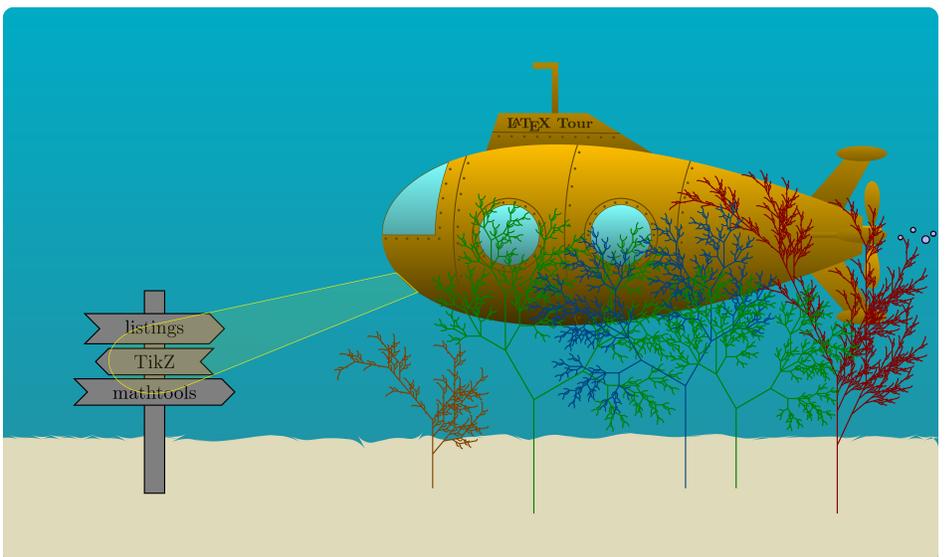


Immersion dans L^AT_EX (Deuxième édition)

NICOLAS KIELBASIEWICZ

29 novembre 2022



1	Introduction	7
2	Installation et utilisation	10
2.1	Installer \LaTeX	10
2.2	Quel éditeur de texte choisir?	10
2.3	Un mot sur les moteurs de compilation	11
3	Premier aperçu de \LaTeX	13
3.1	Un exemple pour commencer	13
3.2	Fichiers générés à la compilation	13
3.3	Comprendre les messages d'erreur	13
	Partie I — Les bases de \LaTeX	16
4	Organisation générale d'un document \LaTeX	17
4.1	La syntaxe de \LaTeX	17
4.2	Classe, préambule, extensions et corps de texte	17
4.3	Quelles extensions utiliser?	19
4.4	Extensions courantes et préambule initial	20
5	Structuration d'un document	23
5.1	Page de garde	23
5.2	Chapitrage et table des matières	24
5.3	Styles de pages	24
5.4	Mise en forme du texte	25
5.5	Notion de paragraphes et d'espacements	28
5.6	Les listes	29
5.7	Les notes de bas de page	29
6	Les mathématiques	31
6.1	Quel environnement pour les mathématiques?	31
6.2	Opérations usuelles	31
6.3	Variante de polices	32
6.4	Symboles mathématiques	33
6.5	Équations et systèmes d'équations	35
6.6	Théorèmes	38
7	Éléments complémentaires	39
7.1	Inclusion d'images	39
7.2	Tableaux et matrices	39
7.3	Les environnements flottants	41

7.4	Les listings de codes	42
8	Rédiger un CV avec <i>moderncv</i>	44
Partie II — Rédaction avancée de documents		48
9	Personnaliser son environnement \LaTeX	49
9.1	Définir de nouvelles commandes	49
9.2	Définir de nouveaux environnements	50
9.3	Définir de nouvelles boîtes de couleur avec <code>TCOLORBOX</code>	50
9.4	Modifier des commandes et environnements existants	51
9.5	Prévoir des formes étoilées pour les commandes que vous définissez	51
9.6	Personnaliser les commandes de titres	52
9.7	Gestion clé-valeur des arguments optionnels	53
9.8	Espaces parasites et commentaires	53
10	La bibliographie	54
10.1	Ecrire un fichier de bibliographie	54
10.2	Un peu d'histoire : <code>bib\TeX</code> et l'extension <code>NATBIB</code>	55
10.3	<code>biber</code> et l'extension <code>BIBLATEX</code>	56
10.4	La bibliographie en pratique	56
11	Les nomenclatures	58
11.1	Définir ses notations	58
11.2	Construire une nomenclature	58
11.3	Définir des catégories de notations	59
12	Références et liens hypertextes	60
12.1	Définir des étiquettes (=labels)	60
12.2	Définir une référence	60
13	Les tables des matières locales avec <code>MINITOC</code>	62
14	Les index	63
14.1	Génération de l'index	63
14.2	Les entrées de l'index	63
14.3	Personnaliser l'index	64
15	Annotations avec l'extension <code>TODONOTES</code>	65
15.1	Définir une annotation	65
15.2	Afficher la liste des notes	65
16	Gestion multi-fichiers	68
16.1	Séparer le corps du document en plusieurs fichiers	68
16.2	Créer une extension à partir de son préambule personnalisé	68

Partie III — Présentations en \LaTeX avec <i>beamer</i>	70
17 Introduction	71
17.1 La classe <i>beamer</i>	71
17.2 Un exemple pour commencer	71
18 Structuration d'une présentation	73
18.1 Le transparent de titre	73
18.2 Les transparents de contenu	73
18.3 Chapitrage, table des matières et automatisation	73
18.4 Les blocs	74
18.5 Les listes	75
18.6 Les environnements flottants	75
18.7 Théorèmes et équations	76
18.8 L'environnement <i>columns</i>	76
18.9 Variantes de polices supplémentaires	77
18.10 Références croisées et boutons	77
19 Animation d'une présentation	79
19.1 Les overlays	79
19.2 Les transitions de transparents	80
19.3 Animer une séquence d'images	81
20 L'apparence de votre présentation	83
20.1 Les thèmes prédéfinis	83
20.2 Personnalisation de l'apparence : cas général	87
20.3 Personnalisation de l'apparence : solutions usuelles	88
A Préambule \LaTeX de ce document	90
Index	97
Index des commandes et environnements	99

1.1	Définition d'une liste numérotée	7
5.2	Commandes et déclarations de changement de style	26
5.3	La commande <code>\emph</code>	26
5.4	Couleurs usuelles	27
5.5	Mélanges de couleurs	27
5.6	Les lettrines	28
5.7	Les espacements verticaux	28
5.8	Les espacements horizontaux	29
5.9	Définition d'une liste non numérotée	29
6.1	Utilisation des délimiteurs	32
6.2	Ajouter des commentaires avec <code>\overbrace</code> ou <code>\underbrace</code>	35
6.3	L'environnement <i>equation</i>	35
6.4	L'environnement <i>align</i>	36
6.5	Utilisation de <code>\nonumber</code> dans un environnement <i>align</i>	36
6.6	L'environnement <i>aligned</i>	37
6.7	L'environnement <i>cases</i> pour aligner des équations	37
6.8	L'environnement <i>cases</i> pour écrire une alternative	37
7.1	Inclure une image	39
7.2	Notre premier tableau	39
7.3	Tableaux et bordures pleines	40
7.4	Tableaux et bordures partielles	40
7.5	Tableaux mathématiques	41
7.6	Tableau flottant	42
7.7	Utilisation de la commande <code>\verb</code>	42
9.1	Définition d'une commande sans arguments	49
9.2	Définition d'une commande avec 2 arguments	49
9.3	Définition d'une commande avec 2 arguments dont un optionnel	49
9.4	Définition d'un environnement personnalisé	50

2.1	Fenêtre d'édition de Texworks	10
3.1	Document généré avec le code du tutoriel	14
5.1	Entête d'un document de classe <i>article</i> à gauche et page de garde d'un document de type <i>report</i> à droite	23
8.1	Le thème <i>classic</i> de la classe <i>moderncv</i> avec la couleur par défaut à gauche, et le choix du vert à droite	46
8.2	Les autres thèmes de CV proposés par la classe <i>moderncv</i> : <i>banking</i> , <i>casual</i> et <i>oldstyle</i>	47
10.1	BibDesk, éditeur de bibliographie pour MacOS	55
10.2	JabRef, éditeur de bibliographie multi-plateforme	55
17.1	Présentation générée avec le code du tutoriel	72
18.1	La table des matières	74
18.2	Les blocs	75
18.3	Les colonnes	77
20.1	Exemple de transparents avec le thème <i>default</i>	84
20.2	Exemple de transparents avec le thème <i>AnnArbor</i>	85
20.3	Exemple de transparents avec le thème <i>Antibes</i>	85
20.4	Exemple de transparents avec le thème <i>Berkeley</i>	85
20.5	Exemple de transparents avec le thème <i>Darmstadt</i>	86
20.6	Exemple de transparents avec le thème <i>Hannover</i>	86
20.7	Exemple de transparents avec le thème <i>Ilmenau</i>	86
20.8	Exemple de transparents avec le thème <i>JuanLesPins</i>	87
20.9	Exemple de transparents avec le thème <i>Luebeck</i>	87

6.1	Fonctions et opérateurs usuels	32
6.2	Les lettres grecques minuscules	34
6.3	Les lettres grecques majuscules	34
6.4	Quelques symboles d'opérateurs	34
6.5	Quelques symboles d'opérateurs de taille variable	34
6.6	Quelques symboles de relations	34
6.7	Quelques symboles divers	34
6.8	Quelques flèches	35
6.9	Les accents mathématiques	35
6.10	Les accents mathématiques de taille variable	35
7.1	Exemple de tableau flottant	42
20.1	Les thèmes existants et leurs composants	84

L'histoire commence avec l'invention de $\text{T}_{\text{E}}\text{X}$ en 1977 par le mathématicien et informaticien Donald Knut. Il s'agit d'un système de composition de documents, essentiellement destiné à l'édition de documents techniques. $\text{T}_{\text{E}}\text{X}$ vient du mot grec désignant l'art, la technique : $\tau\acute{\epsilon}\chi\nu\eta$. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, quant à lui, est créé en 1983 par le chercheur en informatique Leslie Lamport pour simplifier l'utilisation de $\text{T}_{\text{E}}\text{X}$ ¹.

Il s'agit donc d'un langage dédié à la génération de documents de toutes sortes : articles, rapports, mémoires, livres, lettres, romans, conférences, posters, ...et même partitions musicales! En particulier, il offre la possibilité de définir des formules mathématiques extrêmement simplement et avec une typographie qui est la référence aujourd'hui.

De manière générale, pour composer un document, il y a 2 méthodes :

- Utiliser des logiciels « tel écran, tel écrit » (en anglais WYSIWYG, acronyme de « What You See Is What You Get »), comme les logiciels propriétaires Microsoft Word (format .doc ou .docx), Apple Pages (format .pages) ou leurs équivalents dans le monde libre : OpenOffice, LibreOffice, ... (format .odt), et bien d'autres encore ...
- Utiliser des éditeurs de texte « basiques » pour écrire du texte éventuellement en utilisant certaines conventions : fichiers .txt, fichiers .html, fichiers Markdown .md, ...et générer un document dans un ou plusieurs formats spécifiques. On parle de contexte WYSIWYM (acronyme de « What You See is What You Mean », que l'on pourrait traduire par « Ce que vous voyez est ce que vous voulez dire »). L'idée motrice ici est de séparer le fond, le texte, et sa mise en forme.

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ rentre dans cette deuxième catégorie, ce qu'illustre l'exemple suivant :

Exemple 1.1 Définition d'une liste numérotée

<code>\begin{enumerate}</code>	
<code>\item blabla bla</code>	1. blabla bla
<code>\item bla blabla</code>	2. bla blabla
<code>\item bla bla bla</code>	3. bla bla bla
<code>\end{enumerate}</code>	

Si vous n'avez connu que des logiciels WYSIWYG, où la mise en forme de votre document se fait entièrement à la souris, vous serez un peu dérouté par la nécessité de « compiler » votre document pdf pour voir le rendu, et bien entendu, il va falloir le faire régulièrement pour éviter de mauvaises surprises! C'est la même méthodologie que l'écriture d'un programme informatique que l'on teste au fur et à mesure de son développement.

D'un autre côté, je vous garantis que lorsque vous aurez rédigé votre premier document en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, vous verrez tout de suite la différence de qualité en terme de typographie par rapport à la concurrence. Ce n'est pas un hasard si les logiciels WYSIWYG permettent de définir des boîtes dans lesquelles on écrit des formules mathématiques en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

1. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ signifie Lamport $\text{T}_{\text{E}}\text{X}$

Par ailleurs, la communauté \LaTeX est très développée et un très grand nombre d'extensions sont disponibles, certaines d'entre elles étant trop restrictives, d'autres étant désuètes. Savoir identifier quelles sont les extensions à utiliser ou à éviter n'est pas toujours facile.

L'extension d'un fichier \LaTeX est `.tex`. C'est un fichier texte dans lequel on manipule des commandes (ou macros) et des environnements. Le document généré est principalement un fichier PDF², le format de document le plus couramment utilisé à l'heure actuelle³.

A titre personnel, j'ai découvert \LaTeX en 2002 à ENSTA Paris grâce à Marc Baudoin et sa documentation personnelle qu'il a partagé avec les étudiants. A l'époque, elle en était à sa deuxième version, elle en est aujourd'hui à la cinquième : « Apprends \LaTeX (5ème édition) » (voir BAUDOIN, *Apprends \LaTeX (5ème édition)*). Comme tout le monde, j'ai mis un peu de temps à m'y mettre (surtout qu'à l'époque, la compilation directe en PDF était loin d'être le standard et les extensions étaient beaucoup moins nombreuses et moins ergonomiques qu'aujourd'hui), mais aujourd'hui, je ne saurais m'en passer, même pour les documents les plus élémentaires.

Mis à part la documentation de Marc Baudoin, la référence absolue, bien qu'elle commence à dater (2004), est « The \LaTeX Companion (2nd Edition) » de Frank Mittelbach et Michel Goossens (voir MITTELBACH et GOOSSENS, *The \LaTeX Companion*).

Enfin, vous avez :

- le site web de référence, celui du CTAN⁴ <https://www.ctan.org>, où vous trouverez la documentation officielle de la plupart des extensions existantes.
- la site web d'overleaf <https://fr.overleaf.com/learn>
- La principale FAQ française <http://www.grappa.univ-lille3.fr/FAQ-LaTeX>
- Une FAQ anglaise très bien fournie <https://texfaq.org>

Mais vous avez aussi stackexchange, ...

Dans ce cours introductif à \LaTeX , seules les extensions recommandées et les plus couramment utilisées seront évoquées ou présentées, comme étant le résultat de mon expérience personnelle, afin de vous éviter une recherche fastidieuse sans savoir réellement quoi chercher. Dans la bibliographie de ce document page 96, vous trouverez les liens vers la page CTAN de ces extensions.

La partie I est consacrée à vous présenter \LaTeX dans le but de rédiger un document manuscrit simple : un rapport, une lettre, un article, ...

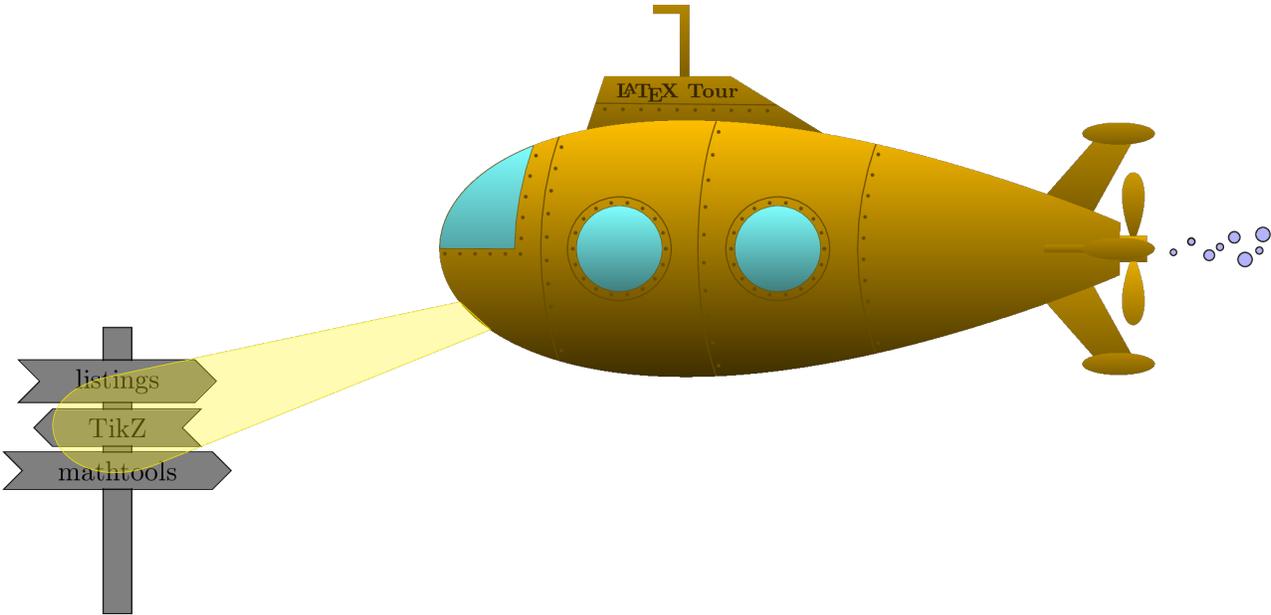
La partie II est consacrée aux éléments plus avancés concernant les gros documents ou les documents complexes : bibliographies, nomenclatures, index, ...

La partie III est consacrée quant à elle à la classe de documents *beamer*, pour écrire une présentation.

2. Mais on peut aussi générer entre autres des fichiers PostScript ou des fichiers HTML

3. PDF est l'acronyme de Portable Document Format, a été créé en 1992 et est devenu une norme ISO en 2008

4. Acronyme de Comprehensive \TeX Archive Network



2.1 Installer \LaTeX

\LaTeX est un ensemble d'extensions à télécharger, qui fait plusieurs Go! On parle de distribution \LaTeX .

sur Windows La distribution à télécharger s'appelle MikTeX <https://miktex.org>

sur Linux La distribution à télécharger s'appelle TeXLive <https://www.tug.org/texlive/>

sur MacOS La distribution à télécharger s'appelle MacTeX <http://www.tug.org/mactex/>. Il s'agit d'une version augmentée de la TeXLive, avec des applications spécifiques à Mac.

2.2 Quel éditeur de texte choisir?

Les éditeurs simples vi, emacs, gedit, kedit, ...(Linux), wordpad (Windows), TextEdit (Mac) ...

Les éditeurs avancés multi-plateformes Visual Studio Code ¹, Sublime Text, Atom, ...

Les éditeurs spécifiques OS-dépendants TexShop (Mac)¹, TeXnicCenter (Windows), ...

Les éditeurs spécifiques multi-plateformes Texworks¹ (figure 2.1), TeXstudio, TeXmaker, ...

```

1 \documentclass[11pt,french]{report}
2
3 \usepackage{myPoly}
4
5 % chargement du fichier de biblio
6 \addbibresource{latex.bib}
7
8 \hypersetup{
9   pdftitle={Initiation à \LaTeX},%
10  pdfauthor={N. Kielbasiewicz},%
11  pdfpagemode=UseOutlines,%
12 }%
13
14 \date{} % date %
15 \title{Initiation à \LaTeX} % titre %
16 \author{Nicolas Kielbasiewicz} % auteur %
17
18 \begin{document}
19
20 \maketitle
21
22 \doparttoc[n]
23 \noprof
24 \setcounter{tocdepth}{0}
25 \tableofcontents
26
27 \part{\LaTeX}
28
29 \chapter{Introduction}
30
31 \section{\LaTeX, c'est quoi ?}
32
33 L'histoire commence avec l'invention de \TeX{} en 1977 par le mathématicien et informaticien Donald Knut. Il s'agit d'un système de
composition de documents, essentiellement destiné à l'édition de documents techniques. \TeX{} vient du mot grec désignant l'art,
la technique: $\tau\alpha\upsilon\ \dot{\epsilon}\pi\sigma\iota\omicron\nu\ \lambda\epsilon\tau\alpha$. \LaTeX, quant à lui, est créé en 1983 par le chercheur en informatique Leslie
Lampport pour simplifier l'utilisation de \TeX{} \footnote{\LaTeX{} signifie Lampport \TeX{}}.

```

FIGURE 2.1 – Fenêtre d'édition de Texworks

Répondre aux questions suivantes peut vous aider à faire votre choix :

1. Les éditeurs que j'ai utilisés à titre personnel

- Souhaitez-vous les commandes les plus courantes dans les menus en mode presse-bouton ?
- Travaillez-vous sur plusieurs OS ?
- Souhaitez-vous la visualisation du document généré, réactualisé à chaque compilation ?
- Souhaitez-vous disposer de fonctionnalités avancées comme d'un correcteur orthographique et grammatical ?

Si vous utilisez un éditeur ne disposant pas d'un bouton pour générer le document (soit un éditeur simple, soit un éditeur avancé pour lequel vous n'avez pas chargé de module spécifique pour \LaTeX), il vous faudra alors lancer la génération en ligne de commande, en tapant par exemple l'une des commandes suivantes :

```
pdflatex monfichier.tex xelatex monfichier.tex lualatex monfichier.tex
```

Cette possibilité de générer le pdf en ligne de commande peut paraître contraignante pour les néophytes, voire trop « informaticienne ». Et pourtant, cela ouvre un champ de possibilités extrêmement large. En effet, \LaTeX apporte par exemple des solutions dans des domaines très divers, y compris dans le domaine du web. Ne vous est-il jamais arrivé d'adhérer en ligne à une association ou autre et de récupérer à la validation de votre inscription un document pdf qui contient toutes les informations remplies dans le formulaire ? Et bien, dites vous qu'il est possible (probable ?) que ce document pdf a été généré par \LaTeX !

Il existe enfin un autre moyen de générer des documents \LaTeX : par le biais de plateformes en ligne, comme Overleaf². Vous pouvez créer des documents en ligne, en bénéficiant de templates, mais aussi et surtout de manière collaborative.

2.3 Un mot sur les moteurs de compilation

2.3.1 \TeX

Créé en 1977 en même temps que le langage du même nom, c'est le tout premier moteur permettant de compiler des documents \LaTeX . La commande pour l'utiliser est `latex` et le format de sortie est DVI.

2.3.2 \pdfTeX

Créé en 1996 et utilisable depuis 2003, ce moteur apporte de nouveaux algorithmes de typographie (l'extension **MICROTYPE**?) ainsi que la possibilité de générer directement un fichier au format PDF. La commande pour l'utiliser est `pdflatex`.

2.3.3 \XeTeX et \LuaTeX

Ces moteurs de dernière génération³ apportent deux améliorations essentielles, le support complet de l'unicode (UTF-8), et une gestion propre des polices de caractères. \LuaTeX apporte en plus un nouveau langage, `lua`, pour écrire les extensions \LaTeX et y remplacer le langage \TeX . Les commandes pour les utiliser sont `xelatex` et `lualatex` et tous 2 génèrent des fichiers PDF.

2. <https://www.overleaf.com>

3. \LuaTeX a été créé en 2007 et est utilisable depuis 2016.

2.3.4 Qui choisir ?

La plupart du temps, utiliser `pdflatex` ou `lualatex` indifféremment est totalement transparent pour vous. Mais certains trucs et astuces liés à la gestion des caractères accentués n'ayant plus lieu d'être avec `lualatex`, je recommande quand même d'utiliser ce dernier. Je vous montrerai par ailleurs comment gérer les petites différences.

3.1 Un exemple pour commencer

```
\documentclass[11pt, a4paper, french]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{amssymb}
\usepackage{fourier}
\usepackage{mathtools}
\usepackage{geometry}
\usepackage{babel}
\usepackage{microtype}

% modification des dimensions de la page et de son centrage %
\geometry{top=2.5cm, bottom=2.5cm, left=1.8cm, right=1.8cm}

\title{Mon premier document en \LaTeX} % titre %
\author{Vous} % auteur %

\begin{document}

\maketitle

Bonjour à tous les nouveaux utilisateurs de \LaTeX en <2022> !!!

\end{document}
```

3.2 Fichiers générés à la compilation

La compilation du document `tutorial.tex`, si elle se déroule sans erreurs, générera principalement 3 fichiers : `tutorial.pdf` bien sûr, un fichier auxiliaire `tutorial.aux` contenant principalement les références croisées (voir chapitre 12) et le fichier de log `tutorial.log`.

3.3 Comprendre les messages d'erreur

La compilation d'un document L^AT_EX génère plusieurs fichiers d'extensions différentes. Le fichier d'extension `.log` contient tout ce qui s'affiche dans le terminal quand vous compilez en ligne de commande, ou dans la console de sortie si vous utilisez un éditeur spécifique.

```
This is pdfTeX, Version 3.141592653-2.6-1.40.24 (TeX Live 2022) (preloaded format=pdflatex
 2022.11.3) 29 NOV 2022 17:07
entering extended mode
restricted \write18 enabled.
%&-line parsing enabled.
** tutorial.tex
(./ tutorial.tex
LaTeX2e <2022-11-01>
```

...

3.3. COMPRENDRE LES MESSAGES D'ERREUR

```
Mon premier document en LATEX
Vous
29 novembre 2022
Bonjour à tous les nouveaux utilisateurs de LATEX en <2022>!!!
```

1

FIGURE 3.1 – Document généré avec le code du tutoriel

```
[1
{/usr/local/texlive/2022/texmf-var/fonts/map/pdftex/updmap/pdftex.map}] (./tutorial.aux)
Here is how much of TeX's memory you used:
8231 strings out of 474971
133273 string characters out of 5776123
523274 words of memory out of 5000000
29572 multiletter control sequences out of 15000+600000
528486 words of font info for 124 fonts, out of 8000000 for 9000
1141 hyphenation exceptions out of 8191
68i,6n,76p,1005b,428s stack positions out of 10000i,1000n,20000p,200000b,200000s
{/usr/local/texlive/2022/texmf-dist/fonts/enc/dvips/base/8r.enc}</usr/local/texlive/2022/texmf-dist/fonts/type1/adobe/utopia/putr8a.pfb>
Output written on tutorial.pdf (1 page, 16660 bytes).
PDF statistics:
14 PDF objects out of 1000 (max. 8388607)
8 compressed objects within 1 object stream
0 named destinations out of 1000 (max. 500000)
30721 words of extra memory for PDF output out of 35830 (max. 10000000)
```

Je vous épargne les quelques 581 lignes intermédiaires qui se rapportent au chargement des extensions.

Voilà quelques messages d'erreurs potentiels :

Vous utilisez une commande qui n'existe pas

```
! Undefined control sequence.
l.22 ...à tous les nouveaux utilisateurs de \latex
                                     \ en <2022> !!!
?
```


Les bases de L^AT_EX

Partie I

4	Organisation générale d'un document L^AT_EX	17
4.1	La syntaxe de L ^A T _E X	17
4.2	Classe, préambule, extensions et corps de texte	17
4.3	Quelles extensions utiliser?	19
4.3.1	A propos des polices de caractères	19
4.3.2	Les extensions systématiques	20
4.4	Extensions courantes et préambule initial	20
5	Structuration d'un document	23
5.1	Page de garde	23
5.2	Chapitrage et table des matières	24
5.3	Styles de pages	24
5.4	Mise en forme du texte	25
5.4.1	Variantes de polices	25
5.4.2	Les couleurs	26
5.4.3	Les letrines	27
5.5	Notion de paragraphes et d'espacements	28
5.6	Les listes	29
5.7	Les notes de bas de page	29
6	Les mathématiques	31
6.1	Quel environnement pour les mathématiques?	31
6.2	Opérations usuelles	31
6.2.1	Les fractions	31
6.2.2	Exposants et indices	31
6.2.3	Fonctions et opérateurs usuels et délimiteurs	32
6.3	Variantes de polices	32
6.4	Symboles mathématiques	33
6.4.1	Lettres grecques	33
6.4.2	Symboles d'opérateurs et de relations	33
6.4.3	Symboles divers et flèches	34
6.4.4	Les accents	34
6.5	Équations et systèmes d'équations	35
6.6	Théorèmes	38
7	Éléments complémentaires	39
7.1	Inclusion d'images	39
7.2	Tableaux et matrices	39
7.3	Les environnements flottants	41
7.4	Les listings de codes	42
8	Rédiger un CV avec <i>moderncv</i>	44

4.1 La syntaxe de L^AT_EX

Pour L^AT_EX, une commande est un mot composé en règle générale uniquement de lettres et qui commence par un antislash. Par exemple, `\LaTeX`, `\documentclass`, `\usepackage` sont des commandes.

