

L^AT_EX HowTo

Version PDF gratuite — <http://www.latex-howto.be/>

SÉBASTIEN COMBÉFIS

L^AT_EX HowTo

Version PDF gratuite — <http://www.latex-howto.be/>

 **UKO** Publication

L^AT_EX HowTo
par Sébastien Combéfis

Copyright © 2010 Sébastien Combéfis.

Ce livre a été entièrement composé sous le système d'exploitation Mac OS X 10.6.5, grâce au système de composition L^AT_EX 2_ε et a été compilé avec pdfT_EX 3.1415926-1.40.11 (T_EX Live 2010).

La photo de couverture provient de http://commons.wikimedia.org/wiki/File:Tafoni_by_Endico.jpg et représente des Tafonis. Les autres images proviennent de <http://www.openclipart.org/>.

Première édition, décembre 2010.

Table des matières

Avant-propos	viii
Introduction	1
1 Structurer le document	5
1.1 Type de document	5
1.2 Page de titre	6
1.3 Sectionnement	10
1.4 Table des matières et liste des figures et tables	15
2 Style et mise en page	19
2.1 Alignement et positionnement	19
2.2 Césure	23
2.3 Style	24
2.4 Mise en page avancée	29
2.5 Dimensions et format de la page	35
3 Listes	41
3.1 Insérer une liste	41
3.2 Style des listes	43
3.3 Gérer la numérotation	46
3.4 Définir une liste	47
3.5 Mise en page avancée	49
4 Images	53

4.1	Insérer une image	53
4.2	Positionnement	58
4.3	Gestion des images	62
4.4	Insérer une photo	63
5	Tableaux	65
5.1	Insérer un tableau	65
5.2	Cellules spéciales	69
5.3	Descripteurs de colonne avancés	72
5.4	Apparence	75
5.5	Dimensions et positionnement	82
5.6	Mise en page avancée	87
6	Boite	91
6.1	Insérer une boite	91
6.2	Boites de transformation	95
6.3	Styles de boites	98
6.4	Grande boite	101
7	Mathématique	103
7.1	Mode mathématique	103
7.2	Symboles	104
7.3	Constructions basiques	107
7.4	Délimiteurs	111
7.5	Tableaux, matrices et équations	112
7.6	Théorèmes	114
8	Autour du texte	117
8.1	Références croisées	117
8.2	Légendes	122
8.3	Entête et pied de page	123
8.4	Notes de bas de page	127
8.5	Notes de marge	130
8.6	Modification	132
9	Codes source	133
9.1	Insertion brute	133

9.2	Codes source d'un programme	136
9.3	Algorithmes	140
10	Bibliographies	145
10.1	Insérer une bibliographie	145
10.2	Utiliser <code>BIBTEX</code>	146
11	Dessin	153
11.1	Arbres	153
11.2	TikZ/PGF	155
11.3	PSTricks	158
12	Personnaliser <code>LATEX</code>	161
12.1	Commandes et environnements	161
12.2	Tests et répétitions	162
12.3	Définir un flottant	165
12.4	Définir un compteur	167
12.5	Définir une longueur	167
13	Divers	169
13.1	Mots-croisés	169
13.2	Code-barres	171
13.3	Filigrane	172
13.4	Nombre aléatoire	172
13.5	Alphabet phonétique international	173
14	Classes de documents	175
14.1	Écrire une lettre	176
14.2	Présenter un CV	178
14.3	Mettre en page un journal	180
14.4	Construire une présentation avec des slides	184
A	Utiliser <code>LATEX</code> sur mon ordinateur	189
B	Quelques liens utiles	191
	Liste des packages	193

viii

AVANT-PROPOS

Index

195

Avant-propos

\LaTeX est certes un système formidable, mais il est loin d'avoir été bien pensé dès le départ. Tous les environnements et commandes s'utilisent de manière hétéroclite et il existe une pléthore de packages développés par différentes personnes, dont plusieurs permettent de réaliser la même chose . . . ou presque.

Afin de maîtriser toutes ces possibilités, il n'y a pas d'autre manière que de pratiquer un maximum. Étant un très grand fan de \LaTeX , et avec maintenant bientôt 8 ans d'expérience de rédaction avec ce système, j'ai eu l'occasion de me frotter à un si grand nombre de packages que mes amis, collègues, étudiants, etc. viennent souvent me trouver quand ils veulent réaliser quelque chose en \LaTeX qu'ils ne savent pas faire.

Voilà donc ce qui a donné lieu à la rédaction de ce livre dont le titre est « \LaTeX HowTo » qu'on pourrait traduire par « \LaTeX comment faire ? ». Ce livre tente d'apporter des réponses à cette question, sur base de toute l'expérience que j'ai acquise dans ce domaine.

Bien entendu, ce livre ne couvre pas tout. Il y aurait encore tant de choses à dire ! Néanmoins, ce livre est avant tout pour vous. Si vous n'y trouvez pas quelque chose qui devrait, selon vous, s'y trouver ou si vous avez quelque remarque que ce soit, n'hésitez pas à vous connecter sur le site web officiel du livre pour en discuter :

<http://www.latex-howto.be>

J'y serai à votre écoute et ferai tout mon possible pour répondre à vos demandes et cela profitera par la même occasion à toute la communauté des lecteurs de ce livre.

Utiliser ce livre

J'aimerais avant tout prendre trente secondes pour vous expliquer ce que ce livre n'est pas. Il ne s'agit pas ici d'un tutoriel sur L^AT_EX et vous n'allez donc pas pouvoir apprendre, à partir de rien, comment fonctionne L^AT_EX et comment rédiger des documents. Il ne s'agit pas non plus d'un livre qui va vous expliquer le fonctionnement interne de L^AT_EX et les algorithmes de mise en page.

