe de \SI{5}{\centi\meter} de large \\
3 & Un troisième paragraphe de \SI{5}{\centi\meter} de large \\
\bottomrule
\end{tabular}
\end{table}
```



Cela produira :

Titre colonne nombre (unité)	Titre colonne texte
1	Un premier paragraphe de 5cm de large
2	Un deuxième paragraphe de 5cm de large
3	Un troisième paragraphe de 5cm de large

TAB. 5.1 – Un premier tableau simple avec `tabular`

5.3 Structure de base

Si l'on se réfère au code de l'exemple 5.1 page 39, on fait donc appel à l'environnement `\begin{tabular}` pour produire un tableau sous sa forme la plus simple. Celui-ci présente une largeur conditionnée principalement par son contenu, c'est-à-dire que si le contenu des cellules est trop large, le tableau va déborder de l'espace de rédaction.

Si l'on décompose son code, on commence par considérer le tableau comme un élément *flottant* en l'insérant dans l'environnement `\begin{table}`. Dans cet environnement flottant est insérée la légende avec l'instruction `\caption{<légende>}`.

Ensuite est déclaré l'environnement `\begin{tabular}` qui produira le tableau. Cet environnement ne propose que le nombre et type de colonnes comme argument obligatoire, selon l'instruction suivante `\begin{tabular}[<position>]{<nombre et type de colonnes>}`. De base, il existe six types de colonnes¹ :

colonne alignée à gauche (left) : `l` ;

colonne alignée au centre (center) : `c` ;

colonne alignée à droite (right) : `r` ;

paragraphe aligné en haut avec largeur spécifiée (paragraph) : `p{<largeur>}` ;

paragraphe aligné au milieu avec largeur spécifiée (middle) : `m{<largeur>}` ;

paragraphe aligné en bas avec largeur spécifiée (bottom) : `b{<largeur>}` .

La largeur à préciser dans les trois derniers argument est absolue, avec les unités détaillées dans la section 1.1 page 1.

Ces variables déterminent donc le *format* du contenu des cellules (texte, liste, grandeur physique...) ainsi que le *nombre* de colonnes.

Selon l'exemple 5.1 page 39, le bloc `{c p{5cm}}` définit donc un tableau à deux colonnes, la première `c` étant centrée horizontalement et alignée verticalement en haut et la seconde `p{5cm}` contenant des paragraphes de 5cm de large.

L'argument facultatif [`<position>`] aligne verticalement le tableau par rapport à d'autres éléments du texte, selon les variables suivantes :

haut de la page (top) : `t` ;

milieu de la page (middle) : `m` ;

bas de la page (bottom) : `b` .

1. D'autres types de colonnes ont été définis pour d'autres types de tableaux en ?? page ??



À l'intérieur de l'environnement `\begin{tabular}` est donc inséré le contenu du tableau, encadré par les instructions suivantes :

- `\toprule` : ligne épaisse pour encadrer le haut du tableau, sans retour à la ligne `\\` ;
- `\thead{<titre de la colonne>}` : insertion du titres des colonnes, avec un passage à la colonne suivante appelé avec le caractère `&` et un saut de ligne `\\` en fin de ligne ;
- `\midrule` : ligne fine pour séparer la ligne des titres de colonne du restant du tableau, sans retour à la ligne `\\` ;
- `<cellulle 1> & <cellule 2>` : insertion du contenu des cellules, avec un passage à la colonne suivante appelé avec le caractère `&` et un saut de ligne `\\` en fin de ligne ;
- `\bottomrule` : ligne épaisse pour encadrer le bas du tableau, sans retour à la ligne `\\` .

Lorsqu'on code un tableau, il est important dans la mesure du possible de structurer le code à l'aide de tabulations, pour mettre en évidence les colonnes et s'y retrouver durant la programmation.

5.4 Tableau avec largeur relative

Dans la pratique, les tableaux produits avec l'environnement `\begin{tabular}` sont relativement rares et petits. Les rédacteurs feront plus souvent face à des cas de figures où les tableaux ont une largeur occupant tout l'espace de rédaction.

Pour plus de cohésion graphique, il convient donc de corréliser automatiquement la largeur du tableau à la largeur des colonnes. Cela est rendu possible avec l'environnement `\begin{tabularx}`, qui introduit un nouveau format de colonne `X` à la largeur adaptative selon l'espace restant. Ce type de tableau reste cependant relativement concis et ne peut pas être réparti harmonieusement sur plus d'une page.

Exemple 5.2: Tableau à la largeur relative

Un tableau à la largeur relative est produit avec l'environnement `\begin{tabularx}{<largeur du tableau>[<position>]{<type de colonnes>}`. On remarquera le nouvel argument `{<largeur du tableau>}` , ainsi qu'un nouveau format de colonne `X` :

Code

```
\begin{table}[H]
\caption{Un deuxième tableau avec \texttt{tabularx}}
\begin{tabularx}{\linewidth}[t]{c X r}
\toprule
\thead{Titre colonne nombre (unité)} & \thead{Titre colonne texte} & \thead{Prix
(\EUR{}} \\
\midrule
1 & Un premier texte à l'agencement format \texttt{paragraph} et à la largeur
automatiquement ajustée à l'espace restant sur la largeur spécifiée
\Verb{\linewidth}} & 15 \\
2 & Un deuxième texte à l'agencement format \texttt{paragraph} et à la largeur
automatiquement ajustée à l'espace restant sur la largeur spécifiée \Verb{\linewidth}} &
30 \\
3 & Un troisième texte à l'agencement format \texttt{paragraph} et à la largeur
automatiquement ajustée à l'espace restant sur la largeur spécifiée \Verb{\linewidth}} &
45 \\
\bottomrule
\end{tabularx}
\end{table}
```

Cela produira :



Titre colonne nombre	Titre colonne texte	Devise (€)
1	Un premier texte à l'agencement format paragraph et à la largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée	15
2	Un deuxième texte à l'agencement format paragraph et à la largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée	30
3	Un troisième texte à l'agencement format paragraph et à la largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée	45

TAB. 5.2 – Un deuxième tableau avec `tabularx`

La colonne **X** voit donc sa largeur s'adapter en fonction de son contenu et des celui des autres cellules pour mesurer précisément la variable `\linewidth` renseignée.

5.4.1 Tableau avec largeur relative



Annexes



Bibliographie

- [1] *Systèmes de management de la qualité*. International Organization for Standardization. 2015.
- [2] BRUNO DOUCHY. *Wiki du template AOCDTF*. Métier des Technologies Associées. 10 fév. 2021. URL : <https://github.com/aocdtf-mta/AOCDTF-template/wiki> (visité le 20/04/2021).
- [3] Bruno DOUCHY. *Recueil de programmations types*. AOCDTF, 2021. URL : <https://github.com/aocdtf-mta/AOCDTF-template/blob/main/matiere/cours/master.pdf>.
- [4] Bruno DOUCHY. « Titre d'un livre de référence issu d'une collection ». Dans : *Collection XYZ*. Dir. Un ÉDITEUR. Une maison d'édition, 2021.



Index

L

liste 22
répertoire 22

M

mot-clé 22

T

terme *voir* mot-clé



Index des termes du thème exemple

I

indexé 22

J

jugé 22

R

référéncé 22

S

situé 22



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.	3, 11, 13, 21, 23, 31, 34, 36
IDE Environnement de Développement (Integrated Development Environment)	13
SI Système International	35