Une commande peut avoir des arguments obligatoires (chaque argument est entre accolades), et/ou des arguments optionnels (l'ensemble des arguments optionnels est entre crochets et chaque argument optionnel est séparé par une virgule à l'intérieur).

```
\documentclass[11pt, a4paper, french]{article}
```

Par exemple, dans l'exemple ci-dessus, `\documentclass` a un argument, *article* et 3 arguments optionnels, *11pt*, *a4paper* et *french*.



Vous avez sans doute remarqué dans l'exemple de la section 3.1 que la commande `\LaTeX` est suivie d'un antislash. La raison est simple, c'est que par défaut il n'y a aucun espacement entre une commande et ce qui la suit. En lieu et place de l'anti-slash, nous aurions pu mettre la commande entre accolades ou lui ajouter un argument vide (donc entre accolades). Un anti-slash suivi d'un espace est une façon d'imposer un espace. Personnellement, je trouve que c'est la manière la plus lisible pour rajouter un espace suite à une commande.

Il existe des commandes sans arguments qui formatent la suite du corps de texte sans avoir de limite de portée. Il s'agit des déclarations. Pour limiter la portée, on utilise les accolades. Cela donne une syntaxe de la forme :

```
{\cmd bla bla bla}
```

Nous avons également vu une autre notion, celle d'un environnement. Le contenu d'un environnement est encerclé par un `\begin` et par un `\end`, prenant tous deux en argument le nom de l'environnement. *document* et *enumerate* sont les 2 exemples que vous avez rencontrés jusqu'ici.

Enfin, nous pouvons également écrire des commentaires, comme dans tout langage de programmation, auquel cas le commentaire doit commencer par un `%` et se terminera avec la fin de la ligne.

4.2 Classe, préambule, extensions et corps de texte

Décryptons un peu l'exemple de la section 3.1.

La première ligne d'un document L^AT_EX définit la classe de document :

```
\documentclass[11pt, a4paper, french]{article}
```

La classe de document définit la mise en page générale du document et les commandes principales de structuration du document. Les classes principales sont :

4.2. CLASSE, PRÉAMBULE, EXTENSIONS ET CORPS DE TEXTE

article pour la rédaction d'articles, c'est-à-dire essentiellement des documents relativement courts sans page de garde, constitués de sections.

report pour la rédaction de rapports ou de mémoires, donc des documents plus longs et constitués de chapitres.

book pour la rédaction de livres, des documents encore plus longs, constitués de chapitres, et destinés à être imprimés en recto-verso et reliés.

beamer pour la rédaction de présentations, composées de diapositives. Il existe d'autres classes, mais **beamer** est de loin la plus complète et la plus utilisée.

tikzposter pour la rédaction de posters, c'est-à-dire un document d'une seule page au format A0.

Le corps de texte est le contenu effectif du document à rédiger. Il est contenu à l'intérieur d'un environnement *document* :

```
\author{Vous} % auteur %  
\begin{document}  
\maketitle  
Bonjour à tous les nouveaux utilisateurs de \LaTeX\ en <2022> !!!  
\end{document}
```

Le préambule d'un document \LaTeX est tout ce qui se trouve entre la définition de la classe et le début de l'environnement *document*.

```
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage{amssymb}  
\usepackage{fourier}  
\usepackage{mathtools}  
\usepackage{geometry}  
\usepackage{babel}  
\usepackage{microtype}  
  
% modification des dimensions de la page et de son centrage %  
\geometry{top=2.5cm, bottom=2.5cm, left=1.8cm, right=1.8cm}
```

Il contient 2 types d'éléments :

1. L'inclusion d'extensions (ou packages), avec la commande `\usepackage`. Une extension est un ensemble de commandes et d'environnements dédiés à un besoin particulier, à priori indépendamment de la classe de document.
2. Le paramétrage des extensions et donc du document (définition des couleurs, styles des titres, commandes personnalisées, ...). C'est aussi là que l'on définit le titre, l'auteur et la date du document, qui seront exploités par la commande `\maketitle`



On peut définir ses propres classes de documents et/ou extensions. Il s'agit de fichiers d'extensions respectives `.cls` et `.sty`.

4.3 Quelles extensions utiliser ?

4.3.1 A propos des polices de caractères

Avec le moteur pdfTeX

Ces polices sont matérialisées par l'utilisation d'une extension \LaTeX . La police de caractères par défaut de \LaTeX est Computer Modern.

Pour les charger, il faut d'abord inclure les extensions `FONTENC`¹ et `INPUTENC`² :

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
```

La première permet de spécifier l'encodage des accents (Windows Latin 1, utf-8, ...) dans votre fichier .tex. La norme aujourd'hui est utf8 par défaut, que ce soit sous Windows, Mac ou Linux.

Ensuite, on charge l'extension dédiée à la police proprement dite. En voici une liste loin d'être exhaustive :

LMODERN permet d'utiliser la police Latin Modern (successeur désigné de Computer Modern). Voir JACKOWSKI et NOWACKI, *lm – Latin modern fonts in outline formats* ;

HELVET permet d'utiliser la police Helvetica. Voir SCHMIDT, *helvet – Load Helvetica, scaled* ;

LIBERTINE permet d'utiliser les polices Libertine et Biolinum. Voir TENNENT, *libertine – Use of Linux Libertine and Biolinum fonts with LaTeX* ;

FOURIER permet d'utiliser la police Fourier, basée sur Adobe Utopia. Voir BOVANI, *fourier – Using Utopia fonts in LATEX documents* ;

KPFFONTS permet d'utiliser la police Kepler, dérivée initialement sur URW Palladio, mais qui en est différente d'aspect. Voir CAIGNAERT, *kpfonts – A complete set of fonts for text and mathematics* ;

AMSFONTS permet d'utiliser la police mathématique par défaut, automatiquement chargée avec les extensions usuelles dédiées aux symboles mathématiques ;

MATHPAZO permettent d'utiliser une police mathématique liée à la police Palatino .

A titre personnel, j'ai utilisé **KPFFONTS** et j'utilise maintenant **FOURIER** essentiellement pour 2 raisons :

- Elles permettent de combiner toutes les variantes de polices (combinaisons de gras, italique, petites majuscules, ...), ce qui n'est pas le cas de toutes les polices.
- Elles sont extrêmement complètes sur les symboles mathématiques.

Avec les moteurs XeTeX et LuaTeX

Ces moteurs ont complètement révolutionné la gestion des polices de caractères dans \LaTeX . Avec eux, on ne charge plus l'extension `INPUTENC`, puisque ces moteurs considèrent que les fichiers .tex sont nécessairement encodés en utf8. Par ailleurs, l'extension `FONTENC` est remplacée par l'extension `FONTSPEC`³.

```
\usepackage{fontspec}
```

Cette extension permet d'utiliser les polices True Type installées sur votre ordinateur (les mêmes que celles que vous pouvez utiliser dans n'importe quel logiciel de traitement de texte WYSIWYG, comme Microsoft Word, Apple Pages, OpenOffice Writer, ...).

1. *fontenc – Standard package for selecting font encodings.*

2. JEFFREY et MITTELBACH, *inputenc – Accept different input encodings.*

3. ROBERTSON, *fontspec – Advanced font selection in XeLaTeX and LuaLaTeX.*

Par exemple, pour charger la police Verdana, on écrit :

```
\setmainfont{Verdana}
```

Néanmoins, la plupart de ces polices n'apportent pas de polices mathématiques. Il est donc nécessaire d'en charger une. C'est la raison pour laquelle l'extension **UNICODE-MATH**⁴ a été créée.

Enfin, les polices de caractères \LaTeX traditionnelles ont aussi été adaptées pour être utilisées dans ce contexte :

NEWCOMPUTERMODERN en lieu et place de **LMODERN** chargée par défaut. Voir TSOLOMITIS, *NewComputer-Modern – Computer Modern fonts including matching non-latin alphabets*;

LIBERTINUS-OTF en lieu et place de **LIBERTINE**. Voir VOR, *libertinus-otf – Support for Libertinus OpenType*;

FOURIER-OTF en lieu et place de **FOURIER**. Voir FLIPO, *Erewhon-math – Utopia based OpenType Math font*;

KPFFONTS-OTF en lieu et place de **KPFFONTS**. Voir FLIPO, *kpffonts-otf – OpenType version of the kpffonts (Type1) designed by Christophe Caignart*

A titre personnel, j'utilise **FOURIER-OTF**, en lieu et place de **FOURIER**, sans avoir testé les autres.

4.3.2 Les extensions systématiques

On a vu ce qui concerne les polices de caractères. Il y a d'autres extensions à inclure systématiquement :

```
...
\usepackage{geometry}
\usepackage{babel}
\usepackage{microtype}
```

GEOMETRY définit les marges d'une page, comme montré dans l'exemple en début de chapitre. Voir CARLISLE et UMEKI, *geometry – Flexible and complete interface to document dimensions*;

BABEL charge la langue dans laquelle vous allez écrire. En effet, chaque langue a ses propres règles typographiques et cette extension est là pour les prendre en compte à la génération du document. Voir LÓPEZ et BRAAMS, *babel – Multilingual support for Plain \TeX or \LaTeX* ;

MICROTYPE apporte des raffinements d'ordre typographique, comme l'ajustement de l'espacement permettant une meilleure coupure des mots. Le simple fait de charger cette extension suffit à avoir des améliorations notables. Voir SCHLICHT, *microtype – Subliminal refinements towards typographical perfection*.

4.4 Extensions courantes et préambule initial

L'extension GRAPHICX Cette extension concerne l'inclusion des images, et les formats d'images pris en comptes. Voir CARLISLE, *graphicx – Enhanced support for graphics*;

L'extension XCOLOR Cette extension concerne la définition des couleurs, avec la possibilité de les mélanger de manière simple. Voir KERN, *xcolor – Driver-independent color extensions for \LaTeX and \pdf\LaTeX* ;

Les extensions de maths Pour pouvoir écrire des formules mathématiques, il faut charger des extensions de symboles et d'environnements. Les extensions suivantes sont en général auto-suffisantes : **MATHTOOLS**⁵ (qui remplace **AMSMATH**) et **AMSSYMB**, pour charger la plupart des symboles, et **AMSTHM**⁶ pour définir et gérer les théorèmes.

4. ROBERTSON et al., *unicode-math – Unicode mathematics support for XeTeX and LuaTeX*.

5. MADSEN et al., *mathtools – Mathematical tools to use with amsmath*.

6. *amsthm – Typesetting theorems (AMS style)*.



L'extension **MATHTOOLS** est chargée par **FOURIER-OTF**. Elle doit donc être chargée avant si vous voulez utiliser des options particulières. **AMSSYMB** est quand à elle complètement réécrite par **FOURIER-OTF**.

L'extension BIBLATEX Cette extension est plus que recommandée dès lors que l'on veut manipuler des bibliographies. Voir KIME et LEHMAN, *BibLaTeX – Sophisticated Bibliographies in L^AT_EX*;

L'extension LISTINGS Cette extension est à l'heure actuelle l'extension la plus complète pour insérer des listings de code, avec un nombre élevé de langages pré-définis, et la possibilité de définir ses propres langages pour la mise en page d'un listing. Voir HOFFMANN, HEINZ et MOSES, *listings – Typeset source code listings using L^AT_EX*;

L'extension HYPERREF Cette extension est dédiée aux références croisées et à la possibilité de mettre des liens cliquables dans des documents. Voir RAHTZ et OBERDIEK, *hyperref – Extensive support for hypertext in L^AT_EX*.

La grande diversité d'extensions fait qu'il existe d'une part des redondances entre certaines d'entre elles, et d'autre part des problèmes de compatibilité. Pour résoudre cela, il faut non seulement choisir les bonnes extensions, mais aussi les charger dans un certain ordre.

Les règles générales sont assez simples :

1. Les extensions gérant la police sont à charger en premier.
2. Sauf contre-indication, **BABEL** est la dernière extension à charger, les exceptions les plus courantes étant **HYPERREF**, **LISTINGS** et bien entendu **MICROTYPE**, à charger juste derrière **BABEL**.
3. Quand une extension doit être chargée avant une autre, elle affiche de plus en plus souvent un warning à la génération du document.

Par ailleurs, on a pu observer des différences notables selon que l'on compile avec le moteur pdfL^AT_EX d'une part, ou avec l'un des moteurs XeL^AT_EX ou LuaL^AT_EX d'autre part. Il existe une extension qui permet de savoir quel moteur est effectivement utiliser : l'extension **IFTTEX**. Voir *iftex – Am I running under pdfTeX, XeTeX or LuaTeX?*

Avec l'ensemble des extensions évoquées jusqu'à présent, mais aussi celles, le préambule s'écrit donc, en utilisant la police Fourier et en exploitant les possibilités de **IFTTEX** :

```
\usepackage{iftex}
\ifPDFTeX
  \usepackage[utf8]{inputenc}
  \usepackage[T1]{fontenc}
  \usepackage{amssymb}
  \usepackage{fourier}
  \usepackage{mathtools}
\else
  \usepackage{fontspec}
  \usepackage{mathtools}
  \usepackage{fourier-otf}
\fi
\usepackage{geometry}
\usepackage{graphicx}
\usepackage{xcolor}
\usepackage{amsthm}
\usepackage[style=authortitle]{biblatex}
\usepackage{babel}
\usepackage{microtype}
\usepackage{listings}
\usepackage[plainpages=false, pdfpageLabels, bookmarksnumbered, breaklinks=true]{hyperref}
```

Pour aller plus loin et voir la liste des extensions utilisées pour le présent document, voir annexe A.



Si vous utilisez dans votre document des commandes spécifiques à une extension qui vous demande de choisir définitivement entre pdf \TeX et Lua \TeX par exemple, alors il est inutile de garder la possibilité d'inclure les bonnes extensions dans tous les cas. C'est le cas par exemple de ce document qui va présenter, entre autres, des commandes issues de **UNICODE-MATH**, qui ne fonctionne qu'avec Lua \TeX . Afin de garantir qu'un document soit compilé uniquement avec le moteur adéquat, on peut là encore exploiter les possibilités de l'extension **IFT \TeX** en utilisant l'une des commandes `\RequirePDF \TeX` , `\RequireXe \TeX` ou `\RequireLua \TeX` (parmi d'autres possibilités).

5.1 Page de garde

Par défaut, 3 informations sont demandées dans le préambule : le titre du document, le ou les auteurs, et la date. Si la date n'est pas précisée, alors celle qui correspond à la date de la dernière compilation du document est utilisée. Ces 3 informations permettent de définir quand elle existe la page de garde du document, ou à défaut l'entête d'un article. L'utilisation de la commande `\maketitle` permet de générer ce titre ou cette page de garde.

```
\geometry{top=2.5cm, bottom=2.5cm, left=1.8cm, right=1.8cm}

\title{Mon premier document en \LaTeX} % titre %
\author{Vous} % auteur %

\begin{document}
```

Mon premier document en L^AT_EX

Vous

11 octobre 2019

Mon premier document en L^AT_EX

Vous

11 octobre 2019

FIGURE 5.1 – Entête d'un document de classe *article* à gauche et page de garde d'un document de type *report* à droite

5.2 Chapitrage et table des matières

Il existe 7 niveaux de titres dans un document \LaTeX . Suivant la classe que vous utilisez (`book`, `report`, ou `article`), certains niveaux ne sont pas disponibles.

`\part` Définition d'un titre de niveau le plus élevé (disons niveau 1) : la partie. La commande `\part` n'est disponible que pour les classes `book` et `report` et chaque partie est numérotée.

`\chapter` Définition d'un titre de niveau 2 : le chapitre. La commande `\chapter` n'est disponible que pour les classes `book` et `report` et chaque chapitre est numéroté.

`\section` Définition d'un titre de niveau 3 : la section. C'est le niveau le plus élevé disponible dans la classe `article`. Chaque section est numérotée.

`\subsection` Définition d'un titre de niveau 4 : la sous-section, qui est là encore numérotée.

`\subsubsection` Définition d'un titre de niveau 5 : la sous-sous-section, qui n'est numérotée que dans la classe `article`.

`\paragraph` Définition d'un titre de niveau 6 : le paragraphe. Il n'est pas numéroté. Contrairement aux 5 premiers niveaux, le texte qui suit n'est pas précédé d'un passage automatique à la ligne.

`\subparagraph` Définition d'un titre de niveau 7 : le sous-paragraphe. Il n'est pas numéroté et par rapport au `\paragraph`, ce titre est indenté.



On peut également définir une annexe, auquel cas les titres de l'annexe ne sont pas numérotés par des chiffres, mais par des lettres. Il suffit pour cela d'utiliser la commande `\appendix`, sans arguments, pour déclarer que tout ce qui suit est l'annexe du document.

Maintenant que l'on a défini la structure de notre document, on peut souhaiter avoir une table des matières de notre document.

Pour cela, à l'endroit souhaité, il suffit juste d'appeler la commande `\tableofcontents`, comme on a pu le voir en page 3



Quand on demande une table des matières, il faut compiler le document 2 fois : une première fois pour générer le fichier auxiliaire d'extension `.toc`, la deuxième fois pour en tenir compte à la génération du document.



Il existe des versions dites étoilées (les commandes sont donc `\part*`, `\chapter*`, ...) qui définissent un titre de niveau correspondant, mais sans numérotation. Dans ce cas, les titres n'apparaîtront pas dans la table des matières, à moins de le demander avec la commande `\addcontentsline`

```
\addcontentsline [ toc ] { niveau } { titre }
```

`toc` permet de préciser que l'on ajoute une entrée dans la table des matières.

Le `niveau` correspond au niveau de titre que l'on veut ajouter dans la table des matières : `part`, `chapter`, `section`, ...

5.3 Styles de pages

Les styles de pages par défaut sont essentiellement : `plain` et `empty`. On définit ce paramètre à l'aide de la commande `\pagestyle`.

En utilisant l'extension `FANCYHDR`¹, on peut définir un style de page personnalisé et donc le contenu de l'entête et du pied d'une page.

Voici la version basique du paramétrage utilisé dans ce document (sans les couleurs) :

```
\pagestyle{fancy} % on va mettre en tete et pied de page personnalise %

\fancypagestyle{plain}{%
  \fancyhf{}
  \fancyfoot{}
  \renewcommand\headrulewidth{0pt}
  \renewcommand\footrulewidth{0pt}
}

\fancyhead[RE]{\leftmark} % entête a droite pour les pages paires : le chapitre en cours (classe
  book)
\fancyhead[LO]{\rightmark} % entête a gauche pour les pages impaires : la section en cours
  (classe book) ou tous les entêtes pour la classe report
\fancyhead[RO,LE]{\thepage} % numero de page dans l'autre emplacement de l'entête
\fancyfoot{} % pas de pied de page
```

5.4 Mise en forme du texte

5.4.1 Variantes de polices

Nous avons vu jusqu'à présent comment organiser du texte sans réellement agir sur celui-ci. Bien entendu, \LaTeX permet de formater le texte lui-même.

On peut tout d'abord contrôler la taille du texte :

`\tiny` Exemple : Ceci est une phrase de taille *tiny*

`\scriptsize` Exemple : Ceci est une phrase de taille *scriptsize*

`\footnotesize` Exemple : Ceci est une phrase de taille *footnotesize*

`\small` Exemple : Ceci est une phrase de taille *small*

`\normalsize` Exemple : Ceci est une phrase de taille *normalsize*

`\large` Exemple : Ceci est une phrase de taille *large*

`\Large` Exemple : Ceci est une phrase de taille *Large*

`\LARGE` Exemple : Ceci est une phrase de taille *LARGE*

`\huge` Exemple : Ceci est une phrase de taille *huge*

`\Huge` Exemple : Ceci est une phrase de taille *Huge*



La commande `\normalsize` ne fait rien, cette taille correspond en effet à la taille définie en option de la classe de document. Elle ne sert qu'à remettre la taille normale si on a changé la taille quelque part dans le document.



Ces commande sont plutôt destinées à la personnalisation des titres, des tableaux ou des listings de codes qu'à modifier le corps de texte proprement dit (ce qui est désagréable à lire).

1. OOSTRUM, *fancyhdr – Extensive control of page headers and footers in \LaTeX 2e*.

On peut également changer le style du texte :

Exemple 5.2 Commandes et déclarations de changement de style

<code>\textit{italique}, {\itshape italique\}</code> \\	<i>italique, italique</i>
<code>\textsl{incliné}, {\slshape incliné\}</code> \\	<i>incliné, incliné</i>
<code>\textsc{petites capitales}, {\scshape petites capitales}</code> \\	PETITES CAPITALES, PETITES CAPITALES
<code>\textup{droit}, {\upshape droit}</code> \\	droit, droit
<code>\textbf{gras}, {\bfseries gras}</code> \\	gras, gras
<code>\textrm{romain}, {\rmfamily romain}</code> \\	romain, romain
<code>\textsf{sans empattement}, {\sffamily sans empattement}</code> \\	sans empattement, sans empattement
<code>\texttt{à chasse fixe}, {\ttfamily à chasse fixe}</code>	à chasse fixe, à chasse fixe



On peut bien entendu combiner les commandes et les déclarations de changements de style. Le résultat dépendra néanmoins de ce qu'autorise la police de caractère chargée. Si utiliser les déclarations pour de tels mélanges économise les accolades (une seule paire suffit), les commandes contrôlent davantage ce qui doit se passer. Si vous considérez `\itshape` et que vous collez un mot juste derrière de style droit, alors la dernière lettre en italique pourra chevaucher la première lettre qui suit. Dans ce cas, il est recommandé d'ajouter `\/` avant l'accolade fermante de `\itshape`. Par contre, rien à faire avec `\textit!`



Un mot sur l'utilisation du texte en italique. On l'utilise en général pour mettre en valeur une partie d'un texte. Il existe en \TeX une commande pour mettre en valeur du texte : `\emph` qui est l'exemple typique de la philosophie WYSIWYM de \TeX , à savoir différencier la structure du document et son rendu visuel. `\emph` modifie en effet le style du texte en fonction du style de départ :

- Pour du texte normal, `\emph` est l'équivalent de `\textit`, le texte mis en valeur sera en italique.
- Pour du texte qui est déjà en italique, si `\textit` ne fera rien, `\emph` va le mettre en valeur en étant cette fois l'équivalent de `\textup`, à savoir du texte normal.

Exemple 5.3 La commande `\emph`

```
{\upshape Bonjour, \emph{cher utilisateur}, comment allez-vous ?} \\
{\itshape Bonjour, \emph{cher utilisateur}, comment allez-vous ?\}
```

Bonjour, *cher utilisateur*, comment allez-vous?
Bonjour, cher utilisateur, *comment allez-vous?*

5.4.2 Les couleurs

On peut enfin changer la couleur du texte à l'aide de la déclaration `\color` :

Exemple 5.4 Couleurs usuelles

<code>\begin{tabular}{cc}</code>			
<code>{\color{blue} bleu}</code>	<code>& {\color{red} rouge} \\</code>	bleu	rouge
<code>{\color{green} vert}</code>	<code>& {\color{yellow} jaune} \\</code>	vert	jaune
<code>{\color{orange} orange}</code>	<code>& {\color{pink} rose} \\</code>	orange	rose
<code>{\color{violet} violet}</code>	<code>& {\color{black} noir} \\</code>	violet	noir
<code>{\color{white} blanc}</code>	<code>& {\color{gray} gris} \\</code>	blanc	gris
<code>{\color{cyan} cyan}</code>	<code>& {\color{magenta} magenta}</code>	cyan	magenta
<code>\end{tabular}</code>			

L'extension **xcolor**² vous apporte un certain nombre de fonctionnalités supplémentaires :

- Elle ajoute la commande `\textcolor` qui peut remplacer la déclaration `\color`;
- Elle permet de charger une bibliothèque plus conséquente de couleurs pré-définies;
- Elle permet également de définir des couleurs (par leur code RGB, CMYK, ...) avec la commande `\definecolor` :

```
\definecolor{maincolor}{RGB}{191,63,0}
```

- Elle permet surtout de mélanger les couleurs avec une syntaxe intuitive, comme le montre l'exemple 5.5.

Exemple 5.5 Mélanges de couleurs

<code>{\color{blue!50!green} 50% bleu , 50% vert} \\</code>	50% bleu, 50% vert
<code>{\color{red!50!yellow} 50% rouge , 50% jaune} \\</code>	50% rouge, 50% jaune
<code>{\color{blue!75!black} 75% bleu , 25% noir} \\</code>	75% bleu, 25% noir
<code>{\color{blue!50!black} 50% bleu , 50% noir} \\</code>	50% bleu, 50% noir
<code>{\color{blue!25!black} 25% bleu , 75% noir}</code>	25% bleu, 75% noir

- On peut enfin définir une couleur à partir d'un mélange avec la commande `\colorlet` :

```
\colorlet{maincolor}{yellow!75!red}
```

5.4.3 Les lettrines

L'extension **LETTRINE**³ permet, comme son nom l'indique, de définir des lettrines, et cela à l'aide de la commande `\lettrine`.

2. KERN, *xcolor – Driver-independent color extensions for ETX and $\text{pdf}\text{E}\text{T}\text{X}$.*
 3. FLIPO, *lettrine – Typeset dropped capitals.*

Exemple 5.6 Les lettrines

```
\lettrine[lines=3,
  nindent=2ex]{\(\mathfrak{L}\)}{'extension}
\package{lettrine} permet, comme son
nom l'indique, de définir des
lettrines, et cela à l'aide de la
commande \cmd{lettrine}. Cette
commande accepte des options sous
forme clé-valeur pour contrôler
l'affichage de la lettrine.
```

 L'EXTENSION **LETRINE** permet, comme son nom l'indique, de définir des lettrines, et cela à l'aide de la commande `\lettrine`. Cette commande accepte des options sous forme clé-valeur pour contrôler l'affichage de la lettrine.

La commande `\lettrine` prend 2 arguments : la lettrine proprement dite puis le reste du premier mot de la phrase, qui va apparaître en petites capitales. Il y a aussi un argument optionnel permettant de paramétrer la lettrine. Voici quelques unes des options disponibles :

lines permet de spécifier la hauteur de la lettrine en nombre de lignes. La valeur par défaut est 2.

nindent permet de définir une marge à droite de la lettrine, qui s'applique à toutes les lignes indentées à partir de la deuxième. La valeur par défaut est 1em.

image sans valeur, qui permet de dire que le premier argument de la commande `\lettrine` est le nom d'une image et pas une lettre.

5.5 Notion de paragraphes et d'espacements

Dans un document `.tex`, pour changer de paragraphe, il suffit de passer une ligne.

Si maintenant on veut passer une ligne dans le document final, on utilise la commande `\`. Si on veut contrôler plus finement l'espacement vertical, on peut utiliser soit l'une des 3 commandes spécifiques `\smallskip`, `\medskip` et `\bigskip`, soit la commande générale `\vspace` qui prend une dimension en argument.

Exemple 5.7 Les espacements verticaux

ligne 0	
ligne 1	
<code>\smallskip</code>	ligne 0
ligne 2	ligne 1
<code>\medskip</code>	ligne 2
ligne 3	ligne 3
<code>\bigskip</code>	ligne 4
ligne 4	

Pour les espacements horizontaux, on dispose de la commande générale `\hspace`, qui prend une dimension en argument, mais aussi d'une série de commandes spécifiques illustrées dans l'exemple suivant :

Exemple 5.8 Les espacements horizontaux