Comme annoncé précédemment, il faut plutôt voir ce livre comme un guide ou comme un pense-bête. Il tente de répondre à la question « How to ... with L^AT_EX ? » ou « Comment faire ... avec L^AT_EX ? ». Le livre est organisé en chapitres, chacun de ceux-ci couvrant un thème (images, tableaux, mathématiques ...) et répondant aux questions s'y rattachant. Afin de trouver une question à votre réponse, vous pouvez soit profiter de la table des matières, soit utiliser l'index. Si vous cherchez des informations sur un package, vous pouvez également vous baser sur la liste des packages présentés dans ce livre se trouvant à la fin de ce dernier.

Remerciements

Pour conclure, et avant de vous laisser vous plonger dans le livre, je tiens à remercier particulièrement trois personnes sans qui ce livre n'aurait jamais vu le jour. Tout d'abord Pierre Bouilliez qui m'a donné l'idée d'un tel livre et qui a fait une relecture complète de la première version. Ensuite, Marie Lateur pour avoir relu intégralement la seconde version du livre et avoir décelé des fautes d'orthographe. Enfin, un tout grand merci à Pierre-Louis Peeters pour la relecture finale et pour ses remarques tant au niveau orthographique et grammatical qu'au niveau du contenu.

Je remercie également toutes les autres personnes qui ont suscité un intérêt grandissant pour un tel livre, du début à la fin. Ceux-ci m'ont permis de ne pas abandonner ce projet en cours de route et de le concrétiser afin qu'aujourd'hui, vous puissiez tenir ce livre entre vos mains.

Louvain-la-Neuve, le 3 décembre 2010.

Sébastien Combéfis
sebastien@combefis.be

Introduction

Ce livre s'adresse principalement aux personnes ayant déjà une connaissance élémentaire de \LaTeX . Cette introduction va discuter, en quelques pages, de ce qu'est le système \LaTeX , de sa philosophie, et de comment l'utiliser afin de rédiger un document minimal.

Vous pouvez voir cette introduction comme un rapide tutoriel sur \LaTeX . Pour en savoir plus sur comment utiliser \LaTeX sur votre ordinateur en pratique, vous pouvez consulter l'annexe A et le site web <http://www.latex-howto.be>. Vous trouverez également toute une série d'informations complémentaires et des liens vers des tutoriels complets sur ce site web.

Bref historique

\LaTeX a été développé par Leslie Lamport en 1984 sur base du système \TeX de Donald Knuth, créé en 1977. Il s'agit d'un ensemble de programmes libres permettant de composer des documents de qualité professionnelle.

La version actuelle, $\text{\LaTeX} 2_{\epsilon}$, est le résultat du travail de toute une communauté de développeurs qui demeure actuellement très active. Pendant ce temps, la communauté travaille également sur la prochaine version de \LaTeX depuis le début des années 90. Cette version est toujours en cours de développement aujourd'hui. Vous trouverez plus d'informations à ce propos sur le site officiel du projet \LaTeX :

<http://www.latex-project.org/latex3.html>

Pour ou contre ?

Pourquoi faudrait-il utiliser \LaTeX ? Tout d'abord parce qu'il s'agit d'un système libre qui existe pour Windows, Linux et Mac OS X. Ce dernier permet également d'obtenir des documents de qualité professionnelle.

Certes, l'apprentissage de ce langage n'est pas aisé au départ, et vous pourriez très certainement préférer des systèmes classiques dits WYSIWYG¹ qui vous permettent d'éditer un document tout en voyant directement le résultat final. Cela dit, il existe un éditeur WYSIWYG pour \LaTeX : LyX². De plus, les éditeurs \LaTeX courants proposent de nombreux outils d'aide à la rédaction et, enfin, une fois \LaTeX appris, vous vous rendrez compte que ça va beaucoup plus vite de rédiger un document en \LaTeX , même sans pouvoir voir directement le document produit.

On vous dira également qu'il n'y a pas de correcteurs orthographiques pour \LaTeX . Cela est bien entendu faux étant donné que la plupart des éditeurs en intègrent un.

De toute façon, il n'y a pas trente-six-mille questions à se poser. Essayez-le et adoptez-le.

Philosophie

Quelle est la philosophie principale de \LaTeX ? Il s'agit essentiellement de séparer le contenu du document de sa présentation. Lorsque vous rédigez votre document, vous devez vous concentrer sur le texte et sur la structuration du document. Ensuite, vous définissez le type de document dont il s'agit afin de définir le style à appliquer.

Bases de \LaTeX

En \LaTeX , il y a essentiellement deux types de constructions : les *commandes* et les *environnements*. On va rapidement les passer en revue ici avec un ou deux exemples à chaque fois.

1. What You See Is What You Get.
2. <http://www.lyx.org/>

Commandes

Les commandes prennent la forme suivante :

$$\backslash\text{nom}[\text{option}]\{\text{paramètre}_1\}\{\text{paramètre}_2\}\dots\{\text{paramètre}_n\}$$

On commence donc avec un backslash, suivi du nom de la commande. Viennent ensuite éventuellement des options entre crochets et enfin une suite de paramètres entre accolades. Voici quelques exemples de commandes avec le résultat produit :

<code>\LaTeX</code>	ℒ _A T _E X
<code>\textcircled{c}</code>	©
<code>\$\$\sqrt[n]{x}\$\$</code>	$\sqrt[n]{x}$

Remarquez que la troisième commande a été placée entre dollars. Cela permet de passer en mode mathématique, ce qui est nécessaire puisque la commande `\sqrt` n'est disponible que dans ce mode, comme on le verra au chapitre 7.

Environnements

Il y a également des environnements, dont la forme générale est la suivante :

$$\backslash\text{begin}\{\text{nom}\}[\text{options}] \dots \text{contenu} \dots \backslash\text{end}\{\text{nom}\}$$

Par exemple, pour faire en sorte d'avoir un texte qui est aligné à droite, et non justifié comme le fait par défaut ℒ_AT_EX, on peut utiliser l'environnement `flushright`.

Texte normal

```
\begin{flushright}
Texte à droite
\end{flushright}
```

Une citation de Donald Knuth :

```
\begin{quote}
“God is a challenge because
there is no proof of his
existence and therefore the
search must continue.”
\end{quote}
```

Texte normal


Texte à droite

Une citation de Donald Knuth :

“God is a challenge because there is no proof of his existence and therefore the search must continue.”

Document minimal

Voici un document minimal pratique à utiliser pour la rédaction d'un document en français, avec l'encodage `latin-1`. Vous pourriez également utiliser l'encodage `utf8`, en fonction de la configuration de votre éditeur.

**Code**

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

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}
\usepackage[french]{babel}

\title{Titre de mon article}
\author{Moi \and Mon Co-auteur}
\date{\today}

\begin{document}
  \maketitle

  Il était une fois \dots
\end{document}
```

CHAPITRE 1

Structurer le document


Dans ce chapitre, on va voir tout ce qui permet de structurer un document. On commencera par présenter les différents types de documents qu'il est possible de définir. Ensuite, on verra comment réaliser une page de titre. On enchainera avec les différentes commandes de sectionnement du document. Enfin, on terminera en voyant comment insérer une table des matières et une liste de figures ou de tables.

1.1 Type de document

On définit le type d'un document avec la commande `\documentclass`. Celle-ci prend en paramètre le type de document que vous voulez. Parmi les valeurs courantes possibles, on retrouve :

- `article` pour un article;
- `report` pour un rapport;
- `book` pour un livre;
- `letter` pour une lettre;
- `beamer` pour des transparents.

On définit ensuite le document avec l'environnement `document`. Voici un exemple de document L^AT_EX minimal :



Code

```
\documentclass{report}
\begin{document}
  Hello World !
\end{document}
```

1.2 Page de titre

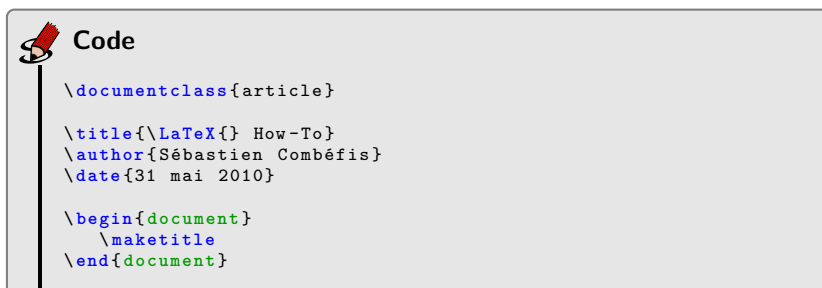
On peut obtenir une page de titre pour un document avec la commande `\maketitle`. On peut choisir si on souhaite que cette page soit sur une page séparée ou non en utilisant respectivement les options `titlepage` et `notitlepage` de la commande `\documentclass`.

LaTeX How-To

Sébastien Combéfis

31 mai 2010

Pour pouvoir définir le titre, il faut avant tout définir les données qu'il va contenir grâce aux commandes `\title`, `\author` et `\date`. Lorsqu'il y a plusieurs auteurs, on les sépare avec la commande `\and`. On peut obtenir la date courante avec la commande `\today`. On fait habituellement ces définitions dans le préambule.



Code

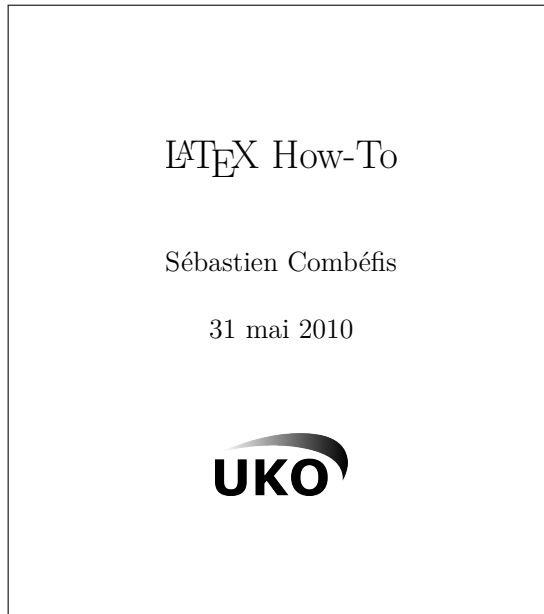
```
\documentclass{article}

\title{\LaTeX{} How-To}
\author{Sébastien Combéfis}
\date{31 mai 2010}

\begin{document}
  \maketitle
\end{document}
```

1.2.1 Insérer une image

Pour insérer une image sur la page de titre, il faut utiliser le package `titlepic` qui définit la commande `\titlepic` qui permet d'ajouter l'image. On peut spécifier une option lors du chargement du package qui permet de spécifier l'alignement du titre. Trois valeurs sont possibles : `tt`, `tc` et `cc`. La première lettre désigne l'alignement du bloc titre, auteur, date et la seconde lettre celui de l'image. La lettre `t` aligne en haut et la lettre `c` permet de centrer.



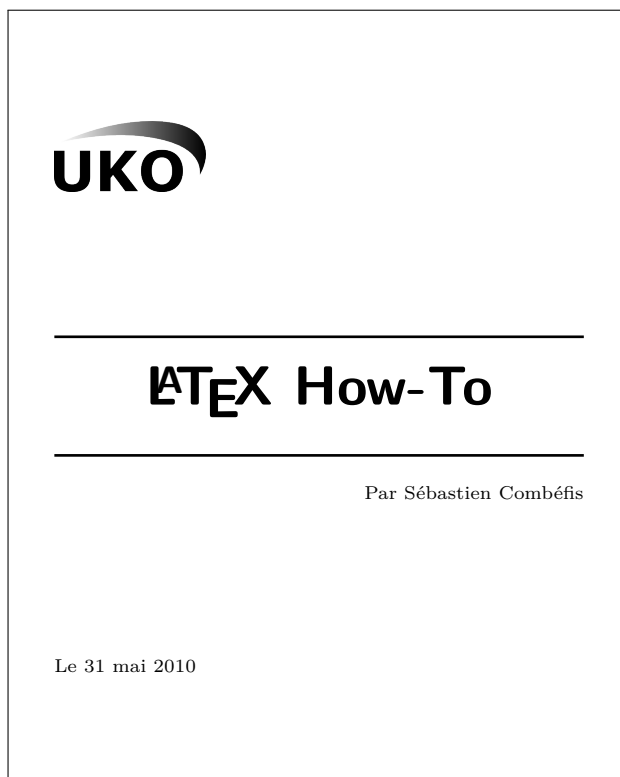
Dans cet exemple, on a importé `\usepackage[tt]{titlepic}`, ce qui permet de centrer tous les éléments de la page de titre.