```
ab c-d\,e\:f\;g\quad h\quad i
```

```
ab c d e f g h i
```

Par défaut, le texte est justifié (aligné à gauche et à droite simultanément). On peut utiliser les environnements *center*, *flushleft* et *flushright* pour centrer le texte ou ne l'aligner que d'un côté.



Il existe une déclaration équivalente pour le centrage du texte : `\centering`. Il est fortement conseillé de l'utiliser à l'intérieur d'autres environnements si vous voulez centrer le contenu, comme c'est le cas des environnements flottants (voir la section 7.3 les concernant).

5.6 Les listes

Les 3 types de listes ont chacun leur environnement :

itemize pour les listes non numérotées. En français, les puces sont des tirets. En anglais, elles sont rondes.

Exemple 5.9 Définition d'une liste non numérotée

```
\begin{itemize}
\item item 1
\item item 2
\item item 3
\end{itemize}
```

- item 1
- item 2
- item 3

enumerate pour les listes numérotées (voir exemple 1.1)

description pour les listes descriptives, telles que celle dans laquelle nous sommes pour définir les 3 environnements de type liste.



On peut souhaiter personnaliser les puces de liste comme mettre de la couleur par exemple, ou choisir la puce d'un *itemize*. L'extension **ENUMITEM** est faite pour cela (voir LÓPEZ, *enumitem – Control layout of itemize, enumerate, description*). Voici 2 exemples d'utilisation de la commande `\setlist` :

```
\setlist{labelindent=\parindent, font=\color{maincolor}} % toutes les listes sont indentées et
tous les items en couleur
\setlist[itemize,1]{label=\(\bullet\)} % changement de la puce des itemize de premier niveau
```

5.7 Les notes de bas de page

Pour écrire une note de bas de page, on utilise la commande `\footnote` avec en argument entre accolades, le texte de la note. La note sera affichée en page de la page et sera numérotée automatiquement⁴.

4. La numérotation est globale au chapitre.

Si l'on souhaite dissocier le texte de la note de bas de page du texte auquel il se rapporte, on peut utiliser la commande `\footnotemark` pour marquer la position d'une note de bas de page, et, dès la fin de la phrase ou du paragraphe par exemple, utiliser `\footnotetext` pour définir le texte de la note correspondante.

C'est aussi avec ces 2 commandes que l'on peut définir une note commune à plusieurs mots. Un exemple avec un paragraphe de ce document qui se trouve page 32 :

```
On utilise pour cela la paire de commande \cmd{left} et \cmd{right}, chacune suivie du délimiteur
souhaité qui peut être différent entre l'ouvrant et le fermant : (, [, |, \textbackslash{,
\textbackslash|, \textbackslash{}llbracket\footnotemark{},
\textbackslash{}rrbracket\footnotemark[\value{footnote}],
\textbackslash{}VERT\footnotemark[\value{footnote}], \ldots. \footnotetext {disponibles avec
l'extension \package{fourier}} On ne peut utiliser \cmd{left} sans \cmd{right} et inversement.
```

La commande `\value` permet dans ce contexte de récupérer la valeur du compteur `footnote` pour la réutiliser.

Pour pouvoir écrire des formules mathématiques et autres symboles, on a juste besoin d'utiliser les extensions **MATHTOOLS**, qui complète et remplace l'extension historique **AMSMATH**, et **AMSSYMB**, pour les symboles mathématiques. Cette dernière n'est pas nécessaire si vous utilisez les polices **KPFFONTS**, **FOURIER-OTF**. Très peu de symboles définis dans l'extension **AMSSYMB** n'ont pas été redéfinis dans l'extension **FOURIER** : il faudra donc charger **AMSSYMB** avant **FOURIER** pour les utiliser.

6.1 Quel environnement pour les mathématiques?

Pour écrire une formule mathématique, il y a 3 environnements de base :

math pour des formules insérées dans le corps de texte. La version \TeX est de démarrer et terminer la formule mathématique par le symbole \$. Un raccourci \LaTeX plus robuste donc fortement recommandé, est d'utiliser la paire de délimiteurs `\(` et `\)`.

displaymath pour des formules hors du corps de texte. La version \TeX est de démarrer et terminer la formule mathématique par `$$` ;

equation pour un environnement numéroté. Sa version étoilée, **equation***, désactive la numérotation. On recommande cette forme étoilée à la place de l'environnement **displaymath**. Un raccourci \LaTeX de **equation***, fortement recommandé, est d'utiliser la paire de délimiteurs `\[` et `\]`.



La police de caractère utilisée pour les formules mathématiques est différente de la police du texte. Si l'on souhaite écrire du texte avec la police du texte standard à l'intérieur des environnements **math** et **displaymath**, on utilisera la commande `\text` (ou `\mbox`).

6.2 Opérations usuelles

6.2.1 Les fractions

Pour écrire une fraction, on utilise la commande `\frac` prenant en arguments le numérateur, puis le dénominateur.

Exemple de `\frac{x+1}{y-1}` en mode **displaymath**, puis en mode **math** :

$$\frac{x+1}{y-1} \quad \frac{x+1}{y-1}$$

6.2.2 Exposants et indices

Pour définir un indice, on utilise le caractère `_` et pour définir un exposant, on utilise le caractère `^`. Si l'exposant ou l'indice comporte 2 caractères ou plus, il faudra utiliser des accolades. De même, on peut utiliser des accolades pour indiquer ce sur quoi s'applique l'indice ou l'exposant.

Exemple de `\{(x_1+y_{12})\}^2=x_1^2+2xy_{12}+\{y_{12}\}^2` : $(x_1 + y_{12})^2 = x_1^2 + 2xy_{12} + y_{12}^2$

6.2.3 Fonctions et opérateurs usuels et délimiteurs

TEX fournit la plupart des symboles de fonctions et opérateurs usuels, comme les fonctions trigonométriques, les fonctions logarithmiques, ... :

cos	<code>\cos</code>	cosh	<code>\cosh</code>	sin	<code>\sin</code>	sinh	<code>\sinh</code>	tan	<code>\tan</code>	tanh	<code>\tanh</code>
arccos	<code>\arccos</code>	arcsin	<code>\arcsin</code>	arctan	<code>\arctan</code>	cot	<code>\cot</code>	coth	<code>\coth</code>		
exp	<code>\exp</code>	ln	<code>\ln</code>	log	<code>\log</code>						
arg	<code>\arg</code>	gcd	<code>\gcd</code>	det	<code>\det</code>	dim	<code>\dim</code>	lim	<code>\lim</code>		
inf	<code>\inf</code>	sup	<code>\sup</code>	min	<code>\min</code>	max	<code>\max</code>				

TABLE 6.1 – Fonctions et opérateurs usuels

A côté de cela, il ne faut pas oublier la commande `\sqrt`, qui peut prendre un argument optionnel pour l'ordre de la racine : $\sqrt{2}$

$$\sqrt[3]{x^2 + \exp y - 1} \quad \sqrt[3]{x^2 + \exp y - 1}$$

Enfin, nous allons voir les délimiteurs. Les délimiteurs sont des paires de symboles dont la taille s'adapte à ce qu'ils englobent. Cela concerne les parenthèses, les accolades, les crochets; ...

On utilise pour cela la paire de commande `\left` et `\right`, chacune suivie du délimiteur souhaité qui peut être différent entre l'ouvrant et le fermant : (,), [,], |, \{, \}, \|, \lbracket¹, \rbracket¹, \VERT¹, On ne peut utiliser `\left` sans `\right` et inversement. Si l'on ne souhaite mettre un délimiteur que d'un côté, on utilisera le « . » de l'autre. Regardez la différence entre les 3 écritures d'une même formule dans l'exemple suivant :

Exemple 6.1 Utilisation des délimiteurs

```
\[
[{\{(x^2+y^2)}^2}] = \left[{\{(x^2+y^2)}^2\right] = \left[{\left\{\left\{x^2+y^2\right\}\right\}^2\right]
\]
```

$$[(x^2 + y^2)^2] = [(x^2 + y^2)^2] = [(x^2 + y^2)^2]$$

6.3 Variantes de polices

On a vu qu'il existe 2 modes pour les formules mathématiques, et que le mode *displaymath* les écrivait de manière plus espacées et plus agréables. De même, les indices et les exposants sont écrits plus petits. On peut forcer la taille à celle de l'un de ces éléments.

`\displaystyle` permet dans un environnement *math* de forcer l'apparence de l'environnement *displaymath*

`\textstyle` permet dans un environnement *displaymath* de forcer l'apparence de l'environnement *math*

`\scriptstyle` permet de forcer la taille à celle d'un exposant

`\scriptscriptstyle` permet de forcer la taille à celle d'un exposant d'exposant

Par ailleurs, nous avons vu précédemment que l'on pouvait changer le style des caractères du texte. On peut aussi le faire dans les formules mathématiques :

1. disponibles avec les extensions `FOURIER` et `FOURIER-OTF`

`\mathrm` pour écrire en romain $\rightarrow a \ a$
`\mathit` pour écrire en italique $\rightarrow a \ a$
`\mathbf` pour écrire en gras $\rightarrow a \ \mathbf{a}$
`\mathsf` pour écrire sans empattements $\rightarrow a \ a$
`\mathtt` pour écrire à chasse fixe $\rightarrow a \ a$
`\mathcal` pour écrire en calligraphique $\rightarrow C \ \mathcal{C}$
`\mathscr` pour écrire en script $\rightarrow C \ \mathcal{C}^2$
`\mathbb` pour écrire en notation ensembliste $\rightarrow R \ \mathbb{R}$
`\mathfrak` pour écrire en gothique $\rightarrow C \ \mathfrak{C}$

Hormis `\mathscr`, ces variantes sont disponibles avec l'extension `AMSFONTS` automatiquement chargée par défaut. Elles sont aussi toutes redéfinies dans les extensions `KPFFONTS`, `KPFFONTS-OTF`, `FOURIER` et `FOURIER-OTF`. Par contre, `\mathcal` fourni par les extensions `FOURIER` et `FOURIER-OTF` est identique à `\mathscr`.

Les symboles de fonctions (vus dans le tableau 6.1) sont tous écrits en romain. Il en est de même par défaut sur tout ce qui se trouve en indice.

L'extension `UNICODE-MATH`, utilisable uniquement avec les moteurs XeTeX et LuaTeX, fournit des commandes supplémentaires (liées à l'utilisation correcte des plices mathématiques avec le moteur LuaTeX), dont voici les principales :

`\symup` pour écrire en romain $\rightarrow a \ a$
`\symit` pour écrire en italique $\rightarrow a \ a$
`\symbf` pour écrire en gras $\rightarrow a \ \mathbf{a}$
`\symsf` pour écrire sans empattement $\rightarrow a \ a$
`\symtt` pour écrire à chasse fixe $\rightarrow a \ a$
`\symcal` pour écrire en calligraphique $\rightarrow C \ \mathcal{C}$
`\symscr` pour écrire en script $\rightarrow C \ \mathcal{C}$
`\symbb` pour écrire en notation ensembliste $\rightarrow R \ \mathbb{R}$
`\symfrak` pour écrire en gothique $\rightarrow C \ \mathfrak{C}$

6.4 Symboles mathématiques

L'extension de base pour utiliser les symboles mathématiques est `AMSSYMB`. Néanmoins, il est inutile de l'inclure si vous utilisez `KPFFONTS` ou `FOURIER-OTF`. Par ailleurs, si vous utilisez `FOURIER` et que vous avez besoin de `AMSSYMB` pour les quelques symboles qui n'ont pas (encore) été redéfinis, incluez ce dernier avant `FOURIER`.

Je vais vous donner ici les commandes permettant d'utiliser les symboles mathématiques les plus courants. Pour une liste assez exhaustive, voir PAKIN, *The Comprehensive L^AT_EX Symbol List – Symbols accessible from L^AT_EX*.

6.4.1 Lettres grecques

Pour les déclarations de variables, on utilise souvent les lettres grecques. La tableau 6.2 liste les commandes de toutes les lettres grecques.

6.4.2 Symboles d'opérateurs et de relations

Certains symboles sont de taille variable suivant que l'on est en mode `math` ou `displaymath` (voir tableau 6.5). C'est notamment le cas des symboles de somme (d'une série) et d'intégrale :

2. Disponible avec les extensions `MATHRSFS`, `KPFFONTS`, `KPFFONTS-OTF`, `FOURIER` et `FOURIER-OTF`

α	<code>\alpha</code>	ζ	<code>\zeta</code>	λ	<code>\lambda</code>	ρ	<code>\rho</code>	ϕ	<code>\phi</code>
β	<code>\beta</code>	η	<code>\eta</code>	μ	<code>\mu</code>	ϱ	<code>\varrho</code>	φ	<code>\varphi</code>
γ	<code>\gamma</code>	θ	<code>\theta</code>	ν	<code>\nu</code>	σ	<code>\sigma</code>	χ	<code>\chi</code>
δ	<code>\delta</code>	ϑ	<code>\vartheta</code>	ξ	<code>\xi</code>	ς	<code>\varsigma</code>	ψ	<code>\psi</code>
ϵ	<code>\epsilon</code>	ι	<code>\iota</code>	π	<code>\pi</code>	τ	<code>\tau</code>	ω	<code>\omega</code>
ε	<code>\varepsilon</code>	κ	<code>\kappa</code>	ϖ	<code>\varpi</code>	υ	<code>\upsilon</code>		

TABLE 6.2 – Les lettres grecques minuscules

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

TABLE 6.3 – Les lettres grecques majuscules

\pm	<code>\pm</code>	\times	<code>\times</code>	\div	<code>\div</code>	$*$	<code>\ast</code>
\star	<code>\star</code>	\circ	<code>\circ</code>	\bullet	<code>\bullet</code>	\cdot	<code>\cdot</code>
\cap	<code>\cap</code>	\cup	<code>\cup</code>	\oplus	<code>\oplus</code>		

TABLE 6.4 – Quelques symboles d'opérateurs

$\cap\cap$	<code>\bigcap</code>	$\Sigma\Sigma$	<code>\sum</code>	$\int\int$	<code>\int</code>
$\cup\cup$	<code>\bigcup</code>	$\prod\prod$	<code>\prod</code>	$\iint\iiint$	<code>\iint</code>
$\oplus\oplus$	<code>\bigoplus</code>	$\oint\oint$	<code>\oint</code>	$\iiint\iiint$	<code>\iiint</code>

TABLE 6.5 – Quelques symboles d'opérateurs de taille variable

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\subset	<code>\subset</code>	\supset	<code>\supset</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\neq	<code>\neq</code>	\perp	<code>\perp</code>
\propto	<code>\propto</code>	\sim	<code>\sim</code>	\simeq	<code>\simeq</code>	\approx	<code>\approx</code>

TABLE 6.6 – Quelques symboles de relations

Les symboles de relations peuvent être précédés de `\not` pour obtenir leurs opposés.

6.4.3 Symboles divers et flèches

Pour terminer ce tour rapide sur les symboles mathématiques, quelques symboles supplémentaires, et surtout les flèches.

\Re	<code>\Re</code>	\Im	<code>\Im</code>	\forall	<code>\forall</code>	\exists	<code>\exists</code>
∂	<code>\partial</code>	∞	<code>\infty</code>				

TABLE 6.7 – Quelques symboles divers

6.4.4 Les accents

Il arrive fréquemment que l'on utilise des signes au dessus ou en dessous de la lettre définissant une variable, ce que l'on appelle des accents.

6.5. ÉQUATIONS ET SYSTÈMES D'ÉQUATIONS

\leftarrow	<code>\leftarrow</code>	\rightarrow	<code>\rightarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>		

TABLE 6.8 – Quelques flèches

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\breve{a}	<code>\breve{a}</code>	\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>
\acute{a}	<code>\acute{a}</code>	\grave{a}	<code>\grave{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>

TABLE 6.9 – Les accents mathématiques

Toutefois, ces accents sont nécessairement de la largeur d'un caractère moyen. Pour définir des accents qui s'adaptent en largeur, voici la liste des plus courants :

\widehat{AB}	<code>\widehat{AB}</code>	\overline{AB}	<code>\overline{AB}</code>	\overrightarrow{AB}	<code>\overrightarrow{AB}</code>
\widetilde{AB}	<code>\widetilde{AB}</code>	\underline{AB}	<code>\underline{AB}</code>	\overleftarrow{AB}	<code>\overleftarrow{AB}</code>
\overbrace{AB}	<code>\overbrace{AB}</code>	\underbrace{AB}	<code>\underbrace{AB}</code>		

TABLE 6.10 – Les accents mathématiques de taille variable



La commande `\overbrace` accepte un commentaire en guise d'exposant. La commande `\underbrace` accepte un commentaire en guise d'indice.

Exemple 6.2 Ajouter des commentaires avec `\overbrace` ou `\underbrace`

```
\[
x^p = \overbrace{x \times x \times \cdots \times x \times x}^{p \text{ fois}}
      = \underbrace{x \times x \times \cdots \times x \times x}_{p \text{ fois}}
\]
```

$$x^p = \overbrace{x \times x \times \cdots \times x \times x}^{p \text{ fois}} = \underbrace{x \times x \times \cdots \times x \times x}_{p \text{ fois}}$$

6.5 Équations et systèmes d'équations

Le seul environnement évoqué jusqu'à présent permettant de numéroté une formule mathématique est l'environnement `equation`. Il est temps ici de montrer comment il fonctionne.

Exemple 6.3 L'environnement `equation`

```
\begin{equation}
\{(x+y)\}^2 = x^2 + 2 x y + y^2
\end{equation}
```

$$(x + y)^2 = x^2 + 2xy + y^2 \quad (6.1)$$

6.5. ÉQUATIONS ET SYSTÈMES D'ÉQUATIONS

Un numéro automatique est attribué et est affiché à droite.

On peut également souhaiter numérotter une suite d'équation qui font partie d'un même calcul en alignant par exemple certains éléments, comme le symbole =. On utilise pour cela l'environnement *align*.

Exemple 6.4 L'environnement *align*

```
\begin{align}
{(2 x+y)}^2 &= {(2 x)}^2 + 2 (2 x) y + y^2 \\
&= 4 x^2 + 4 x y + y^2
\end{align}
```

$$(2x + y)^2 = (2x)^2 + 2(2x)y + y^2 \quad (6.2)$$

$$= 4x^2 + 4xy + y^2 \quad (6.3)$$

A l'intérieur de cet environnement, on insère des symboles & pour signifier un alignement et \\ pour passer à la ligne. L'alignement des colonnes est alterné entre un alignement à gauche pour les colonnes impaires et un alignement à droite pour les colonnes paires.



On peut désactiver la numérotation pour certaines lignes en insérant la commande `\nonumber`, de préférence en fin de ligne pour la lisibilité.

Exemple 6.5 Utilisation de `\nonumber` dans un environnement *align*

```
\begin{align}
{(2 x+y)}^2 &= {(2 x)}^2 + 2 (2 x) y + y^2 \nonumber \\
&= 4 x^2 + 4 x y + y^2
\end{align}
```

$$(2x + y)^2 = (2x)^2 + 2(2x)y + y^2$$
$$= 4x^2 + 4xy + y^2 \quad (6.4)$$



Il existe une version étoilée qui désactive complètement la numérotation.

Les environnements *equation* et *align* sont du même niveau que les environnements *math* et *displaymath*. Il existe toutefois un équivalent interne à l'environnement *align** : *aligned*. Ce qui signifie que l'on peut le mettre autour d'une paire de délimiteurs par exemple, ou attribuer un numéro unique à l'ensemble, suivant dans quel environnement on l'utilise parmi *math*, *displaymath* ou *equation*.

Exemple 6.6 *L'environnement aligned*

```

\begin{equation}
\left\{
\begin{aligned}
(2x+y)^2 &= (2x)^2 + 2(2x)y + y^2 \\
(2x+y)^2 &= 4x^2 + 4xy + y^2
\end{aligned}
\right.
\end{equation}

```

$$\left\{ \begin{array}{l} (2x+y)^2 = (2x)^2 + 2(2x)y + y^2 \\ (2x+y)^2 = 4x^2 + 4xy + y^2 \end{array} \right. \quad (6.5)$$

Il existe un autre environnement pour obtenir un résultat similaire, dans le cas **uniquement d'un unique alignement par ligne (un seul & par ligne)** : *cases*.

Exemple 6.7 *L'environnement cases pour aligner des équations*

```

\left[
\begin{cases}
(2x+y)^2 &= (2x)^2 + 2(2x)y + y^2 \\
(2x+y)^2 &= 4x^2 + 4xy + y^2
\end{cases}
\right]

```

$$\left\{ \begin{array}{l} (2x+y)^2 = (2x)^2 + 2(2x)y + y^2 \\ (2x+y)^2 = 4x^2 + 4xy + y^2 \end{array} \right.$$

On constate une légère différence au niveau de l'espacement avant le symbole aligné : =. Cela nous ramène encore à la philosophie de \LaTeX , selon laquelle on distingue le rendu visuel de la structure du document. *cases* est dédié à écrire une alternative, avec éventuellement un texte qui suit que l'on souhaitera aligner, tandis que *aligned* est, comme *align* dédié à l'alignement d'équations. On pourra par exemple utiliser *aligned* pour définir un système d'équations dans lequel effectivement on alignera le symbole =.

Exemple 6.8 *L'environnement cases pour écrire une alternative*

```

\left[
\begin{cases}
-\Delta u = f & \text{in } \Omega \\
u = 0 & \text{on } \Gamma = \partial\Omega
\end{cases}
\right]

```

$$\left\{ \begin{array}{l} -\Delta u = f \quad \text{in } \Omega \\ u = 0 \quad \text{on } \Gamma = \partial\Omega \end{array} \right.$$



Chaque ligne d'un environnement *cases* est en mode *math*. Pour éviter d'utiliser `\displaystyle` à chaque ligne, on peut utiliser l'environnement *dcases* qui le fait par défaut.

6.6 Théorèmes

Quand on rédige un document scientifique, on peut certes vouloir écrire des formules mathématiques, mais on peut aussi vouloir définir des théorèmes. C'est là le rôle de l'extension **AMSTHM** dont je vais vous parler maintenant.

Pour définir un théorème, on utilise la commande `\newtheorem` :

```
\newtheorem{theorem}{Théorème}[chapter]
```

3 arguments obligatoires : le nom de l'environnement créé, son label et le niveau de numérotation. On dispose alors d'un environnement *theorem* (dans cet exemple), ayant un argument optionnel (pour préciser le nom du théorème ou son auteur par exemple), ce qui donne une fois utilisé un résultat de la forme :

Théorème 6.6.1 (Pythagore). *Dans un triangle rectangle, le carré de l'hypoténuse est égal à la somme des carrés des deux autres cotés.*

Vous pouvez aussi partager la numérotation entre 2 environnements de type théorèmes, en procédant comme suit :

```
\newtheorem{theorem}{Théorème}[chapter]
\newtheorem{lemma}[theorem]{Lemme}
```

Pour les démonstrations, l'extension **AMSTHM** fournit l'environnement *proof*.

Il existent d'autres extensions permettant de définir des environnements de type théorèmes. Les plus intéressantes sont **THMTOOLS** et surtout **TCOLORBOX** (voir section 9.3).

7.1 Inclusion d'images

Pour inclure une image dans un document \TeX , on utilise la commande `\includegraphics` (fournie par l'extension `GRAPHICX`).

On précise la largeur et / ou la hauteur à l'aide des options respectives `width` et `height`

Exemple 7.1 Inclure une image

```
\includegraphics [width=2cm]{images/icons/focus.png}
\includegraphics [height=2em]{images/icons/focus.png}
\includegraphics [width=2cm, height=2em]{images/icons/focus.png}
```



Quand on compile avec $\text{pdf}\text{\TeX}$, $\text{Xe}\text{\TeX}$ ou $\text{Lua}\text{\TeX}$, les formats d'image autorisées sont jpg, jpeg, png, pdf. Par ailleurs, préciser l'extension n'est pas nécessaire en général.

7.2 Tableaux et matrices

Pour définir un tableau dans le corps de texte, on utilise l'environnement `tabular`.

Exemple 7.2 Notre premier tableau

```
\begin{tabular}{ccc}
a & b & c \\
d & e & f \\
g & & \\
\end{tabular}
```

a	b	c
d	e	f
g		

Analysons l'exemple 7.2.

- Le premier argument de `tabular` définit à la fois le nombre de colonnes, mais aussi l'alignement du texte dans chaque cellule. Essentiellement 4 valeurs sont possibles :
 - l** pour une colonne de largeur automatique dans laquelle le texte sera aligné à gauche
 - r** pour une colonne de largeur automatique dans laquelle le texte sera aligné à droite
 - c** pour une colonne de largeur automatique dans laquelle le texte sera centré
 - p{}** pour une colonne de taille donnée entre accolades dans laquelle le texte sera justifié
- Chaque cellule d'une ligne est séparée par un `&`

- A la fin de chaque ligne de tableau, on utilise `\\`. On peut terminer une ligne prématurément, auquel cas les cellules restantes sont considérées vides.

On constate que le tableau résultant ne contient aucune bordure. Pour les rajouter, regardons l'exemple suivant :

Exemple 7.3 Tableaux et bordures pleines

```
\begin{tabular}{|p{1cm}||c|c|}
\hline
& aa & bb \\
\hline
\hline
xx & c & d \\
\hline
yy \\
\hline
zz & & \\
\hline
\end{tabular}
```

	aa	bb
xx	c	d
yy		
zz		

Que s'est-il passé?

- On a inséré dans le premier argument de `tabular` des `|` pour chaque bordure verticale souhaitée.
- On a inséré `\hline` pour chaque bordure horizontale souhaitée.
- On peut doubler ces symboles pour avoir des bordures doubles
- Si une ligne du tableau se termine prématurément, les bordures horizontales ne sont pas affichées pour cette ligne¹. La seule solution est de définir les cellules correspondantes mais de n'y mettre aucun contenu.

Pour la première cellule du tableau de l'exemple 7.3, on souhaiterait que les bordures supérieures et gauche ne soient pas tracées. Pour cela, on peut procéder comme suit :

Exemple 7.4 Tableaux et bordures partielles

```
\begin{tabular}{|p{1cm}|c|c|}
\cline{2-3}
\multicolumn{1}{|l|}{} & \multicolumn{2}{c}{aa} \\
\hline
xx & c & d \\
\hline
yy & e & f \\
\hline
\end{tabular}
```

	aa	
xx	c	d
yy	e	f

- La commande `\cline` permet de définir la longueur de la bordure horizontale à tracer en lui donnant la colonne de départ et la colonne d'arrivée.
- La commande `\multicolumn` permet de fusionner des cellules. Elle prend 3 arguments :
 1. La largeur de la cellule en nombre de colonnes;
 2. L'alignement du texte dans la cellule;
 3. Le contenu de la cellule.
- On peut en particulier utiliser `\multicolumn` pour effacer la bordure de gauche de la première cellule.

1. Elles s'affichent si vous utilisez la classe `beamer`

Nous avons parlé jusqu'ici de l'environnement *tabular* qui fonctionne dans le corps de texte, et pas en mode mathématique. Dans ce cas là, on utilise l'environnement *array*, de la même manière :

Exemple 7.5 Tableaux mathématiques