```
Code graphicx, titlepic  
\title{\LaTeX{} How-To}  
\author{Sébastien Combéfis}  
\date{31 mai 2010}  
\titlepic{\includegraphics[width=2cm]{../images/uko-logo.png}}  
  
\maketitle
```

1.2.2 Page de titre personnalisée

On souhaite parfois définir un style de page personnalisé pour sa page de titre. Il n'y a pas de package qui permet de faire cela, et on va dès lors contourner le problème. Ce qu'on va faire, c'est utiliser `\title` pour définir complètement sa page, et laisser `\author` et `\date` vides.

La page suivante montre une page de titre plus sophistiquée dans laquelle on inclut notamment un logo. Une fois ce principe compris, pour le reste, c'est à vous de jouer en faisant appel à votre créativité.



Comme vous pouvez le voir, on ne peut pas insérer de lignes vides dans la commande `\title`. On doit donc utiliser `\\`, en précisant éventuellement une distance, pour ajouter des espaces verticaux.

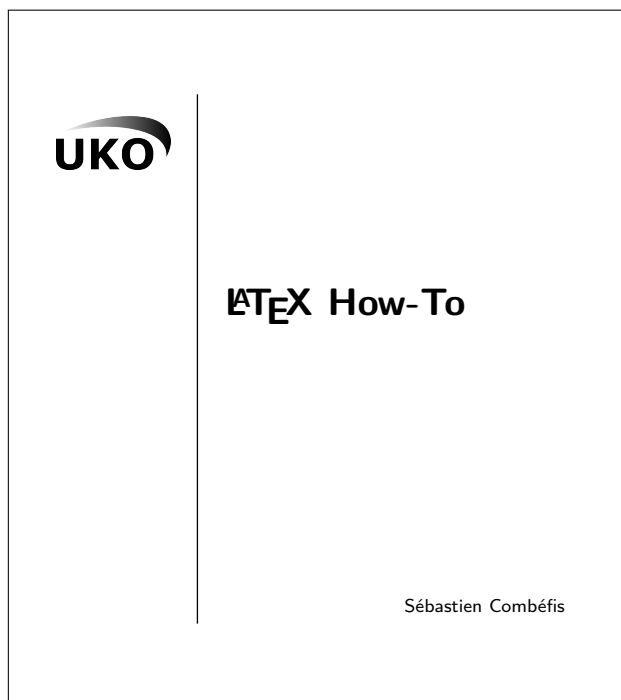
Code
graphicx

```

\title{\parbox{7cm}{
  \includegraphics[width=2cm]{uko-logo.png} \\
  \begin{center}\sf\bfseries\huge
    \rule{7cm}{1pt}
    \medskip
    \LaTeX{} How-To \ll[-4mm]
    \rule{7cm}{1pt}
  \end{center}
  \hfill\scriptsize Par Sébastien Combéfis \ll[2cm]
  Le 31 mai 2010
}}
\author{}
\date{}

```

Une autre solution consiste à utiliser l'environnement `titlepage`, qui permet de définir une page de titre. Son contenu sera placé sur une page non numérotée et le compteur de page sera remis à zéro.



Comme dit précédemment, maintenant, c'est à vous de jouer pour trouver votre propre style. Vous pourrez trouver divers exemples sur Internet, au cas où l'inspiration vous manquerait.

Code
graphicx

```

\begin{titlepage}
  \vspace{1cm}
  \includegraphics[width=1.5cm]{uko-logo.png}%
  \quad \rule[-6cm]{0.5pt}{7cm}%
  \quad \parbox[t]{4.5cm}{\sf
    \ \ [1.5cm]
    {\Large\bfseries \LaTeX{} How-To} \ \ [3.5cm]
    \hspace*{\fill}{\scriptsize Sébastien Combéfis}
  }
\end{titlepage>

```

1.3 Sectionnement

Il existe différentes commandes qui permettent de délimiter les parties logiques d'un document. Celles-ci dépendent de la classe de document utilisée. On peut classer ces commandes en deux catégories : les commandes qui délimitent une partie et celles qui définissent en plus un titre.

On définit les titres avec les commandes `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` et `\subparagraph`. Celles-ci permettent de définir respectivement une partie, un chapitre, différents niveaux de sections et des paragraphes. Elles prennent toutes un paramètre qui est le texte du titre et une option qui est une version courte du titre. Cette dernière sera notamment utilisée dans la table des matières. Enfin, toutes ces commandes existent en version étoilée. Dans ce cas, le titre ne sera pas numéroté et n'apparaîtra pas dans la table des matières.


1 Les abeilles

Introduction

1.1 Rôle dans la pollinisation

1.2 Déclin de la population

Cet exemple a été généré avec la classe de document `article` dans laquelle le plus haut niveau est la `\section`. Pour les livres et articles, il faut commencer avec un `\chapter`.



Code

```

\section{Les abeilles}
\subsection*{Introduction}
\subsection{Rôle dans la pollinisation}
\subsection{Déclin de la population}

```

1.3.1 Style des titres

On va pouvoir modifier le style des titres en utilisant le package `titlesec`, comme on l'a fait pour ce livre. On va pour cela utiliser les commandes `\titlelabel`, `\titleformat` et `\titlespacing` qui permettent respectivement de définir la numérotation, le format et style du titre, ainsi que l'espacement à appliquer.

La commande `\titlelabel` permet de définir comment rendre les numéros des sections, sous-sections ... La commande `\thetitle` représente le numéro. Le style par défaut est :