```
\[
\begin{array}{|c|c|c|}
\cline{2-3}
\multicolumn{1}{|1|}{} & \multicolumn{2}{|c|}{aa} \\
\hline
xx & c & d \\
\hline
yy & e & f \\
\hline
\end{array}
\]
```

	aa	
xx	c	d
yy	e	f



Pour fusionner des cellules suivant les lignes, vous devez charger une extension supplémentaire, **MULTIROW** (voir OOSTRUM et LEICHTER, *multirow – Create tabular cells spanning multiple rows*), qui fournit une commande `\multirow` qui a la même syntaxe que `\multicolumn`.

7.3 Les environnements flottants

On peut souhaiter écrire une légende à une image ou un tableau. On utilise pour cela les environnements *figure* (pour les images) et *table* (pour les tableaux).

Ces deux environnements fonctionnent de la même manière :

- Un numéro de figure ou de tableau est affecté.
- Ces 2 environnements sont dits « flottants » car leur positionnement dépend du corps de texte autour. On peut contrôler le positionnement avec un argument optionnel qui peut comprendre toute séquence composée d'un à 5 caractères parmi :
 - h** Si possible, la figure est insérée dans le corps de texte à l'endroit où l'environnement flottant est défini
 - t** Si possible, la figure est insérée en haut de la page contenant l'endroit où l'environnement est défini ou la page suivante
 - b** Si possible, la figure est insérée en bas de la page contenant l'endroit où l'environnement est défini ou la page suivante
 - p** La figure est insérée sur une page spéciale en contenant que des environnements flottants.
- ! Insister pour que \LaTeX respecte l'ordre indiqué par la séquence des autres caractères.



Il se peut que \LaTeX ne puisse pas accéder à notre demande de positionnement, même avec le point d'exclamation. On peut contourner la difficulté en chargeant l'extension **FLOAT** qui nous donne une possibilité supplémentaire de placement : **H**, qui force à mettre l'environnement flottant là où on l'a défini, quitte à ajouter un saut de page, soit avant, soit après l'environnement flottant.

Exemple 7.6 Tableau flottant

```

\begin{table}[H]
\centering
\begin{tabular}{|c|c|c|}
\cline{2-3}
\multicolumn{1}{|1|}{} & \multicolumn{2}{c|}{aa} \\
\hline
xx & c & d \\
\hline
yy & e & f \\
\hline
\end{tabular}
\caption{Exemple de tableau flottant}
\end{table}

```

	aa	
xx	c	d
yy	e	f

TABLE 7.1 – Exemple de tableau flottant

La commande `\caption`, qui permet de spécifier une légende, et qui porte la numérotation de l'environnement flottant, peut être utilisée soit au début de l'environnement (elle sera affichée avant le contenu), soit à la fin (elle sera donc affichée après le contenu).

On peut lister les figures utilisées dans un document et les afficher, sur le même principe que la table des matières est une liste de titre et de sous-titres.

Pour afficher cette liste de figures, on utilise la commande `\listoffigures`.

De même on peut lister les tables à l'aide de la commande `\listoftables`.

C'est ainsi que sont construites ces listes en pages 5 et 6 de ce document.

7.4 Les listings de codes

Pour écrire du listing de code, on peut utiliser l'environnement *verbatim* ou la commande `\verb`.

La commande `\verb` a une utilisation particulière parce que ce qu'elle prend en argument n'est pas entre accolades, mais entre une paire de caractères identiques non alphanumériques comme `|,!,?, =, ...`

Exemple 7.7 Utilisation de la commande `\verb`

```

\verb?\texte?

```

```

\ttexte

```

Si l'on souhaite toutefois que le listing de code soit mis en forme, il est hautement recommandé d'utiliser l'extension `LISTINGS`. Commençons par l'environnement *lstlisting*, le plus courant :

```

\begin{lstlisting}[language=C++]
int main(int argc, char* argv[]);
\end{lstlisting}

```

```

int main(int argc, char* argv[]);

```

Les possibilités de personnalisation de l'affichage d'un listings sont énormes. On peut par exemple obtenir facilement ceci :

```

int main(int argc, char* argv[]);

```

Ce que cet exemple montre, c'est que l'on peut donner des mots-clé supplémentaires (main dans ce cas), les catégoriser (on parle de niveaux de mot-clés) et définir les couleurs, les marges, ..., que l'on veut. Cela veut donc dire que l'on peut créer une colorisation syntaxique pour un langage qui n'est pas prévu par l'extension (et elle gère déjà une trentaine de langages de programmation différents).

A côté de l'environnement *lstlisting*, on dispose également de la commande `\lstinline` qui fonctionne de la même manière, mais qui va insérer le listing dans le texte courant plutôt que de l'afficher à la ligne dans une boîte à part.

Il y a enfin la commande `\lstinputlisting` qui va permettre de lire le contenu d'un fichier. Cette commande a en particulier 2 options très pratiques qui permettent de définir une plage de lignes à afficher :

firstline pour définir à partir de quelle ligne afficher le contenu d'un fichier. Par défaut, c'est la première ligne du fichier.

lastline pour définir jusqu'à quelle ligne afficher le contenu d'un fichier. Par défaut, c'est la dernière ligne du fichier

Maintenant que vous savez définir des tableaux, vous êtes capables de composer vous-même votre CV. Vous pouvez aussi utiliser une classe qui s'appelle *moderncv* (voir DANAUX, *moderncv – A modern curriculum vitae class*).



Il existe également 2 autres classes, qui ne seront pas détaillées ici : *curve* (voir VERNA, *CurVe – A class for making curriculum vitae*) et *currvita* (voir REICHERT, *currvita – Typeset a curriculum vitae*)

```
\documentclass[11pt, a4paper, french]{moderncv}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{fourier}
\usepackage{geometry}
\usepackage{babel}
\usepackage{microtype}

% modification des dimensions de la page et de son centrage %
\geometry{top=2.5cm, bottom=2.5cm, left=1.8cm, right=1.8cm}

\moderncvtheme{classic}

\name{Harry}{Potter}
\title{Directeur du Département de la Justice Magique}
\address{12, Square Grimmaurd}{Londres}
\phone[mobile]{06 05 04 03 02}
\phone[fixed]{01 02 03 04 05}
\email{potter@mjd.com}
\extrainfo{Marié, 3 enfants}
\photo{../images/potter.png}

\quote{Expelliarmus}
```

Les noms des commandes sont suffisamment clairs :

\name prend le prénom en premier argument et le nom en deuxième argument ;

\address prend le numéro et la rue en premier argument, le code postal et la ville en second argument ;

\phone prend un seul argument, le numéro, et a un argument optionnel pour préciser le type de numéro : *mobile, fixed, fax* ;

\email prend en seul et unique argument l'adresse e-mail ;

\photo prend en seul et unique argument le chemin vers le fichier d'une photo ;

\quote prend en seul et unique argument une citation ;

\homepage prend en seul et unique argument l'adresse d'un site web ;

\social prend un seul et unique argument, un identifiant, et a un argument optionnel pour préciser à quel réseau social l'identifiant correspond : *twitter, linkedin, github*.

Regardons maintenant le corps du document :

```
\makecvtitle

\section{Expérience professionnelle}

\cventry{2007}{Directeur}{Bureau des Aurors}{Londres}{}{}
\cventry{1998}{Auror}{}
{Londres}{}
{ Suite à la victoire contre Lord Voldemort, il entre au Bureau des Aurors }
\cventry{1994 -- 1995}{Vainqueur}{Tournoi des Trois Sorciers}{Poudlard}{}{
\begin{itemize}
\item Subtiliser un \oe{}uf d'or à un dragon;
\item Nager sous l'eau pendant une heure à la recherche d'un trésor qui lui est cher;
\item S'emparer du trophée dans un labyrinthe truffé de créatures dangereuses.
\end{itemize}
}

\section{Références}

\subsection{Ministère de la Magie}

\cvlistdoubleitem{Hermione Granger}{Kingsley Shacklebolt}

\subsection{Poudlard}

\cvlistdoubleitem{Dir. Minerva McGonagall}{Pr. Rubeus Hagrid}

\cvlistitem{Pr. Neville Londubat}

\section{Formation}

\cventry{1991 -- 1997}{\{'E}lève}{Gryffondor}{Poudlard}{}{
Obtient 7 BUSE:
\begin{description}
\item[Optimal] Défense contre les Forces du Mal
\item[Effort exceptionnel] Métamorphose, Sortilèges, Potions, Botanique, Soins aux Créatures Magiques
\item[Acceptable] Astronomie
\end{description}
}

\section{Langues étrangères}

\cvitemwithcomment{Fourchelangue}{courant}{jusqu'en 1998}

\section{Activités extra-professionnelles}

\cvitem{Qwidditch}{Capitaine de l'équipe de Gryffondor de 1996 à 1997}
\cvitem{Association}{Enseignant dans "L'Armée de Dumbledore"}
\cvitemwithcomment{Promenades}{surtout la nuit}{avec la Carte du Maraudeur et la Cape d'Invisibilité}
```

\makecvtitle pour afficher les informations personnelles;

\section pour définir les rubriques principales d'un CV : Expérience, Formation, ...

\subsection pour éventuellement définir des sous-rubriques;

\cventry pour écrire une entrée dans le CV. Cette commande prend 6 arguments obligatoires :

- Le premier argument est toujours supposé être la date ou la période temporelle

- Pour une formation, les arguments suivants peuvent être dans l'ordre : le diplôme, l'institution, la ville, la mention et une description
- Pour une expérience professionnelle, les arguments 2 à 4 peuvent être dans l'ordre : l'intitulé du poste, l'employeur, et la ville; le 5ème argument est laissé vide et le dernier contient la description du poste
- En général, les 3 premiers arguments doivent être remplis et les 3 suivants peuvent être laissés vides.

\cvitem prend 2 arguments : le type d'information, et sa description;

\cvitemwithcomment joue le même rôle que ma commande **\cvitem**, mais a un 3ème argument, destiné à être un commentaire;

\cvlistitem prend un seul argument l'information, en la faisant précéder d'une puce;

\cvlistdoubleitem joue le même rôle que la commande **\cvlistitem** mais avec 2 informations, présentées sur 2 colonnes;

Pour terminer, je reviens sur le style du CV :

```
\moderncvtheme{classic}
```

Cette commande permet de choisir le thème du CV. La valeur de l'argument peut être **classic** (voir figure 8.1), **banking**, **casual** ou **oldstyle** (voir figure 8.2). Et en guise d'option, on peut choisir la couleur du thème.

The figure displays two versions of a CV for Harry Potter, generated using the `\moderncvtheme{classic}` command. The left version uses the default blue color scheme, while the right version uses a green color scheme. Both CVs contain the following information:

- Header:** Harry Potter, Directeur du Département de la Justice Magique, Expelliarmus. Contact info: Square Grimmaurd 12, 06 05 04 03 02, 01 02 03 04 05, potter@mj.com, Marié, 3 enfants.
- Expérience professionnelle:**
 - 2007 **Directeur, Bureau des Aurors**, Londres.
 - 1998 **Auror**, Londres. Suite à la victoire contre Lord Voldemort, il entre au Bureau des Aurors
 - 1994 – 1995 **Vainqueur, Tournoi des Trois Sorciers**, Poudlard.
 - Subtiliser un œuf d'or à un dragon;
 - Nager sous l'eau pendant une heure à la recherche d'un trésor qui lui est cher;
 - S'emparer du trophée dans un labyrinthe truffé de créatures dangereuses.
- Références:**
 - Ministère de la Magie — Kingsley Shacklebolt
 - Poudlard — Dir. Minerva McGonagall — Pr. Rubeus Hagrid
- Formation:**
 - 1991 – 1997 **Élève, Gryffondor**, Poudlard. Obtient 7 BUSE :
 - Optimal Défense contre les Forces du Mal
 - Effort exceptionnel Métamorphose, Sortilèges, Potions, Botanique, Soins aux Créatures Magiques
 - Acceptable Astronomie
- Langues étrangères:** Fourchelangue courant jusqu'en 1998
- Activités extra-professionnelles:**
 - Quidditch Capitaine de l'équipe de Gryffondor de 1996 à 1997
 - Association Enseignant dans "L'Armée de Dumbledore"
 - Promenades surtout la nuit avec la Carte du Maraudeur et la Cape d'Invisibilité

FIGURE 8.1 – Le thème **classic** de la classe **moderncv** avec la couleur par défaut à gauche, et le choix du vert à droite

Harry Potter | Directeur du Département de la Justice Magique

12 - Square Grimmaurd

☎ 06 05 04 03 02 • 📞 01 02 03 04 05 • ✉ potter@mjd.com • Marié, 3 enfants

Expelliarmus

Expérience professionnelle

Bureau des Aurors **Londres**
Directeur **2007**

Londres
1998

Auror

Suite à la victoire contre Lord Voldemort, il entre au Bureau des Aurors

Tournoi des Trois Sorciers **Poudlard**
Vainqueur **1994 - 1995**

- Subtiliser un œuf d'or à un dragon;
- Nager sous l'eau pendant une heure à la recherche d'un trésor qui lui est cher;
- S'emparer du trophée dans un labyrinthe truffé de créatures dangereuses.

Références

Ministère de la Magie
— Hermione Granger — Kingsley Shacklebolt

Poudlard
— Dir. Minerva McGonagall — Pr. Rubeus Hagrid
— Pr. Neville Londubat

Formation

Gryffondor **Poudlard**
Élève **1991 - 1997**

Obtient 7 BUSE :

- Optimal** Défense contre les Forces du Mal
- Effort exceptionnel** Métamorphose, Sortilèges, Potions, Botanique, Soins aux Créatures Magiques
- Acceptable** Astronomie

Langues étrangères

Fourchelange: courant *jusqu'en 1998*

Activités extra-professionnelles

Quidditch: Capitaine de l'équipe de Gryffondor de 1996 à 1997

Association: Enseignant dans "L'Armée de Dumbledore"

Promenades: surtout la nuit *avec la Carte du Maraudeur et la Cape d'Invisibilité*



Harry Potter

Directeur du Département de la Justice Magique

Expelliarmus

Expérience professionnelle

2007 **Directeur**, Bureau des Aurors, Londres.

1998 **Auror**, Londres.
Suite à la victoire contre Lord Voldemort, il entre au Bureau des Aurors

1994 - 1995 **Vainqueur**, Tournoi des Trois Sorciers, Poudlard.

- Subtiliser un œuf d'or à un dragon;
- Nager sous l'eau pendant une heure à la recherche d'un trésor qui lui est cher;
- S'emparer du trophée dans un labyrinthe truffé de créatures dangereuses.

Références

Ministère de la Magie
— Hermione Granger — Kingsley Shacklebolt

Poudlard
— Dir. Minerva McGonagall — Pr. Rubeus Hagrid
— Pr. Neville Londubat

Formation

1991 - 1997 **Élève**, Gryffondor, Poudlard.

Obtient 7 BUSE :

- Optimal** Défense contre les Forces du Mal
- Effort exceptionnel** Métamorphose, Sortilèges, Potions, Botanique, Soins aux Créatures Magiques
- Acceptable** Astronomie

Langues étrangères

Fourchelange: courant *jusqu'en 1998*

Activités extra-professionnelles

Quidditch: Capitaine de l'équipe de Gryffondor de 1996 à 1997

Association: Enseignant dans "L'Armée de Dumbledore"

12 - Square Grimmaurd

☎ 06 05 04 03 02 • 📞 01 02 03 04 05 • ✉ potter@mjd.com
Marié, 3 enfants

1/2

Harry Potter

Directeur du Département de la Justice Magique

Expelliarmus

12
Square Grimmaurd
M 06 05 04 03 02
T 01 02 03 04 05
E potter@mjd.com
Marié, 3 enfants

Expérience professionnelle

Directeur, Bureau des Aurors, Londres. **2007**

Auror, Londres. **1998**

Suite à la victoire contre Lord Voldemort, il entre au Bureau des Aurors

Vainqueur, Tournoi des Trois Sorciers, Poudlard. **1994 - 1995**

- Subtiliser un œuf d'or à un dragon;
- Nager sous l'eau pendant une heure à la recherche d'un trésor qui lui est cher;
- S'emparer du trophée dans un labyrinthe truffé de créatures dangereuses.

Références

Ministère de la Magie
— Hermione Granger — Kingsley Shacklebolt

Poudlard
— Dir. Minerva McGonagall — Pr. Rubeus Hagrid
— Pr. Neville Londubat

Formation

Élève, Gryffondor, Poudlard. **1991 - 1997**

Obtient 7 BUSE :

- Optimal** Défense contre les Forces du Mal
- Effort exceptionnel** Métamorphose, Sortilèges, Potions, Botanique, Soins aux Créatures Magiques
- Acceptable** Astronomie

Langues étrangères

Fourchelange: courant *jusqu'en 1998*

Activités extra-professionnelles

Quidditch: Capitaine de l'équipe de Gryffondor de 1996 à 1997

Association: Enseignant dans "L'Armée de Dumbledore"

Promenades: surtout la nuit *avec la Carte du Maraudeur et la Cape d'Invisibilité*



FIGURE 8.2 – Les autres thèmes de CV proposés par la classe *moderncv* : **banking**, **casual** et **oldstyle**

Rédaction avancée de documents

Partie III

9	Personnaliser son environnement \LaTeX	49
9.1	Définir de nouvelles commandes	49
9.2	Définir de nouveaux environnements	50
9.3	Définir de nouvelles boîtes de couleur avec <code>TCOLORBOX</code>	50
9.4	Modifier des commandes et environnements existants	51
9.5	Prévoir des formes étoilées pour les commandes que vous définissez	51
9.6	Personnaliser les commandes de titres	52
9.7	Gestion clé-valeur des arguments optionnels	53
9.8	Espaces parasites et commentaires	53
10	La bibliographie	54
10.1	Ecrire un fichier de bibliographie	54
10.2	Un peu d'histoire : <code>bib\TeX</code> et l'extension <code>NATBIB</code>	55
10.3	<code>biber</code> et l'extension <code>BIBLATEX</code>	56
10.4	La bibliographie en pratique	56
11	Les nomenclatures	58
11.1	Définir ses notations	58
11.2	Construire une nomenclature	58
11.3	Définir des catégories de notations	59
12	Références et liens hypertextes	60
12.1	Définir des étiquettes (=labels)	60
12.2	Définir une référence	60
13	Les tables des matières locales avec <code>MINITOC</code>	62
14	Les index	63
14.1	Génération de l'index	63
14.2	Les entrées de l'index	63
14.3	Personnaliser l'index	64
15	Annotations avec l'extension <code>TODONOTES</code>	65
15.1	Définir une annotation	65
15.2	Afficher la liste des notes	65
16	Gestion multi-fichiers	68
16.1	Séparer le corps du document en plusieurs fichiers	68
16.2	Créer une extension à partir de son préambule personnalisé	68

9.1 Définir de nouvelles commandes

Pour définir une commande, on utilise la commande `\newcommand`. 3 exemples pour comprendre comment l'utiliser :

Exemple 9.1 Définition d'une commande sans arguments

```
\newcommand{\macommande}{ Hello World }
\macommande
```

Hello World

La commande `\newcommand` prend 2 arguments obligatoires : le premier est le nom de la commande, le second est ce que fait cette commande.

Exemple 9.2 Définition d'une commande avec 2 arguments

```
\newcommand{\macommande}[2]{#1 #2}
\macommande{ Hello }{ World }
```

Hello World

Si la commande a des arguments, il faut simplement préciser le nombre d'arguments. C'est un argument optionnel de `\newcommand` (entre crochets). Pour utiliser les arguments dans la définition d'une commande on utilise le # suivi du numéro de l'argument.

Exemple 9.3 Définition d'une commande avec 2 arguments dont un optionnel

```
\newcommand{\macommande}[2][ Bonjour ]{#1 #2}
\macommande{ Le Monde }
\macommande[ Hello ]{ World }
```

Bonjour Le Monde
Hello World

L'argument optionnel est toujours considéré comme étant le premier argument. La valeur par défaut de l'argument optionnel est spécifié également entre crochets, juste après le nombre d'arguments de la commande.



Il existe un cas de figure où de telles commandes peuvent poser problème, par exemple lorsqu'on définit des commandes de changement de style et qu'on les utilise dans une commande de titre (`\chapter`, ...) dans une légende d'environnement flottant (`\caption`), ... Dans ce cas, essentiellement 2 possibilités :

- Utiliser la commande `\protect` juste avant la commande concernée ;
- Utiliser l'extension `ETOOLBOX` pour disposer de la commande `\newrobustcmd` à utiliser à la place de `\newcommand` avec les mêmes arguments.

9.2 Définir de nouveaux environnements

On peut également définir un environnement à l'aide de la commande `\newenvironment`. Elle prend 3 arguments obligatoires : le nom de l'environnement (il n'y a pas de `\`), ce qui se passe au début de l'environnement et ce qui se passe à la fin. Pour les arguments, cela fonctionne de la même façon que `\newcommand`

Exemple 9.4 Définition d'un environnement personnalisé

```
\newenvironment{monenvironnement}[2][Hello]{#1 #2}{!!}

\begin{monenvironnement}[\itshape]{Hello}
World
\end{monenvironnement}
```

Hello World!!

9.3 Définir de nouvelles boîtes de couleur avec TCOLORBOX

Il existe énormément d'extensions qui permettent de définir des boîtes, de couleur ou non. Mais il existe une extension qui permet de faire tout ce dont vous pourriez avoir envie de faire : **TCOLORBOX** (voir STURM, *tcolorbox – Coloured boxes, for L^AT_EX examples and theorems, etc*)

TCOLORBOX a pour vocation d'offrir des boîtes de couleur personnalisables dans tout type de documents, non seulement des boîtes de textes, mais aussi des boîtes de listing de code, ou des boîtes de type théorème.

Un exemple de boîte de couleur



Voici la définition de l'environnement *focusbox*, qui est la boîte actuelle avec une image de loupe :

```
\newtcolorbox{focusbox}{%
  colback=black!75!white!5!white,%
  colframe=black!75!white!75!black,%
  before upper={\lettrine[image,lines=2,findent=1ex]{focus.png}},%
  before=\vspace{0.5em}\par, after=\vspace{0.25em}\par,%
  height=4em,height plus=\textheight,%
  left=1ex,right=1ex}
```

Voici l'explication des différentes options utilisées :

colback, colframe La couleur de fond / de bord;

before, after Ajout de code autour du contenu de l'environnement (et de la définition de la police et des couleurs);

before upper Ajout de code entre la définition de la police et de la couleur, et le début du contenu. C'est cette option qui me permet de rajouter l'image de la loupe à l'aide d'une lettrine;

left, right Marges à gauche et à droite;

Un exemple de boîte de listing de code



Voici la définition de l'environnement *listingbox*, qui est la boîte utilisée pour le listing ci-dessous :

```
\newtcblisting{listingbox}{enhanced,boxrule=0pt,drop shadow=green!40!blue,sharp corners,rounded corners=southeast,colback=green!4!blue!6!white,colframe=green!40!blue,listing only,top=0pt,bottom=0pt,boxsep=0pt,listing style={main_light}}%
```

Voici l'explication des différentes options utilisées :

enhanced Un skin au sens de **TCOLORBOX**

drop shadow,sharp corners,rounded corners Le style de la bordure

listing only Permet de désactiver la compilation du listing que l'on veut afficher et l'affichage du résultat, ce qui permet de faire une démo de codes \LaTeX par exemple.

listing style Permet de définir le style du listing de code (les paramètres de l'environnement *lstlisting*)

Un exemple de théorème



Voici la définition de l'environnement *exercice*, qui est la boîte utilisée pour les activités pratiques de cette formation :

```
\newtcbtheorem{exercice}{Exercice}%
{titlerule=0pt,enhanced,drop shadow=example,sharp corners,rounded corners=southeast,
colback=white,colbacktitle=white,coltitle=black,colframe=example,fonttitle=\bfseries,label
type=exercice,breakable}{exo}
```

Voici l'explication des différentes options utilisées :

label type Utilisé pour les références croisées à l'aide de **CLEVEREF**

breakable Permet de couper le théorème sur plusieurs pages/transparents, par l'utilisation de la commande **\tcbbreak** dans le contenu de l'environnement.

exo Préfixe du label associé à l'environnement

Un tel environnement aura 2 arguments obligatoires : un titre et le suffixe du label, de sorte que le label complet sera de la forme `prefixe:suffixe`

9.4 Modifier des commandes et environnements existants

Vous avez tout à fait la possibilité de redéfinir des commandes ou des environnements existants. Pour cela il suffit d'utiliser les commandes **\renewcommand** et **\renewenvironment** de la même manière que leurs homologues respectifs **\newcommand** et **\newenvironment**.

Il y a toutefois une contrainte à respecter : la signature d'origine de l'environnement ou de la commande que vous redéfinissez.

Une autre possibilité est de compléter une commande existante à l'aide des fonctionnalités de l'extension **ETOOBOX**.

9.5 Prévoir des formes étoilées pour les commandes que vous définissez

Voici la définition de la commande **\cmd**, avec une version étoilée pour désactiver la couleur :

```

\makeatletter
\newcommand{\cmd}{%
\@ifstar{\cmdNoColored}{\cmdColored}%
}%
\makeatother
\newcommand{\cmdNoColored}[1]{\textbackslash #1}
\newcommand{\cmdColored}[1]{\structure{\cmdNoColored{#1}}}

```

`\@ifstar` permet de débrancher entre les versions classique ou étoilée.



A noter que dans la définition de la commande `\cmd`, on ne précise aucun argument, contrairement aux 2 versions spécifiques

9.6 Personnaliser les commandes de titres

La méthode classique est d'utiliser la syntaxe \TeX pour redéfinir les commandes de titres. Ces commandes contiennent le caractère `@`, comme toutes les commandes internes de \LaTeX , ce qui demande d'utiliser les commande `\makeatletter` et `\makeatother` pour autoriser ces redéfinitions.

Par exemple, pour redéfinir la commande `\part`, on procèdera comme suit :

```

\makeatletter
\def\@part[#1]#2{%
...
}
\makeatother

```

Le titre de la partie est `#2` et le numéro de la partie est accessible via la commande `\thepart`.

Pour redéfinir la commande `\chapter`, il ne faut pas perdre de vue qu'il peut aussi y avoir une version étoilée, ce qui correspond à 2 redéfinitions à faire :

```

\makeatletter
\def\@makechapterhead#1{% -> \chapter
...
}
\def\@makeschapterhead#1{% -> \chapter*
...
}
\makeatother

```

Le titre du chapitre est `#1` et le numéro du chapitre est accessible via la commande `\thechapter`.

Pour la commande `\section`, comme pour les titres précédents, vous pouvez aller voir la bibliothèque d'exemples proposées par Vincent Zoonekynd (voir ZOONEKYND, *LaTeX*).

Une extension très intéressante pour personnaliser les titres semble être `TITLESEC`.

9.7 Gestion clé-valeur des arguments optionnels

Pour définir un système de paramètres sous forme clé-valeur, il existe plusieurs extensions : **PGFKEYS** (utilisée par TiKZ) et **KEYVAL** (utilisée par *beamer*) sont les plus réputés.

Voici un exemple tiré de l'extension **TIKZ-UML** utilisant **PGFKEYS** :

```
\newcommand{\umlport}[3][]{%
  \pgfkeys{/tikzuml/port/.cd, draw/.initial=\tumlDrawColor, fill/.initial=\tumlFillColor,%
    width/.initial=\tumlWidth,%
    .unknown/.code={%
      \let\keyname=\pgfkeyscurrentname%
      \let\keyvalue=\pgfkeyscurrentvalue%
      \errmessage{TIKZUML ERROR : in umlport forbidden option \keyname}%
    }%
  }%
  \pgfkeys{/tikzuml/port/.cd, #1}%
  \pgfkeys{/tikzuml/port/.cd, width/.get=\tikzumlPortWidth,%
    draw/.get=\tikzumlPortDrawColor, fill/.get=\tikzumlPortFillColor}%
  ...
}%
```

Le premier appel à la commande `\pgfkeys` définit les 3 clés souhaitées ainsi que leur valeur initiale, à savoir **draw**, **fill** et **width**. Ces 3 clés sont protégées par un préfixe (ou espace de nom). L'utilisation d'une clé différente lancera un message d'erreur.

Le deuxième appel à la commande `\pgfkeys` traite les options spécifiées lors de l'utilisation de la commande `\umlport`.

Le troisième et dernier appel à la commande `\pgfkeys` stocke la valeur de chaque clé dans une variable spécifique.

9.8 Espaces parasites et commentaires

Lorsque l'on définit des commandes/environnements personnalisés, dont le contenu est sur plusieurs lignes, il peut arriver que l'utilisation de cette commande engendre un espace parasite. Pour y remédier, 2 possibilités :

- Tout écrire sur la même ligne, mais cela nuit grandement à la lisibilité de l'implémentation de la commande ou de l'environnement ;
- Utiliser le symbole de commentaire % à la fin de chaque ligne de l'implémentation. Cette deuxième solution a le mérite d'être une contrainte très légère qui ne gêne en aucune façon la lisibilité. En effet, terminer chaque ligne par % fait que tous les espacements, y compris le caractère de saut de ligne, sont ignorés, jusqu'au début de la ligne suivante. Par conséquent, si espacement parasite il y a, il va désormais « passer en commentaire » et sera rendu inopérant.

Élément indispensable de tout gros document, la bibliographie a très tôt été gérée par \LaTeX . Dès 1985, Leslie Lamport et Oren Patashnik ont créé et rendu disponible le premier moteur de compilation de bibliographie, ainsi que le format de fichier de biblio toujours utilisé aujourd'hui.

Nous allons donc voir ici comment générer ce fichier, puis comment l'utiliser dans un document \LaTeX .

10.1 Ecrire un fichier de bibliographie

Pour pouvoir insérer une bibliographie, encore faut-il en avoir une à insérer. Pour \LaTeX , un fichier de bibliographie est un fichier d'extension `.bib` qui contient des blocs de données du genre :

```
@manual{enumitem,
  author = {Javier Bezos L\{o}pez},
  date-added = {2019-10-14 11:14:32 +0200},
  date-modified = {2022-07-04 14:35:23 +0200},
  title = {enumitem -- Control layout of itemize, enumerate, description},
  url = {https://www.ctan.org/pkg/enumitem},
  year = {2003},
  bdsk-url-1 = {https://www.ctan.org/pkg/enumitem}}

@book{MiGo04,
  address = {Boston, MA, USA},
  author = {Frank Mittelbach and Michel Goossens},
  date-added = {2019-10-14 00:17:18 +0200},
  date-modified = {2019-11-06 10:53:13 +0100},
  edition = {2st},
  isbn = {978-0-201-36299-2},
  publisher = {Addison-Wesley Longman Publishing Co., Inc.},
  title = {The \LaTeX\ Companion},
  year = {2004}}
```

On peut y voir 2 éléments :

- un élément de type `manual` concernant la page web de référence de l'extension `ENUMITEM`, défini à partir d'un identifiant (`enumitem`), d'un titre, d'un auteur et d'une url.
- un élément de type `book` concernant le livre de référence « The \LaTeX Companion », défini à partir d'un identifiant (`MiGo04`), d'un titre, d'un auteur (contenant 2 noms séparés par le mot `and`), d'une édition, d'un numéro ISBN, d'une année de parution et d'un éditeur

Il existe bien entendu d'autres catégories : `article`, `conference`, `phdthesis`, `masterthesis`, `periodical`, `proceedings`, ... Chaque catégorie à son propre jeu de paramètres obligatoires.

Fort heureusement, il existe des logiciels qui nous permettent d'éditer et modifier des fichiers de bibliographie : `BIBDESK` pour MacOS (<https://bibdesk.sourceforge.io>), ou `JABREF` pour Windows, Linux et Mac OS (<http://www.jabref.org>) allant même jusqu'à nous permettre de lier à une fiche le pdf associé, une façon comme une autre d'organiser une bibliothèque!

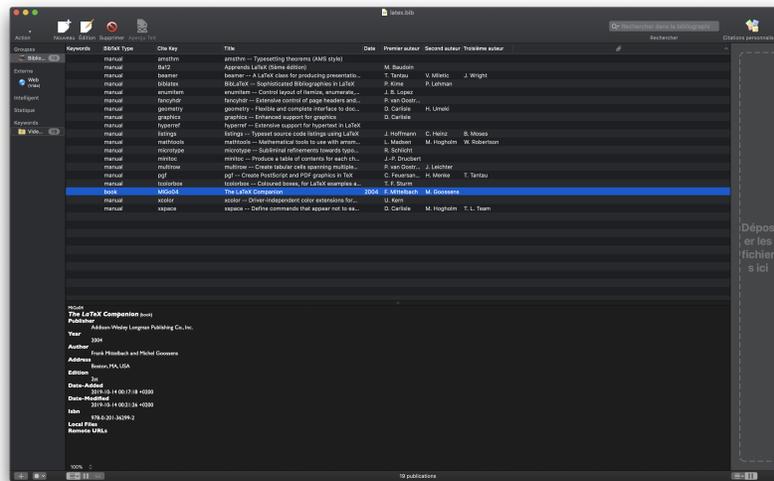


FIGURE 10.1 – BibDesk, éditeur de bibliographie pour MacOS

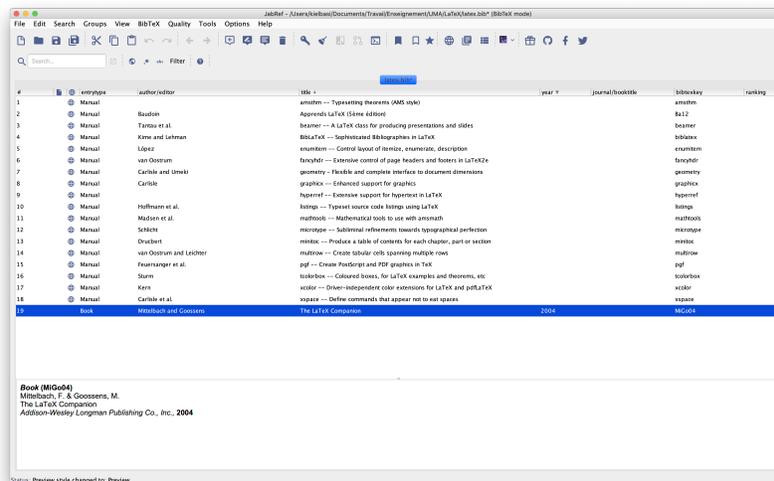


FIGURE 10.2 – JabRef, éditeur de bibliographie multi-plateforme

10.2 Un peu d'histoire : bib_TEX et l'extension NATBIB

Le moteur de gestion de biblio bib_TEX a été conçu par Oren Patashnik et Leslie Lamport en 1985.

L'extension la plus connue reposant sur le moteur bibtex est **NATBIB**, datant de 1993 et qui n'est plus maintenue depuis 2010.

Sa popularité est telle qu'il existe encore aujourd'hui des journaux scientifiques et des maisons d'éditions qui proposent encore ce système dans leur template d'écriture d'un article. Pourquoi? Parce que la force de **NATBIB** repose sur la grande diversité de fichiers de style de bibliographie (extension .bst) disponibles et chacun d'entre eux a mis au point son propre style.

La création de ces fichiers de styles repose en partie sur l'utilisation de l'extension complémentaire **CUSTOM-BIB** qui simplifie le processus.

Car il faut le reconnaître, le langage des fichiers .bst est complexe ! Par ailleurs, les labels des références sont soit numériques soit de la forme auteur-année, le logiciel `bibtex`, en tant que logiciel historique, gère très mal les caractères accentués et les caractères non-européens.

C'est donc tout naturellement qu'un remplaçant a vu le jour : le moteur `biber` associé à l'extension `BIBLATEX`.

10.3 biber et l'extension BIBLATEX

`BIBLATEX` est une extension apparue en 2006 avec la volonté de s'installer dans la continuité dans ce sens où la compatibilité avec le moteur `bibtex` a été assurée dès le début, alors que très vite, un nouveau moteur a vu le jour en 2009 : `biber`.

Qu'est-ce que ça a changé ?

- Les labels peuvent être construits sur un éventail plus large, comme la forme auteurs-titre, très prisée
- Le style de la bibliographie peut être contrôlé à l'aide de commandes \TeX standards. Plus besoin de manipuler un langage complexe supplémentaire comme celui des fichiers .bst.
- La possibilité de manipuler plusieurs fichiers de bibliographies dans un même document
- La possibilité d'afficher plusieurs bibliographies différentes à l'aide de mot-clés permettant de les différencier.

Et du côté de `biber` :

- De nouveaux champs sont gérables dans le format BibTeX (fichier .bib)
- Support Unicode des fichiers BibTeX
- Davantage de possibilités d'ordonnement des items affichés dans une bibliographie.

Bien entendu, utiliser `biber` exige d'utiliser l'extension `BIBLATEX`, et il faut maintenant que les journaux scientifiques et les maisons d'édition passent à ce nouveau système. Certaines ne l'ont pas encore fait !

10.4 La bibliographie en pratique

La première chose à faire est bien entendu l'inclure l'extension `BIBLATEX`. Cette extension dispose d'un ensemble d'options dont voici les plus intéressantes :

`backend` permet de sélectionner le moteur de bibliographie : `bibtex`, la version de référence de l'ancien moteur `bibtex8`, sa version 8 bits, ou `biber`, le nouveau moteur, sélectionné par défaut.

`style` permet de sélectionner à la fois un style de bibliographie et un style de citation (qui peuvent être choisis séparément par des options dédiées `bibstyle` et `citestyle`). Les valeurs les plus courantes sont :

numeric le style standard de \TeX en ce qui concerne les bibliographies. Les références sont affichées avec un label numérique.

alphabetic l'équivalent du style `alpha` de `bibTeX`, le plus utilisé avec l'extension `NATBIB`.

authoryear ce style d'affichage basé sur les auteurs et l'année de publication est destiné aux citations à l'intérieur du texte, mais fonctionne bien aussi avec les notes de bas de page.

authortitle ce style d'affichage basé sur les auteurs et le titre est destiné aux références en notes de bas de page. A titre personnel, je ne trouve pas nécessairement que ce style pose problème à l'intérieur du texte. C'est ce que j'utilise dans ce document.

verbose lors de la première citation à une référence donnée, cela donne le même résultat que dans la page dédiée à l’affichage de la bibliographie. Cela fonctionne ensuite comme une citation traditionnelle.

reading ce style permet de constituer des listes de lectures plus détaillées en affichant notamment les clés *annotation*, *abstract*, *library* et *file*. Pour la citation d’une référence, cela se comporte comme *authortitle*.

Il nous faut d’abord charger dans le préambule le ou les fichiers de bibliographie à l’aide de la commande `\addbibresource` :

```
\addbibresource{latex.bib}
```

Enfin, il nous reste à demander l’affichage de la bibliographie :

```
\printbibliography
```

Pour faire référence à un élément de la bibliographie, on dispose de plusieurs commandes :

\cite la commande classique qui permet de faire une référence, qui prend en argument l’identifiant de la référence à citer. Il existe une variante `\Cite` pour faire commencer la citation par une majuscule, pratique pour un début de phrase ;

\parencite cette commande produit le même résultat que la commande `\cite`, en ajoutant des parenthèses autour. Il existe également une variante `\Parencite` qui a le même rôle que `\Cite` ;

\footcite cette commande produit le même résultat que la commande `\cite`, mais la place dans une note de bas de page. En simplifiant, `\footcite{cle}=\footnote{\cite{cle}}`. Si `\footcite` se rapporte à `\footnote`, il existe une version relative à `\footnotetext` (voir section 5.7) : elle s’appelle `\footcitetext`.

\autocite cette commande peut se comporter comme l’une des 3 précédentes commandes, selon la valeur d’une option d’extension : `autocite` qui est également définie par le choix d’un style de citation.

La majorité des doctorants qui commencent à rédiger vous le diront, la définition des notations utilisées n'est pas toujours simple : où est-ce qu'on la met dans notre mémoire? Comment préciser où ces notations sont utilisées pour la première fois? Comment catégoriser les notations? ...

La partie gestion de cette nomenclature peut être pilotée par l'extension **NOMENCL**.

11.1 Définir ses notations

Le principe de **NOMENCL** est de pouvoir définir ses notations n'importe où dans le document. Pour définir une notation, on utilise la commande **\nomenclature**.

```
\nomenclature[physique]{\(\mathcal{H}_i\)}{Le Hamiltonien du sous-système \(\i\)}
```

Cette commande a 2 arguments obligatoires, respectivement :

- La notation à définir (1er argument entre accolades)
- La signification de la notation (2ème argument entre accolades)
- (optionnel, entre crochets) un nom de catégorie de notations. Par défaut, toutes les notations sont dans une seule et unique catégorie.

En général, les notations sont associées à la première équation dans laquelle elles apparaissent. L'intérêt de la commande **\nomenclature** est de pouvoir l'utiliser à l'intérieur d'un environnement *equation*. Dans ce cas, vous pouvez utiliser les commandes additionnelles suivantes pour faire une référence croisée dans la nomenclature :

\nomrefeq pour compléter la définition par « , voir équation <num> »

\nomrefpage pour compléter la définition par « , page <num> »

\nomrefeqpage pour compléter la définition par « , voir équation <num>, page <num2> »

11.2 Construire une nomenclature

Pour pouvoir utiliser les commandes **\nomenclature** disséminées dans votre document, il y a 2 commandes complémentaires à utiliser :

\makenomenclature Cette commande à placer dans le préambule a pour but de signifier la construction d'une nomenclature, qui sera contenue dans un fichier d'extension .nlo

\printnomenclature C'est la commande qui permet d'imprimer la nomenclature.

Pour renommer le titre de la page de nomenclature, vous devez redéfinir la commande **\nomname** :

```
\renewcommand{\nomname}{Mon titre de nomenclature}
```

Et si vous souhaitez ajouter un paragraphe introductif dans votre nomenclature, vous devez redéfinir la commande **\nompreamble** :

```
\renewcommand{\nompreamble}{Mon préambule de nomenclature}
```

Les différentes notations seront rangées dans l'ordre où elles apparaissent dans votre document dans une liste unique. Mais vous avez la possibilité de définir des catégories, auquel cas, les notations seront regroupées dans leur catégories respectives.

11.3 Définir des catégories de notations

La commande `\nomenclature` prend un argument optionnel, de préférence un mot unique sans accents, qui va référencer une catégorie de notations. Dans l'exemple précédent, on va ranger la définition du Hamiltonien dans une catégorie intitulée physique :

```
\nomenclature[physique]{\(\mathcal{H}_i\)}{Le Hamiltonien du sous-système \(\i\)}
```

Comment donner un titre à chaque catégorie ?

La commande `\nomgroup` permet de définir le titre de la page de nomenclature. Cette commande prend un argument, correspondant au mots-clé de catégories utilisés dans les commandes `\nomenclature`, permettant de définir un titre pour chaque catégorie dans la nomenclature :

```
\renewcommand{\nomgroup}[1]{%
  \item[\bfseries
  \ifstrequal{#1}{physique}{Constantes physiques}{%
  \ifstrequal{#1}{autre}{Divers}{}}%
}
```



L'ajout de nomenclatures va nécessiter de compiler 2 fois le document pour prendre en compte leurs modifications :

1. Régénération du fichier d'extension .nlo par l'appel de la commande `\makenomenclature`
2. Prise en compte de fichier .nlo mis à jour lors de la compilation suivante pour mettre à jour la nomenclature.

On parle de références croisées lorsque l'on souhaite citer un élément numéroté quelconque d'un document et que la citation soit cliquable pour aller directement à l'endroit cité. Tous les éléments numérotés (titres, équations, figures, tables, ...) peuvent être référencés. Pour cela, l'extension la plus courante est sans conteste `HYPERRREF`. Mais pour des raisons de personnalisation, on lui ajoute maintenant l'extension `CLEVEREF`.



L'ajout de références croisées va nécessiter de compiler 2 fois le document pour prendre en compte leurs modifications :

1. Régénération du fichier d'extension `.aux` contenant les informations associant les différents labels et les numéros de page/section/...qui leurs sont associés.
2. Prise en compte de fichier `.aux` lors de la compilation suivante pour mettre à jour les références croisées.

12.1 Définir des étiquettes (=labels)

Pour pouvoir référencer un élément numéroté, il faut au préalable lui donner une étiquette avec la commande `\label` qui prend un seul argument, le nom d'étiquette que l'on choisit. Ce nom d'étiquette doit être unique dans votre document. Il est constitué de caractères alphanumériques mais peut aussi contenir certains symboles de ponctuation comme «:», «.», «-».

Pour des titres, on utilise la commande `\label` juste après la commande de titre.

Pour une équation numérotée, on utilise la commande `\label` juste après le `\begin{equation}` de l'environnement correspondant. Pour des systèmes d'équations numérotées ou des ensembles d'équations alignées, on utilise la commande `\label` en fin de ligne.

Pour une figure ou une table, on utilise la commande `\label` après la commande `\caption`.

12.2 Définir une référence

Voici les commandes principales fournies par l'extension `HYPERRREF` :

`\url` C'est la commande principale pour faire référence à un site web en lui donnant l'adresse en argument.

`\ref` C'est la commande principale pour faire référence à un élément. Elle donne le numéro de l'élément référencé

`\eqref` C'est la commande à priori dédiée au référencement d'une équation. Elle retourne la même chose que `\ref`, mais en le mettant entre parenthèses.

`\pageref` Au lieu de donner le numéro de l'objet référencé, cette commande donne le numéro de la page contenant l'élément référencé.

`\autoref` Cette commande fait précéder le numéro de l'élément référencé par un mot désignant sa nature (chapitre pour un `\chapter`, équation pour un environnement `equation`, ...).

Voici les commandes principales fournies par l'extension `CLEVEREF` :

\cref C'est la commande principale pour faire référence à un ou plusieurs éléments. Elle donne le numéro de l'élément référencé précédé d'un mot désignant sa nature (chapitre pour un `\chapter`, équation pour un environnement *equation*, ...).

\crefrange C'est la commande à priori dédiée au référencement d'une série d'éléments consécutifs. Elle prend le premier et le dernier labels de la liste.

\cpageref Au lieu de donner le numéro de l'objet référencé, cette commande donne le numéro de la page contenant l'élément référencé.

Ces commandes existent avec un C majuscule (`\Cref`, `\Creffrange`, `\Cpageref`, ...) dans le cas où la référence croisée se trouve en début de phrase, ce qui permet de faire commencer la référence par une majuscule.



Avec l'extension **TCOLORBOX**, vous pouvez référencer les théorèmes et boîtes de listings numérotées uniquement à l'aide des commandes fournies par `\cleveref`

Nous avons vu en section 5.2 comment définir une table des matières globale avec la commande `\tableofcontents`. Nous allons voir ici comment définir plusieurs tables des matières locales aux titres de plus hauts niveaux de votre document. L'extension la plus courante est `MINITOC`.

Pour cela, il faut ajouter juste après le `\begin{document}` l'une des 3 commandes suivantes :

`\dominitoc` Pour préparer la construction de tables des matières locales aux chapitres (pour les classes *book* et *report*)

`\doparttoc` Pour préparer la construction de tables des matières locales aux parties (pour les classes *book* et *article*)

`\dosecttoc` Pour préparer la construction de tables des matières locales aux sections (pour la classe *article*)

Ensuite, pour faire apparaître les tables des matières locales, il faut utiliser la commande associée `\minitoc`, `\parttoc` ou `\secttoc` à chaque endroit souhaité.



Cela va générer des fichiers `.ptc`, `.mtc` ou `.stc` (équivalents de `.toc`)

De même, vous disposez de `\dominilof`, ..., `\dominilot`, ... pour les listes de figures et les listes de tables.

Créer un index est aussi une fonctionnalité proposée par \TeX , et ce depuis 1987 et la création par Leslie Lamport du moteur d'index `makeindex`. Les extensions gérant l'index ont évolué au cours du temps, et il semble qu'aujourd'hui, un très bon candidat soit `IMAKEIDX`.

14.1 Génération de l'index

Commençons par le plus simple : l'affichage de l'index. Pour cela, il faut utiliser la commande `\printindex` à l'endroit voulu dans votre document.

Maintenant, il faut le générer cet index. Pour annoncer la mise en place d'un index, il faut utiliser la commande `\makeindex` dans le préambule de votre document. Lors de la compilation de celui-ci, un fichier contenant toutes les entrées de l'index va être généré. Il a pour extension `.idx`.

Ensuite, il faut compiler l'index. C'est le même principe que pour une bibliographie où vous utilisez un moteur dédié. Ici, c'est `makeindex`. Il prend en argument le fichier `.idx` précédemment généré (il n'est pas nécessaire de préciser l'extension). En sortie, vous obtiendrez 2 fichiers : un fichier d'extension `.ilg` (le fichier de log d'exécution de `makeindex`), et un fichier d'extension `.ind`, le fichier d'index proprement dit (équivalent des fichiers `.toc` et `.bbl`).

Il ne vous reste plus qu'à recompiler votre document pour que l'index soit pris en compte.

Nous allons maintenant voir comment ajouter des entrées dans l'index.

14.2 Les entrées de l'index

Pour ajouter une entrée dans l'index, on utilise la commande `\index` avec un seul argument, une expression sans accents désignant l'entrée. Si on a plusieurs entrées à définir au même endroit, il faudra utiliser autant de fois la commande `\index`. Voici les différentes possibilités offertes :

`\index{entree}` ajoute l'entrée `entree` dans l'index (les espaces sont autorisés)

`\index{entree@entrée}` ajoute l'entrée `entree` dans l'index avec le label `entrée`. L'utilisation de `@` a été probablement initialement conçue pour gérer les accents, ce qui est maintenant inutile avec les moteurs XeTeX et LuaTeX , mais en réalité, cela peut permettre de personnaliser le texte affiché en lui donnant du style (italique, ...)

`\index{entree!sous-entree}` ajoute la sous-entrée `sous-entree` à l'entrée `entree` dans l'index

`\index{entree!sous-entree!sous-sous-entree}` ajoute la sous-sous-entrée `sous-sous-entree` à la sous-entrée `sous-entree` de l'entrée `entree` dans l'index

`\index{entree|see{autre entree}}` ajoute l'entrée `autre entree` en référence à l'entrée `entree` dans l'index

14.3 Personnaliser l'index

La définition du style d'un index est déportée dans un fichier dédié d'extension .ist. Pour le prendre en compte, il suffit d'écrire `\makeindex[options=-s myIndex]`, si votre fichier s'appelle `myIndex.ist`. Voici le fichier de style utilisé dans ce document :

```
actual '?'
headings_flag 1
heading_prefix "\tikz \node[fill=maincolor!20!white, text=maincolor, text centered, text
width=\linewidth]{\"
heading_suffix \"};\n\nopagebreak\n\"
symhead_positive "Symboles"
numhead_positive "Nombres"
delim_0 " \dotfill "
delim_1 " \dotfill "
delim_2 " \dotfill "
```

Nous allons voir maintenant l'explication de chacune des options utilisées (bien entendu, il en existe plein d'autres) :

actual permet d'utiliser un autre délimiteur que @ pour distinguer le mot clé brut du texte affiché dans l'index. J'ai utilisé cette option pour pouvoir utiliser des commandes contenant le caractère @.

headings_flag Cette option permet de décider d'afficher la lettre en guise d'entête de bloc dans l'index (1 pour l'afficher en majuscules, -1 pour l'ajouter ne minuscules) ou non (valeur 0)

heading_prefix Code à ajouter avant le contenu de chaque entête

heading_suffix Code à ajouter après le contenu de chaque entête

symhead_positive texte d'en-tête de catégorie pour les entrée de l'index ne commençant pas par un caractère alphanumérique (comme les commandes \TeX qui commencent par \ par exemple) dans le cas où **heading_flags** est positif (il existe **symhead_negative** dans l'autre cas). Il faut comprendre ici que l'extension **IMAKEIDX** n'est pas multilingue. Donc cette catégorie de l'index aura par défaut pour en-tête le mot « Symbols » (ou « symbols » si **heading_flags** est négatif) quelle que soit la langue utilisée dans votre document.

numhead_positive texte d'en-tête de catégorie pour les entrée de l'index commençant par un chiffre dans le cas où **heading_flags** est positif (il existe **numhead_negative** dans l'autre cas). Comme pour **symhead_positive**, il faut comprendre ici que l'extension **IMAKEIDX** n'est pas multilingue. Donc cette catégorie de l'index aura par défaut pour en-tête le mot « Numbers » (ou « numbers » si **heading_flags** est négatif) quelle que soit la langue utilisée dans votre document.

delim_0, delim_1, delim_2 pour définir ce qui s'affiche entre une entrée de niveau 0, 1 ou 2, et le ou les numéros de pages associés.

Il s'agit d'un aspect plus méthodologique de la rédaction d'un gros document, qui concerne le fait de mettre des annotations ou commentaires en prévision de rédaction ultérieure. Ou alors un co-auteur veut faire figurer ses différentes remarques et commentaires sur l'état d'avancement de la rédaction, ...

Pour cela, on peut utiliser l'extension **TODONOTES** .

15.1 Définir une annotation

Le package **TODONOTES** propose 2 commandes pour définir des annotations :

\todo permet de faire un commentaire. Il prend le texte du commentaire en argument obligatoire, et dispose d'un panel d'options dont voici celles qui me semblent les plus intéressantes :

color Elle permet de spécifier la couleur de l'annotation ;

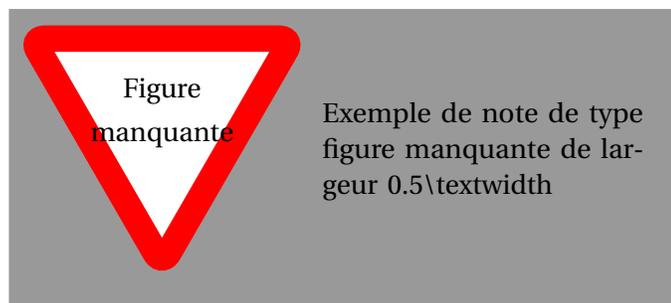
inline Elle permet de décider que l'annotation est intégrée au contenu du texte plutôt que d'apparaître dans la marge. Il y a des paramètres plus fins (à propos du texte, de la bordure de la boîte de couleur, ...),

mais pour ce qu'on cherche à en faire, cela suffit. Voici un exemple de note avec l'option inline

inlinewidth pour spécifier la largeur de la note quand elle est inline.

size Elle permet de spécifier la taille de la police (`\tiny`, ...) Ceci est une note de taille `\scriptsize` ;

\missingfigure permet de faire un commentaire sur la présence future d'une image. On peut par exemple pré-réserver la dimension de l'image manquante à l'aide des options **figwidth** et **figheight** :



On peut spécifier des paramètres globaux dans le préambule avec la commande **\setuptodonotes**.

15.2 Afficher la liste des notes

La commande **\listoftodos** est, du moins à mes yeux, la raison de l'utilité du package **TODONOTES** . Elle liste l'ensemble des commandes **\todos** et **\missingfigure**, sur le même principe que **\tableofcontents**, **\listoffigures**, ...

Dans cette liste, chaque item est précédé d'une petite boîte de couleurs avec les mêmes paramètres de couleurs que celles spécifiées à la commande correspondante (en activant l'option **colorinlistoftodos**



Mettez en place un code de couleur pour catégoriser vos \todos et dans la rédaction d'un gros document, comme un livre, un mémoire de stage ou de thèse, la page affichant la liste générée par \listoftodos vous permet de savoir où il y a des commentaires (point de vue du travail collaboratif) ou s'il en reste encore à traiter (point de vue de l'avancement de la rédaction). Voir page suivante pour un exemple.

 Voici un exemple de note avec l'option inline	65
 Ceci est une note de taille <code>\scriptsize</code>	65
Figure : Exemple de note de type figure manquante de largeur <code>0.5\textwidth</code>	65

16.1 Séparer le corps du document en plusieurs fichiers

La commande `\input`

La commande `\input` permet de charger du code \TeX contenu dans un autre fichier, comme si on l'avait directement écrit. Cet autre fichier ne peut pas contenir de préambule, et encore moins d'environnement *document*, seulement du code \TeX qui a vocation à se retrouver dans cet environnement *document*.

L'extension de ce fichier inclus est en général `.tex`, mais s'il peut s'agir d'une figure TikZ par exemple, on utilisera l'extension `.tikz`



On peut utiliser la commande `\input` en cascade, en chargeant des fichiers qui eux-mêmes en chargent d'autres

La commande `\include`

La commande `\include` va plus loin. Il s'agit toujours de contenu \TeX , mais cette fois, un changement de page va s'opérer lors de l'inclusion du fichier. L'autre changement, c'est que chaque fichier `.tex` inclus avec cette commande aura cette fois son propre fichier auxiliaire `.aux`

En spécifiant à l'aide de la commande `\includeonly` le ou les fichiers à considérer (noms des fichiers séparés par des virgules, et sans espaces), la compilation de votre document recompilera les fichiers sélectionnés et utilisera les fichiers `.aux` existants pour les autres, accélérant ainsi la phase de compilation.



Contrairement à `\input`, `\include` n'autorise pas les inclusions en cascades

16.2 Créer une extension à partir de son préambule personnalisé

Une autre façon de morceler votre fichier `.tex` est de déporter tout ou partie de votre préambule dans un fichier, qui de fait, va se comporter comme une extension \TeX . Si on obtient ainsi physiquement la séparation entre le fond et la forme, chère aux logiciels WYSIWYM dont \TeX fait partie, on organise ainsi ses fichiers de sorte à pouvoir réutiliser ses commandes et environnements personnels sans avoir besoin de faire un copier-coller depuis son fichier `.tex`.

Comment procéder?

1. Créer un fichier d'extension `.sty` (par exemple `monPackage.sty`) et ajouter les 2 lignes suivantes à l'intérieur :

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{monPackage}
```