```
\titlelabel{\thetitle\quad}
```

Par exemple, on pourrait ajouter une parenthèse fermante après les numéros avec :

```
\titlelabel{\thetitle)\quad}
```

1) Les abeilles

Introduction


1.1) Rôle dans la pollinisation

1.2) Déclin de la population

Le style des titres est modifié avec la commande `\titleformat`. Le plus simple est d'utiliser la forme étoilée qui prend deux paramètres : le niveau de titre à modifier et le style à appliquer. Les sections de ce livre sont obtenues avec :

```
\titleformat*{\section}{\large\sf\bfseries}
```

Il existe également une version non-étoilée qui permet plus de paramétrage. Le listing suivant donne la commande qu'on a utilisée dans ce livre pour les chapitres.


Code
titlesec

```
\titleformat{\chapter}[display]{\sf\bfseries\LARGE}
{\vspace{-10ex}
\filleft\MakeUppercase{\chaptertitlename}-\Huge\thechapter}
{4ex}{\titlerule\vspace{2ex}\filright}{\vspace{2ex}\titlerule}
```

La forme générale de la commande `\titleformat` est :

```
\titleformat{command}[shape]{format}{label}{sep}{before}[after]
```

- `command` représente la commande de titre à modifier ;
- `shape` permet de repartir d'un modèle de base (`hang` pour un titre comme un paragraphe séparé, `runin` pour un titre qui reste dans le paragraphe, `frame` pour un titre encadré ...);
- `format` correspond au second paramètre de `\titleformat*`;
- `label` correspond à la commande `\titlelabel` ;

- `sep` correspond à l'espace horizontal à ajouter entre la numérotation et le texte titre ;
- `before` correspond au code à insérer avant le titre ;
- `after` correspond au code à insérer après le titre.

Pour redéfinir un chapitre, il faut utiliser la commande `\chaptertitlename` au lieu de `\chaptername` pour insérer le texte du titre du chapitre. Ceci permet d'avoir une même définition qui sera également appliquée aux chapitres situés dans les annexes.


Enfin, on peut modifier les espaces du titre en utilisant la commande `\titlespacing`. Celle-ci nécessite quatre paramètres et une option :

```
\titlespacing{command}{left}{beforesep}{aftersep}[right]
```

- `command` représente la commande de titre à modifier ;
- `left` indique l'espace à ajouter à gauche du titre ;
- `beforesep` indique l'espace vertical ajouté avant le titre ;
- `aftersep` indique l'espace ajouté entre le titre et le texte qui suit (vertical ou horizontal selon le modèle du titre) ;
- `right` indique l'espace à ajouter à droite du titre.

Tous ces paramètres n'ont pas la même signification selon le modèle de titre utilisé. Consultez la documentation du package pour en savoir plus.

Il existe également le package `sectsty` pour modifier le style des titres. Avec ce dernier, on ne peut modifier que le style. On ne va pas détailler les commandes de ce package ici, puisqu'on peut faire l'équivalent avec le package `titlesec` qu'on vient de voir. Tout se base sur la commande `\allsectionsfont` qui permet de définir le style de tous les titres, et `\sectionfont` pour les sections, `\chapterfont` pour les chapitres ...

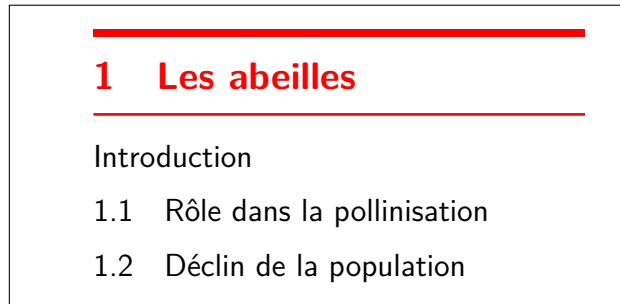


Code


```
\allsectionsfont{\sf}
\chapterfont{\raggedleft}
```

sectsty

On peut obtenir d'autres styles avec ce package, notamment en insérant des traits horizontaux au-dessus et en-dessous du texte du titre avec la commande `\sectionrule`.



La commande `\sectionrule` prend quatre paramètres qui sont l'espacement en-dessous du premier trait et son épaisseur, et l'espacement au-dessus du second trait et son épaisseur.



Code
color, sectsty

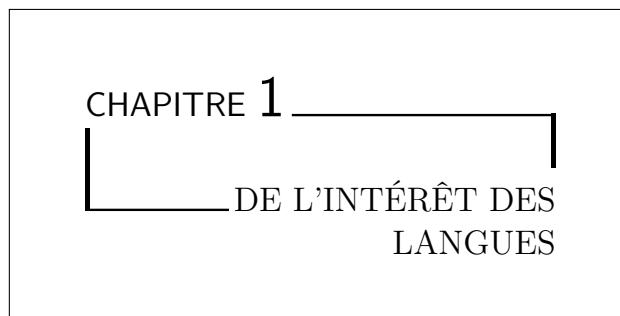
```

\allsectionsfont{\sf}
\sectionfont{\sffamily\color{red}\sectionrule{3ex}{3pt}{%
-1.5ex}{1pt}}


```

1.3.2 Style des chapitres

Lorsqu'on désire modifier le style des titres des chapitres, on peut également utiliser le package `fncychap` qui offre des possibilités de personnalisation riches pour ces derniers.



L'exemple présenté ci-dessus a été obtenu en chargeant le package avec l'option `Glenn`. Il existe six styles différents qui sont `Sonny`, `Lenny`, `Glenn`, `Conny`, `Rejne` et `Bjarne`.

 **Code** fncychap