16.2. CRÉER UNE EXTENSION À PARTIR DE SON PRÉAMBULE PERSONNALISÉ

2. Copier-coller dedans la partie de votre préambule que vous souhaitez y faire figurer.
3. Remplacer votre préambule par `\usepackage{monPackage}`

Bien que des commandes existent par défaut pour gérer les options d'une extension, je recommande d'utiliser l'extension **PGFOPTS**, qui permet en gros d'utiliser **PGFKEYS** pour les options d'extension. Voici un exemple tiré de l'extension **TIKZ-UML**

```
\def\tikzumlPackageLayersNum{3}%
\pgfkeys{/tikzuml/options/.cd, packageLayers/.initial=3}%
\pgfkeys{/tikzuml/options/.cd, packageLayers/.store in=\tikzumlPackageLayersNum}%
\def\tikzumlStateLayersNum{3}%
\pgfkeys{/tikzuml/options/.cd, stateLayers/.initial=3}%
\pgfkeys{/tikzuml/options/.cd, stateLayers/.store in=\tikzumlStateLayersNum}%
\def\tikzumlFragmentLayersNum{3}%
\pgfkeys{/tikzuml/options/.cd, fragmentLayers/.initial=3}%
\pgfkeys{/tikzuml/options/.cd, fragmentLayers/.store in=\tikzumlFragmentLayersNum}%
\def\tikzumlComponentLayersNum{3}%
\pgfkeys{/tikzuml/options/.cd, componentLayers/.initial=3}%
\pgfkeys{/tikzuml/options/.cd, componentLayers/.store in=\tikzumlComponentLayersNum}%
%
\ProcessPgfOptions{/tikzuml/options}%
```

Présentations en L^AT_EX avec *beamer*

Partie III

17 Introduction	71
17.1 La classe <i>beamer</i>	71
17.2 Un exemple pour commencer	71
18 Structuration d'une présentation	73
18.1 Le transparent de titre	73
18.2 Les transparents de contenu	73
18.3 Chapitrage, table des matières et automatisation	73
18.4 Les blocs	74
18.5 Les listes	75
18.6 Les environnements flottants	75
18.7 Théorèmes et équations	76
18.8 L'environnement <i>columns</i>	76
18.9 Variantes de polices supplémentaires	77
18.10 Références croisées et boutons	77
19 Animation d'une présentation	79
19.1 Les overlays	79
19.2 Les transitions de transparents	80
19.3 Animer une séquence d'images	81
20 L'apparence de votre présentation	83
20.1 Les thèmes prédéfinis	83
20.2 Personnalisation de l'apparence : cas général	87
20.3 Personnalisation de l'apparence : solutions usuelles	88
20.3.1 Changer la couleur principale du thème	88
20.3.2 Mettre une image en fond de transparent	88
20.3.3 Changer la couleur associée à la commande <code>\alert</code>	88
20.3.4 Changer la couleur associée à l'environnement <i>example-block</i>	88
20.3.5 Changer les couleurs du titre d'un transparent	88
20.3.6 Définir le comportement de la commande <code>\uncover</code>	89
20.3.7 Changer les boutons de navigation	89
A Prépambule L^AT_EX de ce document	90
Index	97
Index des commandes et environnements	99

Pour réaliser un exposé, une conférence, ..., il est inconcevable de ne pas avoir le moindre support de présentation : bien que cela soit devenu très rare, ce support peut encore constitué de transparents papiers à utiliser sur un rétroprojecteur, mais il s'agit le plus souvent de présentations sur ordinateurs branchés à un vidéoprojecteur.

Pour le premier cas, \LaTeX a prévu une classe de documents qui s'appelle *slides* .

Pour le deuxième cas, il existe différentes classes. Nous allons nous concentrer sur la plus importante d'entre elles et probablement la plus puissante : *beamer* .

17.1 La classe *beamer*

La classe *beamer* a été conçue par Till Tantau, professeur d'informatique théorique à l'Université de Lübeck en Allemagne. Il est également l'auteur de l'extension *TIKZ* . Pour la documentation officielle, voir TANTAU et al., *beamer – A \LaTeX class for producing presentations and slides* .

17.2 Un exemple pour commencer

```
\documentclass[11pt, french]{beamer}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{fourier}
\usepackage{babel}
\usepackage{microtype}

\usetheme{Madrid}
\usefonttheme{serif}

\title{First slides with  $\text{\LaTeX}$ } % titre %
\subtitle{With Beamer class}
\author{You} % auteur %
\institute{ENSTA-Paris}

\begin{document}

\maketitle

\begin{frame}
\frametitle{Hello}
\framesubtitle{World}

Hello  $\text{\LaTeX}$  world !!!
\end{frame}

\end{document}
```

Par rapport à un document, comme on l'a vu dans la partie précédente, on n'inclut pas l'extension *GEOMETRY* . La classe *beamer* fournit une définition des dimensions d'un transparent, que l'on peut contrôler avec l'option de classe *aspectratio* . Par défaut, le pdf a pour taille 128mm par 96mm (format 4/3).

17.2. UN EXEMPLE POUR COMMENCER

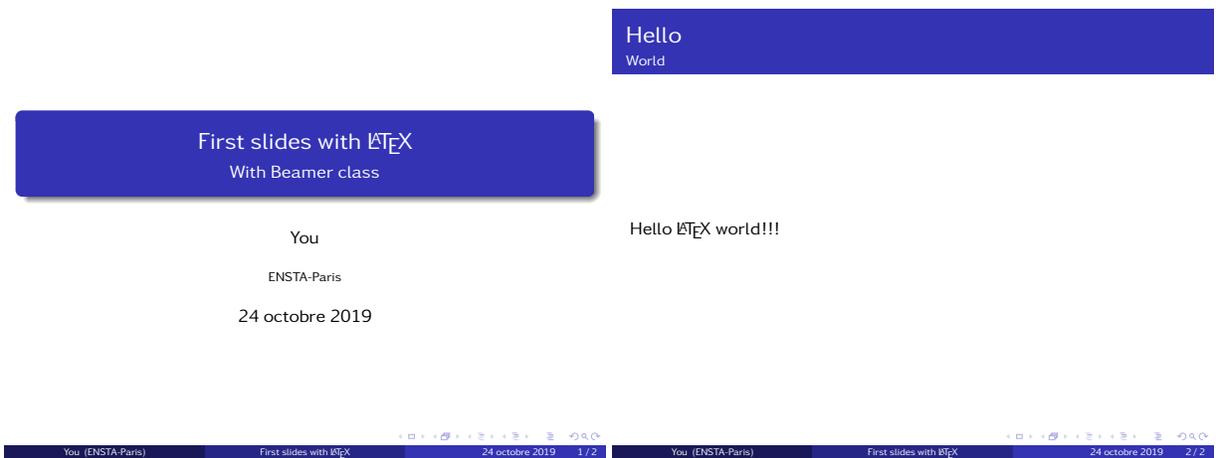


FIGURE 17.1 – Présentation générée avec le code du tutoriel

On obtient donc un document de 2 pages, chaque page correspondant à un transparent :

- Un transparent de titre
- Un transparent contenant une phrase



beamer met à disposition une série de thèmes graphiques (voir section 20.1). Dans cet exemple, le thème Madrid est chargé.



Pour pouvoir utiliser la police **FOURIER** avec *beamer*, il faut préciser à ce dernier que l'on utilise une police dite serif, à l'aide de la commande `\usefonttheme`, comme dans cet exemple.

18.1 Le transparent de titre

Comme nous venons de le voir, un transparent de titre est généré par la commande `\maketitle`.

Contrairement à un document classique, une présentation *beamer* autorise 5 informations dans le transparent de titre, chacune ayant une commande dédiée :

`\title` Le titre de la présentation

`\subtitle` Le sous-titre de la présentation

`\author` Le ou les auteurs de la présentation

`\date` La date de la présentation, ou le nom de l'événement dans lequel la présentation est faite

`\institute` Le ou les instituts du ou des auteurs de la présentation.

18.2 Les transparents de contenu

Un transparent est défini par un environnement *frame*. Un transparent peut avoir un titre (commande `\frametitle`) et éventuellement un sous-titre (commande `\framesubtitle`).



Il existe une commande `\frame` qui joue exactement le même rôle. Néanmoins, et ce n'est que mon avis personnel, utiliser l'environnement rend le fichier \LaTeX plus clair.

En guise de contenu, on peut mettre à peu près n'importe quoi en \LaTeX . Mais comme nous allons le voir dans la suite, *beamer* fournit un certain nombre de commandes et d'environnements pour aider à la mise en page.

Quelques options utiles de l'environnement *frame* :

`b` pour aligner verticalement le contenu d'un transparent vers le bas;

`c` pour centrer verticalement le contenu d'un transparent (comportement par défaut);

`t` pour aligner verticalement le contenu d'un transparent vers le haut;

`fragile` option nécessaire lorsque le contenu du transparent contient un listing de code ou du texte verbatim.

`allowframebreaks` option à utiliser lorsque le contenu est trop long par rapport à la taille d'un transparent. Dans ce cas, autant de transparents que nécessaires sont créés et un numéro est ajouté au titre du transparent. Très utile par exemple pour le transparent contenant de la bibliographie.

18.3 Chapitrage, table des matières et automatisation

Il est fortement recommandé de structurer votre présentation et d'en générer une table des matières. La classe *beamer* contient 4 niveaux de titres :

`\part` pour les titres de parties

`\section` pour les titres principaux

`\subsection` pour les sous-titres

`\subsubsection` pour les titres de niveau 4.

Vous pouvez également définir une annexe de votre présentation en utilisant la commande `\appendix` comme pour un document standard (section 5.2).



On peut souhaiter réutiliser les titres dans la définition des titres et sous-titres d'un transparent, auquel cas on utilisera à bon escient les commandes `\insertpart`, `\insertsection`, ...

Si l'on peut définir des titres, on peut bien entendu afficher une table des matières. Il s'agit toujours de la commande `\tableofcontents`, sauf que dans la classe *beamer*, elle dispose d'un certain nombre d'options intéressantes, dont les suivantes :

`currentsection` pour afficher la table des matières de la section courante et afficher celles des autres sections en filigrane;

`hideothersubsections` pour masquer les sous-sections des sections autres que la section courante;

`hideallsubsections` pour n'afficher que les sections dans la table des matières;



Par défaut, la table des matières de la partie courante est affichée.

FIGURE 18.1 – La table des matières



En général, on affiche dans un premier temps une table des matières principale, puis, pour chaque partie ou section, une table des matières propre plus détaillée. Il est possible d'automatiser le processus en utilisant les commandes `\AtBeginPart`, `\AtBeginSection`, ...

18.4 Les blocs

Les blocs sont en fait des boîtes de texte avec la possibilité d'y ajouter un titre. Elles permettent de structurer visuellement le contenu de vos transparents. Il existe principalement 3 environnements pour définir des blocs :

block, *exampleblock*, et *alertblock*. Ces environnements ne diffèrent que par la couleur dominante du bloc, comme on peut le voir ci-après :

```
\begin{block}{Bonjour le monde}
\begin{itemize}
\item Ca va bien ?
\item Oui, et vous ?
\end{itemize}
\end{block}

\begin{alertblock}{Bonjour le monde}
\begin{enumerate}
\item Ca va bien ?
\item Oui, et vous ?
\end{enumerate}
\end{alertblock}

\begin{exampleblock}{Bonjour le monde}
\begin{description}
\item[Vous] Ca va bien ?
\item[Moi] Oui, et vous ?
\end{description}
\end{exampleblock}
```



FIGURE 18.2 – Les blocs

18.5 Les listes

Rien de nouveau sur les listes que vous ne sachiez déjà d'après la section 5.6, hormis que leur apparence est contrôlée par le thème de votre présentation *beamer* et plus précisément par le style des environnements fournis par *beamer* dans lesquels on définit des listes, comme les blocs : voir figure 18.2.

18.6 Les environnements flottants

Il y a deux différences avec leur utilisation présentée dans la section 7.3 :

1. Un environnement flottant n'est en réalité pas destiné à être utilisé en tant que flottant, mais juste comme un environnement permettant de définir une légende à un tableau ou une figure. Cela n'a pas de sens de le positionner autrement que dans le transparent dans lequel on le définit.

2. Si les environnements flottants sont quand même numérotés, cette numérotation n'a pas d'utilité en pratique, c'est pourquoi il n'y a aucun numéro visible dans la légende. À la base, ce numéro est destiné au cas où l'on fait référence à une figure qui n'est pas nécessairement dans la page que l'on est en train de lire, ce qui n'est pas le cas avec une présentation.

18.7 Théorèmes et équations

La classe *beamer* charge automatiquement l'extension *AMSTHM*, abordée en section 6.6.

9 environnements de type théorèmes sont déjà pré-définis dans la classe *beamer* : *theorem*, *corollary*, *definition*, *definitions*, *fact*, *lemma*, *proof*, *example*, and *examples*.

Rien ne vous empêche bien entendu d'en définir d'autres ...

Il y a toutefois une différence de taille avec les théorèmes vus en section 6.6 : tout comme les environnements flottants, les théorèmes n'ont pas de numérotation visible, même s'ils sont numérotés .

Par contre, les équations sont normalement numérotées, pour la simple et bonne raison que l'on peut écrire plusieurs équations dans un même transparent et que l'on peut la référencer dans les explications autour.

18.8 L'environnement *columns*

Environnement très intéressant de *beamer*, l'environnement *columns* permet, comme son nom l'indique, de définir de manière simple des colonnes et surtout de maîtriser l'alignement vertical dans ces colonnes.

```
\begin{columns}
\begin{column}{0.4\textwidth}
\begin{block}{Bonjour le monde}
\begin{itemize}
\item Ca va bien ?
\item Oui, et vous ?
\end{itemize}
\end{block}
\end{column}
\begin{column}{0.57\textwidth}
\begin{exampleblock}{Bonjour le monde}
\begin{enumerate}
\item Ca va bien ?
\item Oui, et vous ?
\end{enumerate}
\end{exampleblock}
\end{column}
\end{columns}
```

Quelques options utiles de l'environnement *columns* :

- b** pour aligner verticalement la dernière ligne de chaque colonne;
- c** pour centrer verticalement le contenu d'une colonne (comportement par défaut);
- t** pour aligner verticalement la première ligne de chaque colonne;
- T** pour aligner verticalement le contenu des colonnes par le haut.



Pour l'environnement *column*, on peut modifier le comportement choisi en utilisant également l'une de ces options.

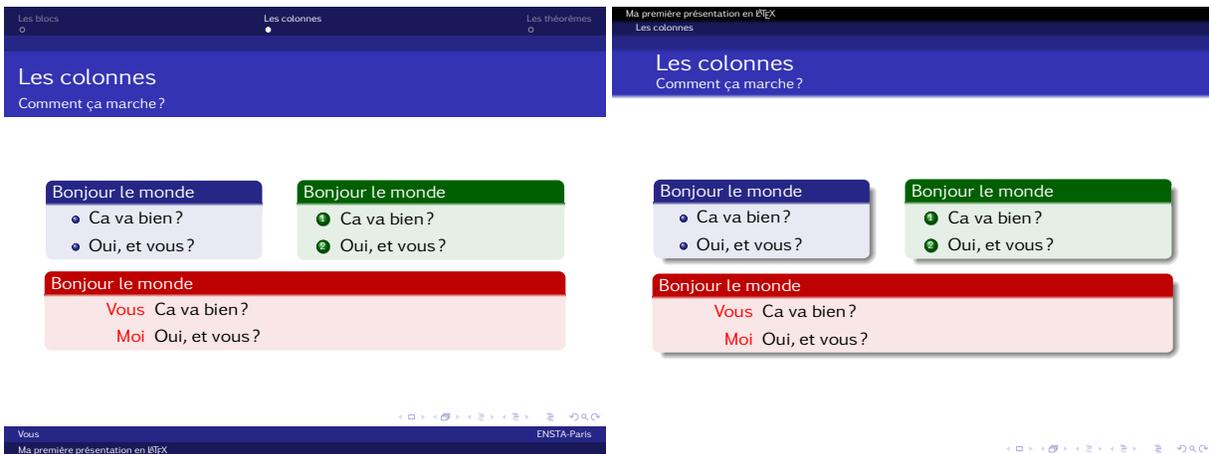


FIGURE 18.3 – Les colonnes

18.9 Variantes de polices supplémentaires

On peut bien entendu utiliser les commandes de changement de taille du texte vues en section 5.4. *beamer* ajoute 2 commandes supplémentaires pour du texte très petit : `\Tiny` et `\TINY`.



Même remarque qu'en section 5.4, ces commande sont plutôt destinées à la personnalisation des transparents (titres, entêtes, pieds de transparents, barre latérale, ...), des tableaux ou des listings de codes qu'à modifier le corps de texte proprement dit (ce qui est désagréable à lire).

A propos des couleurs, on peut faire exactement de la manière vue en section 5.4 avec l'utilisation de l'extension *XCOLOR*. Néanmoins, 2 commandes propres à *beamer* permettent de mettre en lumière du texte :

`\structure` pour utiliser la couleur principale du thème de votre présentation, pour mettre en valeur les éléments qui articulent le propos de votre présentation

`\alert` pour utiliser une couleur en général dans les tons rouges

18.10 Références croisées et boutons



Ce principe de *beamer* a déjà été évoqué, mais rappelons le : tous les éléments de contenu numérotables (titres de parties, sections ou sous-sections, environnements flottants, théorèmes, ...) dans les classes *article*, *book*, ... n'ont pas de numérotation visible dans *beamer*, même s'ils sont quand même numérotés en interne.

Seules les équations font exception à cette règle. La raison est simple, faire référence à un élément d'une présentation présenté dans un transparent antérieur par un numéro n'a pas de sens réel, sauf si le numéro correspond à celui du transparent (sous-entendu l'environnement *frame*) dans lequel il est défini, alors que pour des équations, il peut y en avoir plusieurs dans un même transparent.

L'extension *HYPERREF* étant automatiquement chargée par la classe *beamer*, on dispose des commandes `\ref`, `\eqref`, ... vues au chapitre 12 en se plaçant dans le contexte que nous venons d'expliquer.

Pour résumer, le seul cas de figure que l'on n'avait pas avant est de pouvoir faire référence à un transparent, pour récupérer son numéro, et faire en sorte que cette référence soit cliquable pour pouvoir naviguer vers le

transparent référencé.

Ce cas de figure correspond à une situation bien identifiée : prévoir des explications supplémentaires au transparent en cours, en les mettant par exemple en annexe, et en pouvant naviguer vers cette annexe et en revenir.

```
\begin{frame}\label{f.resultat}
\frametitle{Resultat brut}

...

\hyperlink{f.details}{\beamergotobutton{Explications}}

\end{frame}

...

\begin{frame}\label{f.details}
\frametitle{Details du resultat 1/2}

...

\end{frame}

\begin{frame}
\frametitle{Details du resultat 2/2}

...

\hyperlink{f.resultat}{\beamerreturnbutton{Retour}}

\end{frame}
```

Option label L'environnement *frame* dispose d'une option **label** pour lui attribuer une étiquette. Cette option est de loin préférable à l'utilisation de la commande **\label** vue au chapitre 12), car elle tient compte de la notion de couches d'un transparents qui sera abordée en section 19.1.

\hyperlink pour définir une référence croisée vers un label (Cela remplace les commande de type **\ref** vues au chapitre 12).

On peut bien entendu définir une référence sur du texte standard, mais comme nous sommes dans une présentation, des boutons, c'est quand même plus joli!!! Les commandes suivantes permettent de définir des boutons, avec leur texte en argument obligatoire (donc entre accolades) :

\beamerbutton commande générale pour définir un bouton.

\beamergotobutton commande pour définir un bouton dont le texte est précédé en général d'une flèche vers la droite pour indiquer que l'on va quelque part ailleurs.

\beamerkipbutton commande pour définir un bouton dont le texte est précédé d'une double flèche vers la droite pour indiquer que l'on va prendre un raccourci

\beamerreturnbutton commande pour définir un bouton dont le texte est précédé en général d'une flèche vers la gauche pour indiquer que l'on va retourner à un transparent déjà vu auparavant.

Dans le chapitre précédent, nous avons vu finalement les commandes et les environnements de la classe *beamer* qui nous permettent de mettre en forme du contenu. Ce qui fait qu'une présentation n'est pas qu'un simple document réside dans ce que nous allons aborder dans ce chapitre : la manière dont les éléments des transparents s'affichent à l'écran, autrement dit les animations de transparents. C'est là à mon avis que réside le point fort de la classe *beamer* et qui a fait qu'il a supplanté très vite la concurrence, parce que la manière d'animer le contenu se fait finalement de manière assez intuitive avec le concept d'« overlays ».

19.1 Les overlays

Jusqu'à présent, nous avons dit qu'un environnement *frame* correspond à un transparent, ce qui est toujours vrai. Ce qu'il faut comprendre avant tout avec les overlays, c'est que dès qu'un élément de contenu change (de couleur, de position, ...), une page de pdf est générée, donc les overlays sont certes indispensables mais très consommateurs de temps de compilation de votre présentation.

A titre personnel, j'ai eu l'occasion de préparer un exposé contenant 45 transparents, mais parce que j'avais beaucoup d'animations (graphiques animés essentiellement), le pdf faisait 250 pages!!!

Le principe de base est d'utiliser des commandes spécifiques de la classe *beamer* qui prennent 2 arguments : le deuxième entre accolades pour donner le contenu à animer, et le premier entre chevrons pour dire quand afficher ce contenu.

```
\begin{frame}
\frametitle{Exemple d'utilisation des overlays}

\visible<1>{Texte visible uniquement en page 1 du transparent.}

\only<2->{Texte visible uniquement en page 2 et suivantes du transparent.}

\end{frame}
```

Il existe essentiellement 3 commandes spécifiques pour les animations :

\only Quand le contenu passé en argument de cette commande n'est pas affiché, non seulement il n'est pas visible, mais aucune place ne lui est réservée dans le corps de texte du transparent.

\uncover Quand le contenu passé en argument de cette commande n'est pas affiché, une place lui est réservée et le contenu est visible en filigrane si la transparence est activée (voir la section 20.3.6), invisible dans le cas contraire.

\visible Quand le contenu passé en argument de cette commande n'est pas affiché, une place lui est réservée mais le contenu n'est pas visible.

Il existe d'autres commandes plus avancées mais finalement ces 3 là suffisent (encore que **\uncover** soit beaucoup moins utilisée que **\only** et **\visible**).

Parlons un peu de la syntaxe de ce qui est entre chevrons :

- Si cela contient un nombre entier, alors ce nombre est celui de la page du transparent dans laquelle le contenu sera affiché;

- Si cela contient un nombre entier suivi d'un tiret, alors le contenu sera affiché à partir de la page indiquée;
- Si cela contient un nombre entier précédé d'un tiret, alors le contenu sera affiché jusqu'à la page indiquée;
- Si cela contient 2 nombres entiers séparés d'un tiret, alors le contenu sera affiché pour les pages de la séquence indiquée.

On peut bien entendu avoir plusieurs spécifications de la sorte à donner, auquel cas on donne la liste des spécifications séparées par des virgules.

```
\begin{frame}
\frametitle{Exemple d'utilisation des overlays}

\visible<1,3-5>{Texte visible uniquement en pages 1, 3, 4 et 5 du transparent.}

\only<2,6->{Texte visible uniquement en pages 2, 6 et suivantes du transparent.}

\end{frame}
```

Maintenant, les chevrons peuvent être directement appliqués à toutes les commandes et tous les environnements de *beamer*, ainsi qu'à un certain nombre d'environnements et commandes classiques mais souvent présents dans des transparents :

\structure Les chevrons indiquent sur quelles pages du transparent opérer le changement de style. Il en va de même pour la commande **\alert**, mais aussi les commandes de changement de style vues en exemple 5.2, ainsi que **\emph** et **\color**

\includegraphics Les chevrons indiquent sur quelles pages du transparent afficher l'image.

Théorèmes et blocs Les chevrons indiquent sur quelles pages du transparent afficher le théorème ou le bloc

Listes Attardons nous un peu sur les listes, car en terme d'animations, les possibilités sont plus nombreuses :

- Nous pouvons utiliser bien entendu les chevrons sur les commandes **\item** pour afficher les listes item(s) par item(s).
- Lorsqu'on souhaite afficher une liste item par item, on peut indiquer le chevron en option de l'environnement *itemize*, *enumerate* ou *description* :

```
\begin{frame}
\frametitle{Exemple d'affichage d'une liste item par item}

\begin{itemize}[<+-->]
\item item 1
\item item 2
\item item 3
\end{itemize}
\end{frame}
```



On peut décider du mode d'animation des items de la liste. En général, on souhaite découvrir les items les uns après les autres, les items suivants étant en filigrane. Voir la section 20.3.6

19.2 Les transitions de transparents

La classe *beamer* propose un certain nombre de transitions de transparents. Tout d'abord, il faut avoir conscience que cela se traduit en langage javascript dans le pdf qui sera généré en compilant votre présentation. Par conséquent, il vous faudra un lecteur pdf qui accepte du javascript. Grosso modo, cela vous demande d'utiliser

Adobe Acrobat Reader, sauf qu'actuellement, ce dernier désactive les transitions sur Mac OS Big Sur ou Mac OS Monterey. En attendant le correctif ...

Les commandes suivantes permettent de définir un type de transition :

\transblindshorizontal transition basée sur l'animation de stores horizontaux

\transblindsvvertical transition basée sur l'animation de stores verticaux

\transboxin transition basée sur une fenêtre qui rétrécit des bords du transparent à son centre

\transboxout transition basée sur une fenêtre qui s'agrandit du centre du transparent à ses bords

\transcover transition basée sur le glissement de la nouvelle page par dessus l'ancienne

\transdissolve transition basée sur la dissolution de l'ancienne page du transparent pour faire apparaître la nouvelle.

\transfade transition basée sur un fondu

\transglitter ressemble beaucoup à **\transdissolve**, mais avec un balayage en plus, dont on peut préciser la direction.