```

\documentclass{report}
\usepackage[Glenn]{fncychap}
\begin{document}
  \chapter{De l'intérêt des langues}
\end{document}

```

1.3.3 Numérotation des titres dans la marge

Comme vous avez pu le constater dans ce livre, les numéros des sections sont placés dans la marge. Pour ce faire, il suffit d'ajouter la définition suivante dans le préambule.

 **Code**

```

\makeatletter
\def\@secntformat#1{%
  \protect\makebox[0pt][r]{\csname the#1\endcsname\quad}}
\makeatother

```

1.3.4 Parties de livre

On peut délimiter les différentes parties d'un livre avec les commandes `\frontmatter`, `\mainmatter`, `\backmatter` et `\appendix`. Ces commandes vont notamment avoir pour effet de changer le style de numérotation des pages.

La commande `\mainmatter` indique le début du corps du livre. La commande `\frontmatter` délimite tout ce qui vient avant (préface, introduction ...) et la commande `\backmatter` ce qui vient après (colophon, index ...).

La commande `\appendix` indique le début des annexes. La numérotation des sections va se faire avec des lettres romaines plutôt qu'avec des nombres arabes.

1.4 Table des matières et liste des figures et tables

Dans cette section, on va s'intéresser à l'insertion de listes dont le contenu est essentiellement généré automatiquement. On va voir comment insérer une table des matières, mais également les tables des figures et des tableaux. On verra également comment modifier leurs styles et leurs contenus.

1.4.1 Insérer une table

On insère les différentes tables avec les commandes `\tableofcontents`, `\listoffigures` et `\listoftables`. Notez que pour que les tables soient à jour dans votre document, il va falloir deux compilations successives. La première permet de récolter l'information à partir des commandes `\chapter`, `\section` ... et `\caption` et de les placer dans les fichiers `.toc`, `.lof` et `.lot` ; la seconde compilation intègre ces données dans le document.


1.4.2 Ajouter une entrée dans une table

Les entrées des différentes tables sont automatiquement ajoutées. Parfois, on souhaite néanmoins les ajouter manuellement. C'est par exemple le cas si vous utilisez les versions étoilées des commandes `\chapter`, `\section`, `\subsection` ... qui ne reçoivent pas de numéro et n'apparaissent pas dans la table des matières.

Pour ajouter une entrée dans une table, il faut utiliser la commande :

```
\addcontentsline{Extension}{Division}{Texte}
```

à l'endroit du texte où doit figurer cette nouvelle entrée. Par exemple, voici comment ajouter une entrée pour la table des matières :



Code

```
\addcontentsline{toc}{chapter}{Table des matières}
\tableofcontents
```

Le premier paramètre indique la table dans laquelle l'entrée doit être ajoutée (`toc` pour *table of contents*). Le second paramètre indique le niveau de l'entrée (`chapter` pour le niveau chapitre) et enfin, le troisième paramètre indique le texte qui doit figurer dans la table.

1.4.3 Style des tables

On peut modifier l'aspect des différentes tables, comme par exemple modifier l'espacement entre les différentes lignes. Pour ce faire, il suffit d'utiliser les commandes \LaTeX standard. Par exemple, pour insérer une table des figures, avec un millimètre entre les lignes de la table, il suffit de faire :

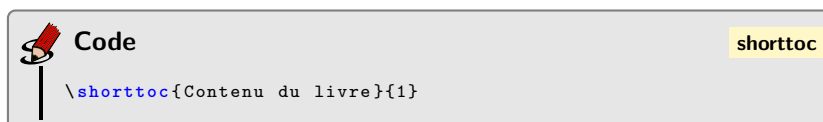
```
\setlength{\parskip}{1mm} \listoffigures}
```

Pour la table des matières, on peut spécifier son niveau, c'est-à-dire jusqu'à quel type de sous-entrée il faut aller. Cette valeur est contrôlée par le compteur `tocdepth`. Par exemple, pour ne lister que les chapitres, il faut écrire avant `\tableofcontents` :

```
\setcounter{tocdepth}{1}
```

1.4.4 Insérer plusieurs tables

On ne peut utiliser la commande `\tableofcontents` qu'une seule fois dans un document. Or, il n'est pas rare qu'on souhaite insérer plusieurs fois la table des matières, parfois même avec un niveau différent. La solution à ce problème consiste à utiliser la commande `\shorttableofcontents` ou `\shorttoc` du package `shorttoc`. La commande prend deux paramètres qui sont le nom de la table et son niveau.

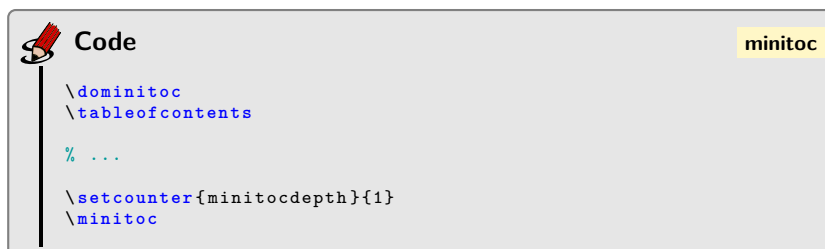


Attention, pour pouvoir utiliser cette commande, il faut absolument que le fichier `.toc` soit disponible, c'est-à-dire que la commande `\tableofcontents` doit être présente. De plus, la commande `\shorttoc` doit être placée avant `\tableofcontents`.

Il existe une autre solution qui permet d'insérer une mini-table des matières, par partie, chapitre ou section. Pour ce faire, on va utiliser le package `minitoc`. Voici par exemple la table des matières de ce chapitre :

Sommaire		
1.1	Type de document	6
1.2	Page de titre	4
1.3	Sectionnement	9
1.4	Table des matières et liste des figures et tables .	14

Pour l'obtenir, il faut avant tout appeler la commande `\dominitoc` juste avant de faire appel à `\tableofcontents`. Ensuite, on utilise la commande `\minitoc` pour insérer la mini-table des matières.



```
Code minitoc  
\dominitoc  
\tableofcontents  
  
% ...  
  