\transpush Transition basée sur la poussée de l'ancienne page du transparent par la suivante.

\transreplace Comportement par défaut, la nouvelle page de transparent s'affiche normalement.

\transsplitverticalin Transition basée sur le déplacement vertical de 2 lignes entrantes, l'une venant du haut, l'autre venant du bas

\transsplitverticalout Transition basée sur le déplacement vertical de 2 lignes sortantes, l'une allant vers le haut, l'autre vers le bas

\transsplithorizontalin Transition basée sur le déplacement horizontal de 2 lignes entrantes, l'une venant de la gauche, l'autre venant de la droite

\transsplithorizontalout Transition basée sur le déplacement horizontal de 2 lignes sortantes, l'une allant vers la gauche, l'autre vers la droite

\transwipe Transition basée sur le déplacement d'une ligne dans une direction donnée, affichant au fur et à mesure la nouvelle page du transparent.

\transduration Ce n'est pas réellement une transition, mais le moyen de spécifier la durée en secondes de la transition vers la page suivante du transparent



Vous pouvez constater que dans cette liste de transitions, il n'y a rien d'extravagant, uniquement l'essentiel et je dirais même le sérieux.

Pour que le défilement des pages d'un transparent soit automatique, quelles que soient les transitions utilisées, on utilise la commande **\animate** avec entre chevrons la séquence de pages à animer.

19.3 Animer une séquence d'images

La classe *beamer* autorise l'insertion de vidéos dans une présentation, tout comme elle permet d'insérer du son!!! Or, dans une présentation à caractère scientifique, les vidéos sont souvent des animations de résultats de simulations, donc des vidéos générées à partir de séquences d'images.

Or il se trouve que l'on peut directement afficher une séquence d'images grâce à la commande **\animategraphics** de l'extension **ANIMATE** (voir GRAHN, *animate – Create PDF and SVG animations from graphics files and inline graphics*).

19.3. ANIMER UNE SÉQUENCE D'IMAGES

```
\begin{frame}  
\frametitle{Exemple d'affichage d'une séquence d'images}  
  
\animategraphics[autoplay, loop, width=4cm]{12}{images/solution.00}{00}{19}  
  
\end{frame}
```

Dans l'exemple ci-dessus, on affiche une série de 20 images dont le nom est de la forme `solution.0000`, `solution.0001`, ...`solution.0019`. L'extension (dans ce cas `.png`) est toujours omise. La largeur des images est 4 cm. L'animation se lance tout de suite et passe en boucle, avec une vitesse de 12 images par seconde.

20.1 Les thèmes prédéfinis

beamer propose une petite trentaine de thèmes graphiques complets. Leurs noms sont ceux de villes dans lesquelles Till Tantau, l'auteur de *beamer*, a participé à une conférence soit personnellement, soit à travers la participation d'un co-auteur.

Pour charger un thème complet, on utilisera la commande `\usetheme`.

En réalité, ces thèmes sont tous composés d'une sélection de thèmes élémentaires. Voici les différentes famille de thèmes élémentaires :

Le thème interne se préoccupe de la définition du contenu : page de garde, listes, blocs, théorèmes, ... Sont disponibles : *circles*, *default*, *inmargin*, *rectangles*, *rounded*. Pour charger un thème interne, on utilisera la commande `\useinnertheme`

Le thème externe se préoccupe de la mise en page des *frame* : entête, pied de transparent, logo, barre latérale, ... Sont disponibles : *default*, *infolines*, *miniframes*, *shadow*, *smoothbars*, *smoothtree*, *split*, *tree*, et *sidebar*. Pour charger un thème externe, on utilisera la commande `\useoutertheme`

Le thème de couleur se préoccupe de la couleur de tous les éléments :

Les thèmes de couleur globaux définissent les couleurs de chaque élément. Sont disponibles : *albatross*, *beaver*, *beetle*, *crane*, *default*, *dove*, *fly*, *monarca*, *seagull*, *spruce*, *sidebartab*, *structure*, et *wolverine*;

Les thèmes de couleur internes définissent les couleurs des éléments internes, à savoir le contenu. Sont disponibles : *lily*, *orchid* et *rose*;

Les thèmes de couleur externes définissent les couleurs des éléments externes, constitutifs des *frame*. Sont disponibles : *dolphin*, *seahorse* et *whale*.

Pour charger un thème de couleur, quel qu'il soit (global, interne ou externe), on utilisera la commande `usecolortheme`

La tableau 20.1 détaille la composition des thèmes prédéfinis.

Thème complet		Thème interne	Thème externe	Thème de couleur		
				complet	interne	externe
default	(figure 20.1)	default	default	default		
AnnArbor	(figure 20.2)	rounded	infolines	wolverine		
Antibes	(figure 20.3)	rectangles	tree		orchid	whale
Bergen		inmargin	default		orchid	
Berkeley	(figure 20.4)	rectangles	sidebar		orchid	whale
Berlin		rectangles	miniframes		orchid	whale
Boadilla		rounded	infolines		rose	dolphin
CambridgeUS		rounded	infolines	beaver		
Copenhagen		rounded	split		orchid	whale
Darmstadt	(figure 20.5)	rounded	smoothbars		orchid	whale
Dresden		default	miniframes			whale
EastLansing		rounded	infolines	spruce		
Frankfurt		rounded	smoothbars		orchid	whale
Goettingen		default	sidebar	structure		
Hannover	(figure 20.6)	circles	sidebar			seahorse
Ilmenau	(figure 20.7)	rounded	miniframes		orchid	whale
JuanLesPins	(figure 20.8)	rounded	smoothtree		orchid	whale
Luebeck	(figure 20.9)	rectangles	split		orchid	whale
Madrid		rounded	infolines		orchid	whale
Malmoe		default	split			whale
Marburg		default	sidebar			whale
Montpellier		default	tree			
PaloAlto		rounded	sidebar		orchid	whale
Pittsburg		circles	default			
Rochester		rectangles	sidebar		orchid	whale
Singapore		default	miniframes			
Szeged		default	miniframes			
Warsaw		rounded	shadow		orchid	whale

TABLE 20.1 – Les thèmes existants et leurs composants

Ma première présentation en \LaTeX
Utiliser Beamer

Vous
ENSTA-Paris
23 octobre 2019

Les blocs
Comment ça marche ?

Bonjour le monde
▶ Ca va bien ?
▶ Oui, et vous ?

Bonjour le monde
1. Ca va bien ?
2. Oui, et vous ?

Bonjour le monde
Vous Ca va bien ?
Moi Oui, et vous ?

FIGURE 20.1 – Exemple de transparents avec le thème *default*

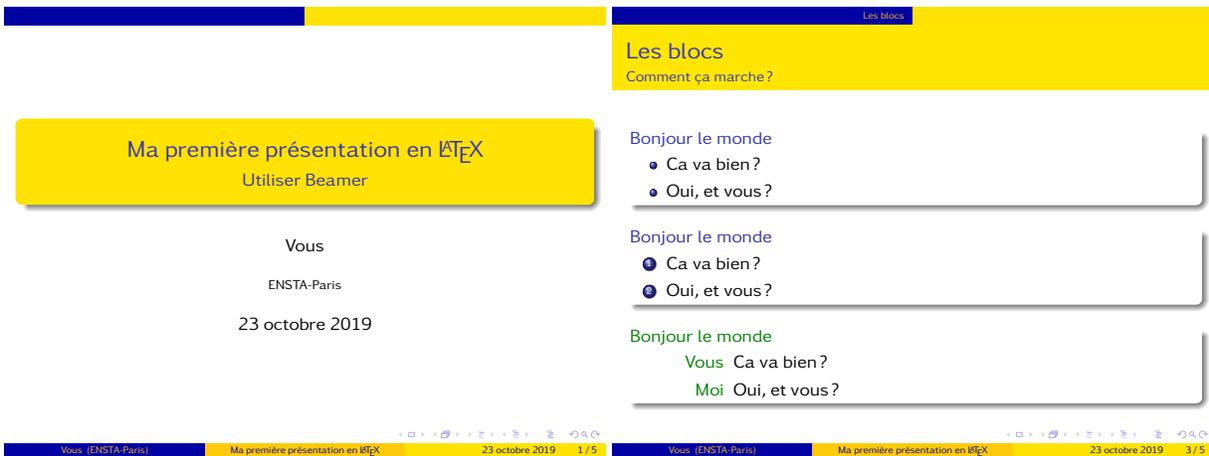


FIGURE 20.2 – Exemple de transparents avec le thème *AnnArbor*

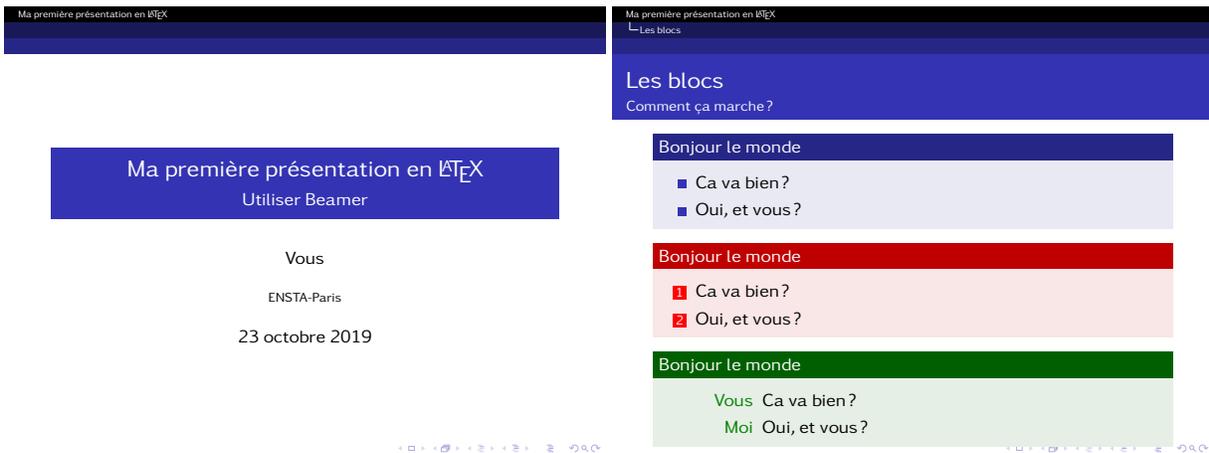


FIGURE 20.3 – Exemple de transparents avec le thème *Antibes*

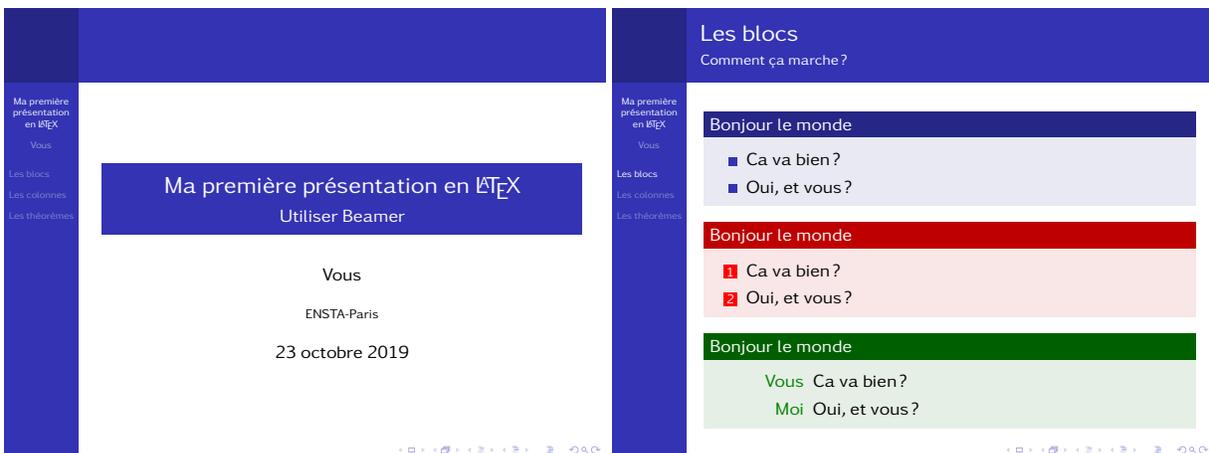


FIGURE 20.4 – Exemple de transparents avec le thème *Berkeley*

20.1. LES THÈMES PRÉDÉFINIS



FIGURE 20.5 – Exemple de transparents avec le thème *Darmstadt*

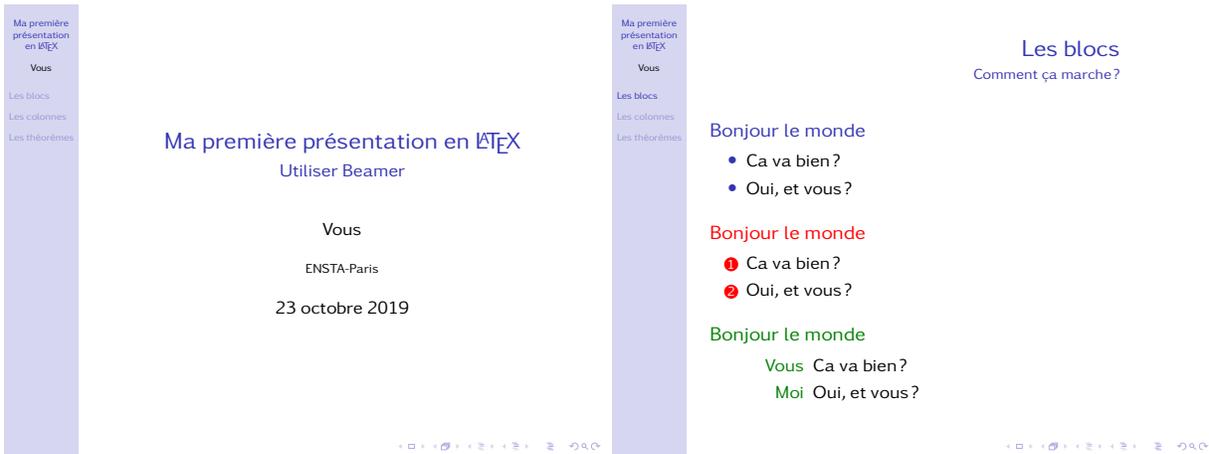


FIGURE 20.6 – Exemple de transparents avec le thème *Hannover*



FIGURE 20.7 – Exemple de transparents avec le thème *Ilmenau*



FIGURE 20.8 – Exemple de transparents avec le thème *JuanLesPins*

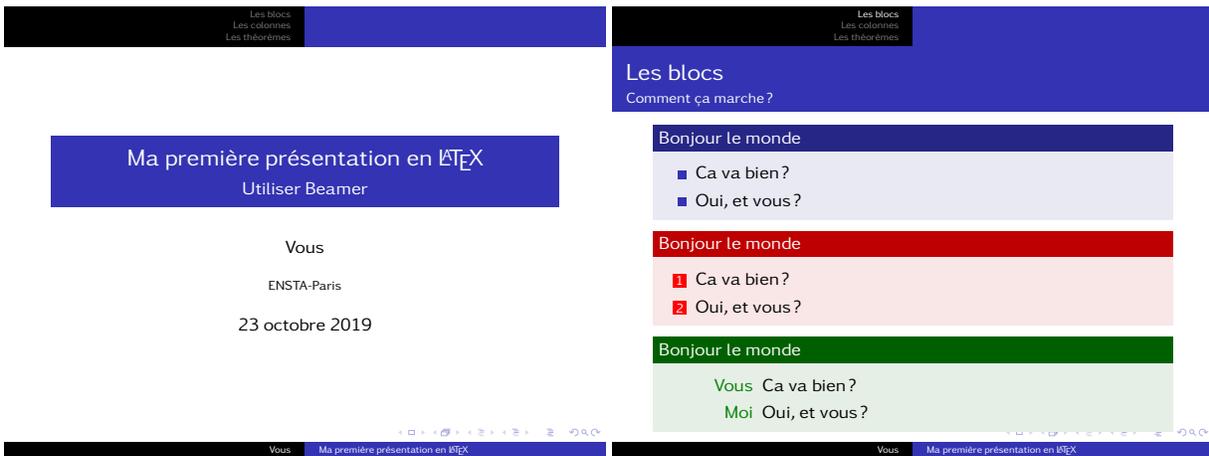


FIGURE 20.9 – Exemple de transparents avec le thème *Luebeck*

20.2 Personnalisation de l'apparence : cas général

Suivant le thème général et / ou les thèmes internes et externes choisis, *beamer* met à disposition un certain nombre de paramètres et d'options pour contrôler l'apparence de chaque élément des transparents. Pour définir / modifier ces templates, tels qu'ils sont nommés, *beamer* met à disposition un certain nombre de commandes :

- `\setbeamertemplate`** permet de définir un template, soit en sélectionnant une option pré-définie, soit en écrivant directement son contenu;
- `\defbeamertemplate`** permet de définir une nouvelle option pour un template donné;
- `\usebeamertemplate`** permet d'exécuter le contenu d'un template;
- `\addtobeamertemplate`** permet d'ajouter du texte avant et après le contenu d'un template;
- `\usebeamercolor`** permet de charger la couleur de texte et la couleur de fond d'un template donné;
- `\setbeamercolor`** permet de modifier la couleur de texte et la couleur de fond d'un template donné;
- `\usebeamerfont`** permet de charger la police de caractères d'un template donné

`\setbeamerfont` permet de modifier la police de caractères d'un template donné.

Ces commandes permettent réellement de tout faire en terme de personnalisation, et on voit que c'est la seconde plus-value de cette classe par rapport à la concurrence. La documentation de *beamer* (TANTAU et al., *beamer – A L^AT_EX class for producing presentations and slides*) est du coup très bien écrite sur ce point. Toutefois, je vais vous donner quelques exemples pratiques dans la section qui suit.

20.3 Personnalisation de l'apparence : solutions usuelles

20.3.1 Changer la couleur principale du thème

Pour changer la couleur principale d'un thème, il suffit de charger le thème de couleur *structure* et de passer la couleur en arguments.

```
% definition d'une couleur (extension xcolor)
\definecolor{macouleur}{RGB}{0,127,127}
...
% utilisation de cette couleur
\usecolortheme[named=macouleur]{structure}
```

20.3.2 Mettre une image en fond de transparent

```
\defbeamertemplate{background}{bg}{%
\includegraphics[width=\paperwidth,height=\paperheight]{backgrounds/bg-subtitle}%
}

\setbeamertemplate{background}[bg]
```

20.3.3 Changer la couleur associée à la commande `\alert`

On souhaite changer la couleur de texte lors de l'utilisation de la commande `\alert`. Le template dédié s'appelle *alerted text*

```
\setbeamercolor*{alerted text}{fg=MediumOrchid}
```

20.3.4 Changer la couleur associée à l'environnement *exampleblock*

On souhaite changer la couleur de la boîte *exampleblock*. Le template dédié s'appelle *example text*

```
\setbeamercolor*{example text}{bg=blue, fg=white}
```

20.3.5 Changer les couleurs du titre d'un transparent

On souhaite changer la couleur du titre d'un transparent, tout en conservant la couleur de fond. Il nous faut donc d'une part charger le thème externe *infolines* (ou un thème global basé sur ce thème externe). Puis charger les couleurs définies pour le template *frametitle* et les modifier ensuite, comme dans l'extrait ci-dessous :

```
\usebeamercolor{frametitle}
\setbeamercolor*{frametitle}{fg=white, bg=frametitle.bg}
```

20.3.6 Définir le comportement de la commande `\uncover`

Si l'on souhaite faire en sorte que le texte contrôlé par `\uncover` soit en filigrane quand il n'est pas censé être affiché, on utilise la commande `\setbeamercovered` avec l'option *transparent* :

```
\setbeamercovered{transparent}
```

20.3.7 Changer les boutons de navigation

Vous avez pu constater la présence de petites icônes en filigranes, généralement en bas à droite de vos transparents. Il s'agit des boutons de navigation. Il y a, entre autres :

- la navigation entre transparents ;
- la navigation entre sections ;
- la navigation entre sous-sections.

On peut bien entendu personnaliser ces boutons avec le paramètre *navigation symbols* :

```
% Pour masquer les boutons de navigations
\setbeamertemplate{navigation symbols}{}
% Pour activer les boutons de navigations dans une barre horizontale
\setbeamertemplate{navigation symbols}[default]
% Pour afficher les boutons de navigation dans une barre horizontale
\setbeamertemplate{navigation symbols}[horizontal]
% Pour afficher les boutons de navigation dans une barre verticale
\setbeamertemplate{navigation symbols}[vertical]
% Pour n'afficher que les boutons de navigation entre transparents
\setbeamertemplate{navigation symbols}[only frame symbol]
```

Cette dernière option existe en réalité parce qu'à l'usage, la navigation entre transparents est de loin la plus utilisée.

C'est ici que se termine ce bref panorama de la classe *beamer*. La présentation associée à ce support de cours viendra compléter cette partie.

```

\usepackage{iftex}
\ifPDFTeX
  \usepackage[utf8]{inputenc}
  \usepackage[T1]{fontenc}
  \usepackage{amssymb}
  \usepackage{fourier}
  \usepackage{mathtools}
\else
  \usepackage{fontspec}
  \usepackage{mathtools}
  \usepackage{fourier-otf}
\fi
\usepackage{geometry}
\usepackage{graphicx}
\usepackage{graphbox} % pour l'alignement vertical
\usepackage{xcolor}
\usepackage{calc}
\usepackage{amsthm}
\usepackage{fancyhdr}
\usepackage{fancyvrb}
\usepackage{enumitem}
\usepackage{tikz}
\usepackage{pgfplots}
\usepackage[style=authortitle]{biblatex}
\usepackage{float}
\usepackage{multirow}
\usepackage{tcolorbox}
\usepackage{etoolbox}
\usepackage{lettrine}
\usepackage{babel}
\usepackage{microtype}
\usepackage{csquotes}
\usepackage{listings}
\usepackage[plainpages=false, pdfpagelabels, bookmarksnumbered, breaklinks=true,
  linkbordercolor=black, linkcolor=black, citecolor=black, colorlinks=true]{hyperref}
\usepackage{cleveref}
\usepackage{minitoc}
\usepackage{todonotes}
\usetikzlibrary{fit, arrows, shadows, intersections, shapes, positioning, patterns, external}

```

Quelques mots sur les extensions utilisées :

INPUTENC concerne l'encodage des caractères. Cela doit se conformer à l'encodage du fichier L^AT_EX que vous êtes en train d'écrire. Il est quand même fortement recommandé de prendre **utf8** (même sous Windows pour lequel peu d'éditeurs de textes sont encore en latin1). Cette extension n'est pas à inclure avec les moteurs Xe_LA_TE_X et Lua_LA_TE_X.

FONTEC concerne l'encodage de la police de caractères. *T1* est l'encodage le plus couramment utilisé. C'est pour dire que l'on utilise des polices de caractères de type 1. Cette extension n'est pas à inclure avec les moteurs Xe_LA_TE_X et Lua_LA_TE_X.

FONTSPEC Cette extension concerne la gestion des polices de caractères pour les moteurs Xe_LA_TE_X et Lua_LA_TE_X. Voir ROBERTSON, *fontspec – Advanced font selection in XeLaTeX and LuaLaTeX*

AMSSYMB Cette extension charge des symboles mathématiques et introduit notamment la commande **\mathbb** (vue en section 6.3). Si utilisée conjointement avec l'extension **FOURIER**, comme c'est le cas dans

ce document, il faut la charger avant. Avec d'autres polices, comme **KPFFONTS** ou **FOURIER-OTF**, il ne faut pas la charger du tout.

FOURIER charge la police de caractères **FOURIER**, adaptée de Adobe Utopia par l'association GUTenberg¹. C'est l'une des polices recommandées avec **KPFFONTS**, et éventuellement **LIBERTINE** ou **LMODERN**. A utiliser avec le moteur pdf \TeX . Voir BOVANI, *fourier – Using Utopia fonts in LATEX documents*

FOURIER-OTF charge la police de caractères Erewhon-math, adaptée de **FOURIER** pour être utilisée avec les moteurs Xe \TeX et Lua \TeX . Voir FLIPO, *Erewhon-math – Utopia based OpenType Math font*

MATHTOOLS Cette extension est dédiée à la mise en forme des environnements mathématiques. Elle corrige et complète l'extension **AMSMATH** qu'elle charge automatiquement. Elle est à charger après **FOURIER**, mais avant **FOURIER-OTF**. Voir MADSEN et al., *mathtools – Mathematical tools to use with amsmath*

GEOMETRY définit les marges d'une page, comme montré dans l'exemple en début de chapitre. Voir CARLISLE et UMEKI, *geometry – Flexible and complete interface to document dimensions*;

GRAPHICX Cette extension concerne l'inclusion des images, et les formats d'images pris en comptes. Voir CARLISLE, *graphicx – Enhanced support for graphics*

GRAPHBOX Cette extension augmente les possibilités d'inclusion d'images proposées par **GRAPHICX**, avec notamment l'alignement vertical. Voir BEISERT, *graphbox – Extend graphicx to improve placement of graphics*

XCOLOR Cette extension concerne la définition des couleurs, avec la possibilité de les mélanger de manière simple. Voir KERN, *xcolor – Driver-independent color extensions for \TeX and pdf \TeX*

CALC Cette extension permet de réaliser des calculs arithmétiques. Voir THORUP, JENSEN et TEAM, *calc – Simple arithmetic in \TeX commands*

AMSTHM C'est l'extension à charger pour définir des théorèmes. Voir *amsthm – Typesetting theorems (AMS style)*. Il existe une extension additionnelle **THMTOOLS** qui simplifie la personnalisation de l'apparence des théorèmes. Tout comme **TCOLORBOX** (voir plus bas dans la liste) ...

FANCYHDR Cette extension est dédiée à la personnalisation des entêtes et pieds de page. Voir OOSTRUM, *fancyhdr – Extensive control of page headers and footers in \TeX 2e*

FANCYVRB Cette extension augmente les possibilités de manipulation des contenus *verbatim*. Je l'utilise conjointement avec **TCOLORBOX** pour pouvoir définir des propres environnements. Voir ZANDT, *fancyvrbs – Sophisticated verbatim text*

FVEXTRA Cette extension augmente les capacités de l'extension **FANCYVRB** en permettant des passages à la ligne automatiques pour les lignes très longues. Voir POORE, *fvextra – Extensions and patches for fancyvrbs*

ENUMITEM Extension qui permet de définir complètement le style des listes (*itemize*, *enumerate* et *description*). Voir LÓPEZ, *enumitem – Control layout of itemize, enumerate, description*

TIKZ Extension incontournable dédiée au graphisme et au dessin. Elle permet aussi de tracer des courbes à partir de fichiers de coordonnées, des arbres, des fractales et bien d'autres choses encore. TikZ est la surcouche utilisateur de l'extension **PGF**. Voir FEUERSÄNGER, MENKE et TANTAU, *pgf – Create PostScript and PDF graphics in \TeX* pour accéder à leur documentation extrêmement complète et bien rédigée. Si je vous dis qu'elle fait plus de 1000 pages, vous aurez un aperçu de la puissance de cette extension.