\setcounter{minitocdepth}{1}  
\minitoc
```

Afin d'avoir le titre en français, il va falloir spécifier la langue lorsque vous incluez le package et donc écrire `\usepackage[french]{minitoc}`. Notez qu'avec ce package, vous pouvez également insérer des mini-listes des figures ou tables avec les commandes `\dominilof`, `\minilof`, `\dominilot` et `\minilot`.

CHAPITRE 2

Style et mise en page

Dans ce chapitre, nous allons voir comment mettre en page le texte et modifier le style de ce dernier. On commencera par voir comment modifier l’alignement du texte et insérer des citations ou des poèmes. On verra ensuite comment modifier le style de la police et utiliser les couleurs. Enfin, on terminera avec des mises en page avancées du texte.

2.1 Alignement et positionnement

Dans cette section, on va voir trois choses. On va commencer avec l’alignement du texte. Ensuite, on verra comment gérer les espaces et en insérer dans le document. Enfin, on terminera en voyant comment remplir de l’espace avec des traits (continus, pointillés ...).

2.1.1 Alignement


Par défaut, les paragraphes d’un document sont justifiés, c’est-à-dire que le texte est accolé aux marges gauches et droites. Cet effet est obtenu en rendant la largeur des espaces entre les mots variable.

Aligné à gauche

Centré

Aligné à droite

On peut modifier ce comportement et forcer le texte à être collé à gauche ou à droite, ou centrer le texte en utilisant respectivement les commandes `\raggedright`, `\raggedleft` et `\centering`.



Code

```

\raggedright Aligné à gauche
\centering Centré
\raggedleft Aligné à droite

```


Il faut bien faire attention que `\raggedright` permet de coller le texte à gauche.

On peut également modifier l'alignement du texte avec les environnements `flushleft`, `flushright` et `center`. La différence est qu'un espace vertical sera ajouté avant le texte de l'environnement étant donné que ces environnements commencent un nouveau paragraphe.

Aligné à gauche

Centré

Aligné à droite



Code

```

\begin{flushleft}
  Aligné à gauche
\end{flushleft}
\begin{center}
  Centré
\end{center}
\begin{flushright}
  Aligné à droite
\end{flushright}


```

2.1.2 Espacement

Il y a différents types d'espacements possibles. Tout d'abord, on a le choix entre des espacements horizontaux ou verticaux. Ensuite, on peut choisir entre des espaces de longueur fixe ou élastique.

On obtient des espaces horizontaux de longueur fixe avec `\hspace` et des verticaux avec `\vspace`.

Texte précédé de 1.5cm.
Paragraphe précédé de -1mm.



Code

```

Une ligne de texte \\
\hspace{1cm} Pas de décalage \\
\hspace*{1cm} Un décalage de 1 centimètre \\
\hspace*{\fill} Texte poussé à droite


```

La commande `\\` permet d'insérer un retour à la ligne. On peut utiliser l'option de cette commande pour spécifier un certain espacement vertical.

Voici un texte :

HELLO

Dans l'exemple ci-dessus, on a placé `Hello` suivi d'un retour à la ligne suivi de `/////`. Afin d'avoir les deux textes superposés, on a fait remonter la seconde ligne de `\baselineskip` (la hauteur d'une ligne de texte).



Code

```

\it Voici un texte : \\[2mm]
\tt
Hello \\[-\baselineskip]
/////


```

2.1.3 Remplissage

On peut remplir des espaces horizontaux de différentes manières. Les commandes `\hrulefill` et `\dotfill` insèrent un espace horizontal élastique rempli avec un trait continu ou des pointillés.

A _____ B C

On peut utiliser les différentes commandes sur la même ligne, et elles se partageront l'espace disponible équitablement.



Code

```


A \hrulefill B \dotfill C

```

Une autre solution consiste à utiliser la commande `\dashrulefill` du package `nccrules`.

(—— ———)

A B =:~::~: C



Code nccrules

```
(\dashrule[1mm]{1cm 5mm 1cm}{1}) \\[3mm]
A \dashrulefill{2 4}{1} B \dashrulefill{4 2 1 2}{1 4 1} C
```

La commande `\dashrulefill` prend deux paramètres, chacun de ceux-ci étant une liste de nombres entiers. Le premier nombre représente la longueur du premier trait plein, le second la longueur du premier trait blanc, etc. Le motif est ensuite répété. Le premier paramètre est utilisé pour le motif horizontal et le second pour le motif vertical.

Avec `\dashrulefill{2 4}{1}`, on définit donc pour le motif horizontal un trait plein de largeur 2 suivi d'un blanc de largeur 4. On n'a qu'un seul trait selon l'axe vertical.

Avec `\dashrulefill{4 2 1 2}{1 4 1}`, on définit que le motif horizontal est formé d'un trait plein de largeur 4 suivi d'un blanc de largeur 2, d'un trait plein de largeur 1 et d'un blanc de largeur 2. Verticalement, on a une ligne de hauteur 1, un blanc de hauteur 4 et à nouveau une ligne de hauteur 1.

Contrairement à `\dashrulefill`, la commande `\dashrule` ne répète pas le motif. Les deux commandes prennent également une option qui indique une longueur de laquelle le trait sera remonté verticalement.

2.2 Césure

L^AT_EX est capable d'automatiquement découper les mots lorsque ceux-ci sont trop longs pour tenir sur la ligne courante et si le fait de faire passer tout le mot sur la ligne suivante est inesthétique. Pour cela, il se base sur un dictionnaire de césure lui permettant de savoir où découper chaque mot.

Néanmoins, il se peut qu'il ne sache pas où découper certains mots et afin de lui indiquer, vous pouvez utiliser la commande `\-` qui indique les points de découpe possibles. Par exemple, on pourrait écrire :

```
Ceci est an\-\ti\-\cons\-\ti\-\tu\-\tion\-\nel
```

Si vous avez un mot qui revient souvent dans vos documents, il n'est pas pratique de devoir indiquer à chaque fois comment le découper. Pour cela, il vous suffit d'utiliser la commande `\hyphenation` dans le préambule, une fois pour toute.