PGFPLOTS Cette extension a pour but de faciliter davantage les tracés de fonctions par rapport à ce que propose déjà l'extension **TIKZ** sur laquelle elle s'appuie. Voir FEUERSÄNGER, *pgfplots – Create normal/logarithmic plots in two and three dimensions*

BIBLATEX C'est l'extension qui gère la bibliographie. Elle a été conçue pour rassembler et proposer davantage encore que les plupart des extensions qui existaient auparavant. Voir KIME et LEHMAN, *BibLaTeX – Sophisticated Bibliographies in \TeX*

1. Le groupe francophone des utilisateurs de \TeX

- FLOAT** Cette extension augmente les possibilités de positionnement des environnements flottants, notamment quand on veut absolument forcer un environnement flottant à être placé dans le corps de texte là où on l'a défini. Voir LINGNAU, *float – Improved interface for floating objects*
- MULTIROW** Cette extension ajoute la commande `\multirow` pour définir des cellules de tableaux sur plusieurs lignes. Voir OOSTRUM et LEICHTER, *multirow – Create tabular cells spanning multiple rows*
- TCOLORBOX** Cette extension très puissante permet de définir des boîtes de couleurs autour de presque n'importe quoi. Elle permet aussi de redéfinir des équivalents de listings et de théorèmes. ... Voir STURM, *tcolorbox – Coloured boxes, for L^AT_EX examples and theorems, etc.* Elle est utilisée abondamment dans ce document (voir la section 9.3 ci-après pour plus d'informations pratiques).
- ETOOLBOX** Cette extension permet des manipulations diverses de bas niveaux, comme l'introduction de commande de test (`\ifstrequal`, `\ifnumcomp`, ...). Je l'ai utilisée dans ce document pour la mise en forme de ma liste d'exemples (extension **TCOLORBOX**) en souhaitant séparer par des lignes blanches les exemples pour matérialiser les chapitres. Voir WRIGHT et LEHMAN, *etoolbox – e- $T_{E}X$ tools for L^AT_EX*
- LETRINE** Cette extension est dédiée, comme son nom l'indique, à la manipulation de lettrines. Elle est utilisée ici pour positionner les images (loupe, ampoule, ...) dans les boîtes de dialogues. Voir FLIPO, *lettrine – Typeset dropped capitals*
- PDFPAGES** Cette extension introduit une commande `\includepdf` qui permet d'inclure une ou plusieurs pages d'un autre document pdf, comme si ces pages faisaient partie du document courant. Voir POORE, *fvextra – Extensions and patches for fancyvrb*
- MORESIZE** Cette extension introduit une commande de taille de caractères supplémentaire : `\HUGE`, utilisée dans les présentations associées pour les watermarks. Voir CORNELSEN, *moresize – Allows font sizes up to 35.83pt*
- PGFOPTS** Cette extension est dédiée à la définition des arguments optionnels du chargement d'une extensions. Il y en a d'autres mais j'utilise celle-ci parce qu'elle est liée à l'extension **TIKZ**. Voir WRIGHT, *pgfopts – L^AT_EX package options with pgfkeys*
- BABEL** Cette extension charge les préférences de la langue dans laquelle vous allez écrire votre document. En effet, chaque langue a ses propres règles typographiques et cette extension est là pour les prendre en compte à la génération du document. Voir LÓPEZ et BRAAMS, *babel – Multilingual support for Plain T_EX or L^AT_EX*
- MICROTYPE** Cette extension apporte des raffinements d'ordre typographique, comme l'ajustement de l'espacement permettant une meilleure coupure des mots. Le simple fait de charger cette extension suffit à avoir des améliorations notables. Voir SCHLICHT, *microtype – Subliminal refinements towards typographical perfection* pour plus de détails.
- CSQUOTES** Cette extension est dédiée à améliorer la gestion et la personnalisation des citations. Elle est notamment requise par l'extension **BIBLATEX**. Voir WRIGHT et LEHMAN, *csquotes – Context sensitive quotation facilities*
- LISTINGS** Cette extension est à l'heure actuelle l'extension la plus complète pour insérer des listings de code, avec un nombre élevé de langages pré-définis, et la possibilité de définir ses propres langages pour la mise en page d'un listing. Voir HOFFMANN, HEINZ et MOSES, *listings – Typeset source code listings using L^AT_EX*. Son seul véritable défaut est la non prise en compte des caractères accentués sauf dans des cas particuliers si l'on utilise le moteur pdf $T_{E}X$. TOut va bien par contre avec lua $T_{E}X$.
- HYPERRREF** Cette extension est dédiée aux références croisées et à la possibilité de mettre des liens cliquables dans des documents. Voir RAHTZ et OBERDIEK, *hyperref – Extensive support for hypertext in L^AT_EX*.
- CLEVEREF** Cette extension complète et simplifie quelque peu l'utilisation de l'extension **HYPERRREF**. Elle est utilisée ici pour pouvoir référencer mes boîtes de couleurs numérotées (extension **TCOLORBOX**). Voir CUBITT, *cleveref – Intelligent cross-referencing*

MINITOC Cette extension est dédiée à la construction de tables des matières partielles attachées à une partie, un chapitre ou une section. Voir DRUCBERT, *minitoc – Produce a table of contents for each chapter, part or section*

TODONOTES Cette extension permet de manipuler 2 commandes `\todo` et `\missingfigure` pour indiquer ce que l'on a prévu d'écrire à tel ou tel endroit dans le document en cours de rédaction. Le grand intérêt est de pouvoir afficher une liste de ces éléments avec la commande `\listoftodos` de sorte à savoir si on n'a effectivement rien oublié avant de finaliser notre document et le diffuser. Très pratique pour de gros documents comme ce support de cours. Voir MIDTIBY, *todonotes – Marking things to do in a L^AT_EX document*

- amsthm* – *Typesetting theorems (AMS style)*. The American Mathematical Society. 1995. URL : <https://ctan.org/pkg/amsthm>.
- BARNARD, Richard et al. *tikzposter* – *Create scientific posters using TikZ*. 2014. URL : <https://www.ctan.org/pkg/tikzposter>.
- BAUDOIN, Marc. *Apprends L^AT_EX (5ème édition)*. 2012. URL : http://www.babafou.eu.org/Apprends_LaTeX/Apprends_LaTeX.pdf.
- BECCARI, Claudio et Enrico GREGORIO. *imakeidx* – *A package for producing multiple indexes*. 2010. URL : <https://www.ctan.org/pkg/imakeidx>.
- BEISERT, Niklas. *graphbox* – *Extend graphicx to improve placement of graphics*. 2013. URL : <https://www.ctan.org/pkg/graphbox>.
- BOVANI, Michel. *fourier* – *Using Utopia fonts in L^AT_EX documents*. 2022. URL : <https://ctan.org/pkg/fourier>.
- CAIGNAERT, Christophe. *kpfonts* – *A complete set of fonts for text and mathematics*. 2022. URL : <https://ctan.org/pkg/kpfonts>.
- CARLISLE, David. *graphicx* – *Enhanced support for graphics*. 1994. URL : <https://ctan.org/pkg/graphicx>.
- *keyval* – *Process 'key=value' schemes*. The L^AT_EX Team. 1993. URL : <https://www.ctan.org/pkg/keyval>.
- CARLISLE, David, Morten HOGHOLM et The L^AT_EX TEAM. *xspace* – *Define commands that appear not to eat spaces*. 1991. URL : <https://ctan.org/pkg/xspace>.
- CARLISLE, David, Leslie LAMPORT et The L^AT_EX TEAM. *ifthen* – *Conditional commands in L^AT_EX documents*. 1993. URL : <https://www.ctan.org/pkg/ifthen>.
- CARLISLE, David et Hideo UMEKI. *geometry* – *Flexible and complete interface to document dimensions*. 1996. URL : <https://ctan.org/pkg/geometry>.
- CORNELSEN, Christian. *moresize* – *Allows font sizes up to 35.83pt*. 1997. URL : <https://www.ctan.org/pkg/moresize>.
- CUBITT, Toby. *cleveref* – *Intelligent cross-referencing*. 2006. URL : <https://www.ctan.org/pkg/cleveref>.
- DANAUX, Xavier. *moderncv* – *A modern curriculum vitae class*. 2006. URL : <https://www.ctan.org/pkg/moderncv>.
- DESELAERS, Thomas et Philippe DREUW. *beamerposter* – *Extend beamer and a0poster for custom sized posters*. 2015. URL : <https://www.ctan.org/pkg/beamerposter>.
- DRUCBERT, Jean-Pierre. *minitoc* – *Produce a table of contents for each chapter, part or section*. 1993. URL : <https://ctan.org/pkg/minitoc>.
- FEUERSÄNGER, Christian. *pgfplots* – *Create normal/logarithmic plots in two and three dimensions*. 2007. URL : <https://www.ctan.org/pkg/pgfplots>.
- FEUERSÄNGER, Christian, Henri MENKE et Till TANTAU. *pgf* – *Create PostScript and PDF graphics in T_EX*. 2005. URL : <https://www.ctan.org/pkg/pgf>.
- FLIPO, Daniel. *De pdfLaTeX à LuaLaTeX*. 2022. URL : <http://daniel.flipo.free.fr/doc/luatex/pdf2lua.pdf>.
- *Erewhon-math* – *Utopia based OpenType Math font*. 2022. URL : <https://ctan.org/pkg/erewhon-math>.
- *kpfonts-otf* – *OpenType version of the kpfonts (Type1) designed by Christophe Caignaert*. 2022. URL : <https://ctan.org/pkg/kpfonts-otf>.
- *lettrine* – *Typeset dropped capitals*. 1999. URL : <https://www.ctan.org/pkg/lettrine>.
- fontenc* – *Standard package for selecting font encodings*. URL : <https://ctan.org/pkg/fontenc>.

- GRAHN, Alexander. *animate* – Create PDF and SVG animations from graphics files and inline graphics. 2007. URL : <https://www.ctan.org/pkg/animate>.
- GRAY, Norman. *textpos* – Place boxes at arbitrary positions on the \LaTeX page. 1999. URL : <https://www.ctan.org/pkg/textpos>.
- HOFFMANN, Jobst, Carsten HEINZ et Brooks MOSES. *listings* – Typeset source code listings using \LaTeX . 1996. URL : <https://ctan.org/pkg/listings>.
- iftex* – Am I running under pdfTeX, XeTeX or LuaTeX? The LaTeX Team. 2022. URL : <https://ctan.org/pkg/iftex>.
- JACKOWSKI, Bogusław et Janusz Marian NOWACKI. *lm* – Latin modern fonts in outline formats. 2021. URL : <https://www.ctan.org/pkg/lm>.
- JEFFREY, Alan et Frank MITTELBACH. *inputenc* – Accept different input encodings. 2021. URL : <https://ctan.org/pkg/inputenc>.
- KERN, Uwe. *xcolor* – Driver-independent color extensions for \LaTeX and pdf \LaTeX . 2003. URL : <https://ctan.org/pkg/xcolor>.
- KIME, Philip et Philipp LEHMAN. *BibLaTeX* – Sophisticated Bibliographies in \LaTeX . 2006. URL : <https://ctan.org/pkg/biblatex>.
- LINGNAU, Anselm. *float* – Improved interface for floating objects. 2001. URL : <https://www.ctan.org/pkg/float>.
- LÓPEZ, Javier Bezos. *enumitem* – Control layout of itemize, enumerate, description. 2003. URL : <https://www.ctan.org/pkg/enumitem>.
- *titlesec* – Select alternative section titles. 1998. URL : <https://ctan.org/pkg/titlesec>.
- LÓPEZ, Javier Bezos et Johannes L. BRAAMS. *babel* – Multilingual support for Plain \TeX or \LaTeX . 1989. URL : <https://www.ctan.org/pkg/babel>.
- MADSEN, Lars et al. *mathtools* – Mathematical tools to use with *amsmath*. 2002. URL : <https://ctan.org/pkg/mathtools>.
- MATTHIAS, Andreas. *pdfpages* – Include PDF documents in \LaTeX . 2001. URL : <https://www.ctan.org/pkg/pdfpages>.
- MIDTIBY, Henrik Skov. *todonotes* – Marking things to do in a \LaTeX document. 2008. URL : <https://www.ctan.org/pkg/todonotes>.
- MITTELBACH, Frank et Michel GOOSSENS. *The \LaTeX Companion*. 2st. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 2004. ISBN : 978-0-201-36299-2.
- NETHERTON, Lee et al. *nomencl* – Produce lists of symbols as in nomenclature. 1996. URL : <https://www.ctan.org/pkg/nomencl>.
- OOSTRUM, Piet van. *fancyhdr* – Extensive control of page headers and footers in $\text{\LaTeX}2\epsilon$. 1994. URL : <https://ctan.org/pkg/fancyhdr>.
- OOSTRUM, Piet van et Jerry LEICHTER. *multirow* – Create tabular cells spanning multiple rows. 1994. URL : <https://www.ctan.org/pkg/multirow>.
- PAKIN, Scott. *The Comprehensive \LaTeX Symbol List* – Symbols accessible from \LaTeX . 1999. URL : <https://www.ctan.org/pkg/comprehensive>.
- POORE, Geoffrey. *fvextra* – Extensions and patches for *fancyvrb*. 2016. URL : <https://www.ctan.org/pkg/fvextra>.
- RAHTZ, Sebastian et Heiko OBERDIEK. *hyperref* – Extensive support for hypertext in \LaTeX . The LaTeX Team. 1995. URL : <https://ctan.org/pkg/hyperref>.
- REDAELLI, Massimo et al. *CircuitikZ* – Draw electrical networks with TikZ. 2007. URL : <https://ctan.org/pkg/circuitikz>.
- REICHERT, Axel. *currvita* – Typeset a curriculum vitae. 1999. URL : <https://www.ctan.org/pkg/currvita>.
- ROBERTSON, Will. *fontspec* – Advanced font selection in XeLaTeX and LuaLaTeX. 2022. URL : <https://ctan.org/pkg/fontspec>.
- ROBERTSON, Will et al. *unicode-math* – Unicode mathematics support for XeTeX and LuaTeX. 2020. URL : <https://ctan.org/pkg/unicode-math>.

- SCHLICHT, Robert. *microtype* – Subliminal refinements towards typographical perfection. 2004. URL : <https://ctan.org/pkg/microtype>.
- SCHMIDT, Walter. *helvet* – Load Helvetica, scaled. 2020. URL : <https://www.ctan.org/pkg/helvet>.
- STURM, Thomas F. *tcolorbox* – Coloured boxes, for \LaTeX examples and theorems, etc. 2006. URL : <https://ctan.org/pkg/tcolorbox>.
- TANTAU, Till. *pgfkeys* – Key value control for PGF. 2005. URL : <https://www.ctan.org/pkg/pgfkeys>.
- TANTAU, Till et al. *beamer* – A \LaTeX class for producing presentations and slides. 2003. URL : <https://ctan.org/pkg/beamer>.
- TELLECHEA, Christian. *chemfig* – Draw molecules with easy syntax. 2010. URL : <https://www.ctan.org/pkg/chemfig>.
- TENNENT, Bob. *libertine* – Use of Linux Libertine and Biolinum fonts with \LaTeX . 2022. URL : <https://www.ctan.org/pkg/libertine>.
- THORUP, Kresten Krab, Frank JENSEN et The \LaTeX TEAM. *calc* – Simple arithmetic in \LaTeX commands. 1992. URL : <https://www.ctan.org/pkg/calc>.
- TSOLOMITIS, Antonis. *NewComputerModern* – Computer Modern fonts including matching non-latin alphabets. 2021. URL : <https://www.ctan.org/pkg/newcomputermodern>.
- VERNA, Didier. *CurVe* – A class for making curriculum vitae. 2000. URL : <https://www.ctan.org/pkg/curve>.
- VOB, Herbert. *libertinus-otf* – Support for Libertinus OpenType. 2021. URL : <https://www.ctan.org/pkg/libertinus-otf>.
- WRIGHT, Joseph. *pgfopts* – \LaTeX package options with *pgfkeys*. 2008. URL : <https://www.ctan.org/pkg/pgfopts>.
- WRIGHT, Joseph et Philipp LEHMAN. *csquotes* – Context sensitive quotation facilities. 2003. URL : <https://www.ctan.org/pkg/csquotes>.
- *etoolbox* – \LaTeX tools for \LaTeX . 2007. URL : <https://www.ctan.org/pkg/etoolbox>.
- ZANDT, Timothy Van. *fancyvrb* – Sophisticated verbatim text. 1992. URL : <https://www.ctan.org/pkg/fancyvrb>.
- ZOONEKYND, Vincent. *LaTeX*. URL : <http://zoonek.free.fr/LaTeX/>.

Symboles

% 17, 53

A

accents mathématiques 35

alternative mathématique 37

B

bib_TE_X 55

biber 56

bibliographie 54

bibtex 56

C

calque de transparent 79

classe 17

classes

 article 17, 23, 24

 beamer 18, 71

 book 18, 24

 currvita 44

 curve 44

 moderncv 44

 report 18, 23, 24

 tikzposter 18

commande 17

commentaire 17, 53

D

déclaration 17

délimiteur 32

E

environnement 17

environnements flottants 41, 75

exposant 31

extension 18

extensions

 amsmath voir *extensions* - mathtools

 amssymb 20, 31, 33, 90

 amsthm 20, 38, 76, 91

animate 81

babel 20, 21, 92

beamer 71

biblatex 21, 56, 91

calc 91

cleveref 60, 61, 92

csquotes 92

custom-bib 55

enumitem 29, 91

etoolbox 49, 51, 92

fancyhdr 25, 91

fancyvrb 91

float 41, 92

fontenc 90

fontspec 90

fourier 72, 91

fourier-otf 91

fvextra 91

geometry 20, 71, 91

graphbox 91

graphicx 20, 39, 91

hyperref 21, 60, 77, 92

iftex 21

imakeidx 63, 64

inputenc 90

keyval 53

lettrine 27, 92

listings 21, 42, 92

mathtools 20, 31, 91

microtype 20, 21, 92

minitoc 62, 93

moresize 92

multirow 41, 92

natbib 55

nomenc1 58

pdfpages 92

pgfkeys 53, 69

pgfplots 69, 92

pgfplots 91

polices

 amsfonts 19, 33

fontenc 19
 fontspec 19
 fourier 19, 31–33
 fourier-otf 20, 31–33
 helvet 19
 inputenc 19
 kpfonts 19, 31, 33
 kpfonts-otf 20
 libertine 19
 libertine-otf 20
 lmodern 19
 mathdesign 19
 mathpazo 19
 newcomputermmodern 20
 unicode-math 20
 slides 71
 tcolorbox 38, 50, 61, 92
 thmtools 38
 tikz 71, 91
 titlesec 52
 todonotes 65, 93
 unicode-math 33
 xcolor 20, 27, 77, 91

F

fichier de style 68
 flèches 35
 fonctions logarithmiques 32
 fonctions trigonométriques 32

I

images 39
 index 63
 indice 31

L

lettres grecques 34
 Lua \TeX 11, 19

M

makeindex 63

N

nomenclature 58

P

package *voir* extension
 pdf \TeX 11, 19
polices
 à chasse fixe 26, 33
 calligraphique 33
 droit 26
 ensembliste 33
 gothique 33
 gras 26, 33
 incliné 26
 italique 26, 33
 petites capitales 26
 romain 26, 33
 sans empattement 26, 33
 script 33
 préambule 18, 23
 présentation 71

R

référence croisée 60, 77

S

symboles d'opérateurs 34
 symboles de relation 34
 symboles mathématiques 34
 système d'équations 36, 37

T

tableau 39
 \TeX 11
 théorème 38, 76

X

Xe \TeX 11, 19

Index des commandes et environnements

Symboles

<code>\@ifstar</code>	52
<code>\\</code>	28

A

<code>\addbibresource</code>	57
<code>\addcontentsline</code>	24
<code>\address</code>	44
<code>\addtobeamertemplate</code>	87
<code>\alert</code>	77, 80
<code>alertblock</code>	75
<code>align</code>	36, 37
<code>aligned</code>	36, 37
<code>\animate</code>	81
<code>\animategraphics</code>	81
<code>\appendix</code>	24, 74
<code>array</code>	41
<code>\AtBeginPart</code>	74
<code>\AtBeginSection</code>	74
<code>\author</code>	23, 73
<code>\autocite</code>	57
<code>\autoref</code>	60

B

<code>\beamerbutton</code>	78
<code>\beamergetobutton</code>	78
<code>\beamerreturnbutton</code>	78
<code>\beamerskipbutton</code>	78
<code>\bigskip</code>	28
<code>block</code>	75

C

<code>\caption</code>	42, 60
<code>cases</code>	37
<code>center</code>	29
<code>\centering</code>	29
<code>\chapter</code>	24, 52
<code>\Cite</code>	57
<code>\cite</code>	57
<code>\cline</code>	40
<code>\color</code>	80

<code>column</code>	76
<code>columns</code>	76
<i>couleurs</i>	
<code>\color</code>	26
<code>\colorlet</code>	27
<code>\definecolor</code>	27
<code>\textcolor</code>	27
<code>\Cpageref</code>	61
<code>\cpageref</code>	61
<code>\Cref</code>	61
<code>\cref</code>	61
<code>\Crefrange</code>	61
<code>\crefrange</code>	61
<code>\cventry</code>	45
<code>\cvitem</code>	46
<code>\cvitemwithcomment</code>	46
<code>\cvlistdoubleitem</code>	46
<code>\cvlistitem</code>	46

D

<code>\date</code>	23, 73
<code>dcases</code>	38
<code>\defbeamertemplate</code>	87
<code>description</code>	29, 75, 80
<code>displaymath</code>	31
<code>\displaystyle</code>	32
<code>document</code>	18
<code>\documentclass</code>	17
<code>\dominitoc</code>	62
<code>\doparttoc</code>	62
<code>\dosectoc</code>	62

E

<code>\email</code>	44
<code>\emph</code>	26, 80
<code>enumerate</code>	29, 75, 80
<code>\eqref</code>	60, 77
<code>equation</code>	31, 35, 58
<code>exampleblock</code>	75

F

<code>figure</code>	41
---------------------------	----

INDEX DES COMMANDES ET ENVIRONNEMENTS

<code>flushleft</code>	29
<code>flushright</code>	29
<code>\footcite</code>	57
<code>\footcitetext</code>	57
<code>\footnote</code>	29
<code>\footnotemark</code>	30
<code>\footnotesize</code>	25
<code>\footnotetext</code>	30
<code>\frac</code>	31
<code>frame</code>	73
<code>\frame</code>	73
<code>\framesubtitle</code>	73
<code>\frametitle</code>	73

H

<code>\hline</code>	40
<code>\homepage</code>	44
<code>\hspace</code>	28
<code>\Huge</code>	25
<code>\huge</code>	25
<code>\hyperlink</code>	78

I

<code>\include</code>	68
<code>\includegraphics</code>	39, 80
<code>\includeonly</code>	68
<code>\input</code>	68
<code>\insertpart</code>	74
<code>\insertsection</code>	74
<code>\institute</code>	73
<code>itemize</code>	29, 75, 80

L

<code>\label</code>	60, 78
<code>\LARGE</code>	25
<code>\Large</code>	25
<code>\large</code>	25
<code>\left</code>	32
<code>\lettrine</code>	27
<code>\listoffigures</code>	42
<code>\listoftables</code>	42
<code>\listoftodos</code>	65
<code>\linline</code>	43
<code>\lstinputlisting</code>	43
<code>lstlisting</code>	42, 51

M

<code>\makeatletter</code>	52
----------------------------------	----

<code>\makeatother</code>	52
<code>\makecvttitle</code>	45
<code>\makeindex</code>	63
<code>\makenomenclature</code>	58, 59
<code>\maketitle</code>	23, 73
<code>math</code>	31
<code>\mbox</code>	31
<code>\medskip</code>	28
<code>\minitoc</code>	62
<code>\missingfigure</code>	65
<code>\multicolumn</code>	40
<code>\multirow</code>	41

N

<code>\name</code>	44
<code>\newcommand</code>	49, 51
<code>\newenvironment</code>	51
<code>\newrobustcmd</code>	49
<code>\newtcblisting</code>	51
<code>\newtcbtheorem</code>	51
<code>\newtcolorbox</code>	50
<code>\newtheorem</code>	38
<code>\nomenclature</code>	58, 59
<code>\nomgroup</code>	59
<code>\nomname</code>	58
<code>\nompreamble</code>	58
<code>\nomrefeq</code>	58
<code>\nomrefeqpage</code>	58
<code>\nomrefpage</code>	58
<code>\nonumber</code>	36
<code>\normalsize</code>	25

O

<code>\only</code>	79
<code>\overbrace</code>	35

P

<code>\pageref</code>	60
<code>\pagestyle</code>	24
<code>\paragraph</code>	24
<code>\Parencite</code>	57
<code>\parencite</code>	57
<code>\part</code>	24, 52, 73
<code>\parttoc</code>	62
<code>\pgfkeys</code>	53
<code>\phone</code>	44
<code>\photo</code>	44
<i>polices</i>	

<code>\bfseries</code>	26
<code>\itshape</code>	26
<code>\mathbb</code>	33
<code>\mathbf</code>	33
<code>\mathcal</code>	33
<code>\mathfrak</code>	33
<code>\mathit</code>	33
<code>\mathrm</code>	33
<code>\mathscr</code>	33
<code>\mathsf</code>	33
<code>\mathtt</code>	33
<code>\rmfamily</code>	26
<code>\scshape</code>	26
<code>\sffamily</code>	26
<code>\slshape</code>	26
<code>\symbb</code>	33
<code>\symbf</code>	33
<code>\symcal</code>	33
<code>\symfrak</code>	33
<code>\symit</code>	33
<code>\symscr</code>	33
<code>\symsf</code>	33
<code>\symtt</code>	33
<code>\symup</code>	33
<code>\textbf</code>	26
<code>\textit</code>	26
<code>\textrm</code>	26
<code>\textsc</code>	26
<code>\textsf</code>	26
<code>\textsl</code>	26
<code>\texttt</code>	26
<code>\textup</code>	26
<code>\ttfamily</code>	26
<code>\upshape</code>	26
<code>\printbibliography</code>	57
<code>\printindex</code>	63
<code>\printnomenclature</code>	58
<code>proof</code>	38
<code>\protect</code>	49

Q

<code>\quote</code>	44
---------------------------	----

R

<code>\ref</code>	60, 77
<code>\renewcommand</code>	51
<code>\renewenvironment</code>	51
<code>\right</code>	32

S

<code>\scriptscriptstyle</code>	32
<code>\scriptsize</code>	25
<code>\scriptstyle</code>	32
<code>\section</code>	24, 45, 73
<code>\sectoc</code>	62
<code>\setbeamercolor</code>	87
<code>\setbeamerfont</code>	88
<code>\setbeamertemplate</code>	87
<code>\setlist</code>	29
<code>\setuptodonotes</code>	65
<code>\small</code>	25
<code>\smallskip</code>	28
<code>\social</code>	44
<code>\sqrt</code>	32
<code>\structure</code>	77, 80
<code>\subparagraph</code>	24
<code>\subsection</code>	24, 45, 74
<code>\subsubsection</code>	24, 74
<code>\subtitle</code>	73

T

<code>table</code>	41
<code>\tableofcontents</code>	24, 62, 74
<code>tabular</code>	39
<code>\tcbbreak</code>	51
<code>\text</code>	31
<code>\textstyle</code>	32
<code>\thechapter</code>	52
<code>\thepart</code>	52
<code>\TINY</code>	77
<code>\Tiny</code>	77
<code>\tiny</code>	25
<code>\title</code>	23, 73
<code>\todo</code>	65
<code>\transblindshorizontal</code>	81
<code>\transblindsvertical</code>	81
<code>\transboxin</code>	81
<code>\transboxout</code>	81
<code>\transcover</code>	81
<code>\transdissolve</code>	81
<code>\transduration</code>	81
<code>\transfade</code>	81
<code>\transglitter</code>	81
<code>\transpush</code>	81
<code>\transreplace</code>	81
<code>\transsplithorizontalin</code>	81
<code>\transsplithorizontalout</code>	81

\transsplitverticalin 81
 \transsplitverticalout 81
 \transwipe 81

U

\uncover 79
 \underbrace 35
 \url 60
 \usebeamercolor 87
 \usebeamerfont 87
 \usebeamertemplate 87
 \usecolortheme 83

\usefonttheme 72
 \useinnertheme 83
 \useoutertheme 83
 \usetheme 83

V

\value 30
 \verb 42
 verbatim 42
 \visible 79
 \vspace 28