```
\hyphenation{an-ti-cons-ti-tu-tion-nel}
```

2.3 Style

Dans cette section, on va voir les commandes de base pour gérer le style global du document ainsi que celui du texte en particulier.

2.3.1 Style du document

Le style global du document est modifié avec la commande `\pagestyle`. On peut également modifier le style d'une seule page avec `\thispagestyle`. Les trois valeurs de base possibles sont :

- `empty` laisse l'entête et le pied de page vides ;
- `plain` ajoute le numéro de page au milieu du pied de page ;
- `headings` insère le titre du chapitre courant et le numéro de page dans l'entête.

Le style par défaut est `plain`. Notez également que le style de la première page de chaque chapitre est différent du style des autres pages.

2.3.2 Style de la numérotation

On peut modifier le style de la numérotation de deux manières. Celui-ci sera altéré en fonction de la partie du document, définie avec les commandes vues à la section 1.3.4.

L'autre solution consiste à utiliser la commande `\pagenumbering` qui permet de changer le style de la numérotation à partir de la page courante. Le paramètre de ces commandes peut prendre l'une des valeurs suivantes :

- `arabic` nombres arabes
- `roman` ou `Roman` nombres romains minuscules ou majuscules
- `alph` ou `Alph` lettres de l'alphabet minuscules ou majuscules


2.3.3 Style de la police

On peut modifier trois éléments de la police du texte : la famille, la forme et la graisse. Il y a trois familles, quatre formes et deux graisses.

Romain, Sans-serif, Machine à écrire
 Droit, *Italique*, *Incliné*, PETITE MAJUSCULE
 Moyen, **Gras**

Il y a plusieurs manières d'obtenir ces styles. Il y a tout d'abord les commandes déclaratives qui modifient le style à partir de l'insertion de

la commande, jusqu'à la fin du bloc courant (délimité par l'accolade fermante).



Code

```
{\rmfamily Romain,
\sffamily Sans-serif,
\ttfamily Machine à écrire} \

{\upshape Droit,
\itshape Italique,
\slshape Incliné,
\scshape Petite majuscule} \

{\mdseries Moyen,
\bfseries Gras}
```

Ces commandes existent également en version courte : `\rm`, `\sf`, `\tt`, `\it`, `\sl`, `\sc` et `\bf`. La commande déclarative `\normalfont` permet de remettre le style par défaut (romain, droit, grasse moyenne).

On peut également utiliser les commandes qui vont modifier le style de leur paramètre. Il s'agit des commandes `\textsf`, `\texttt`, `\textit`, `\textsl`, `\textsc`, `\textbf` et `\textnormal`. On va par exemple pouvoir écrire :

```
\textit{Texte en italique} et texte normal
```


à comparer avec la version déclarative :

```
{\it Texte en italique} et texte normal
```

Enfin, il est également possible d'utiliser les environnements `sf`, `sffamily`, `tt`, `ttfamily`, `it`, `itshape`, `sl`, `slshape`, `sc`, `scshape`, `bf` et `bfseries`.

Texte en gras sans-serif. **Ceci est en gras**, en normal, *et en italique*.

Comme vous avez pu le voir sur l'exemple, utiliser les environnements ne crée pas un nouveau paragraphe. De plus, vous voyez qu'il est possible de combiner plusieurs effets en sélectionnant un style dans chaque catégorie. Cependant, certaines combinaisons n'existent pas, comme par exemple petites majuscules grasses.

 **Code**

```
{\bfseries\sf Texte en gras sans-serif.}
\begin{bf}
  Ceci est en gras, \rm en normal, \textit{et en italique}.
\end{bf}
```


Enfin, on peut également emphaser un texte en utilisant la commande `\emph`. Par défaut, un texte emphasé est rendu en italique.

2.3.4 Texte souligné

Pour souligner un texte, il suffit d'utiliser la commande `\underline`. Néanmoins, il n'est pas possible de souligner un texte qui dépasse une ligne avec cette commande.

- Voici un très long texte qui est tellement long qu'il ne tient pas sur une seule ligne de texte.
- Voici un très long texte qui est tellement long qu'il ne tient pas sur une seule ligne de texte.


Pour pouvoir souligner un texte qui s'étend sur plusieurs lignes, il faut utiliser la commande `\uline` du package `ulem`.

 **Code** **ulem**

```
\underline{- Voici un très ... une seule ligne de texte.}
\uline{- Voici un très ... une seule ligne de texte.}
```

L'insertion du package `ulem` va modifier le comportement de la commande `\emph`. Les textes emphasés seront soulignés. Pour remettre en place le comportement par défaut, il suffit d'utiliser la commande `\normalem`.

Emphase *Emphase*

 **Code** **ulem**

```
\emph{Emphase} \normalem \emph{Emphase}
```


D'autres commandes de soulignement sont proposées dans le package `ulem`. On peut barrer du texte, souligner deux fois, en vague ...

<u>Texte souligné</u>	<code>\uline{Texte souligné}</code>
<u><u>Double souligné</u></u>	<code>\uuline{Double souligné}</code>
Texte barré	<code>\sout{Texte barré}</code>
Texte rayé	<code>\xout{Texte rayé}</code>
<u>~Souligné en vagues</u>	<code>\uwave{Souligné en vagues}</code>
<u>-Souligné en tirets</u>	<code>\dashuline{Souligné en tirets}</code>
<u>.Souligné en points</u>	<code>\dotuline{Souligné en points}</code>

Les deux dernières commandes sont disponibles depuis la toute dernière version du package (21/01/2010). On peut également définir ses propres types de soulignements avec `\markoverwith` dont le paramètre définit le type de trait à utiliser.

Souligné style perso

Remarquez qu'on a utilisé un trait défini avec la commande `\dashrule` du package `nccrules`.



Code
`ulem, nccrules`

```

\newcommand\myunderline{
  \bgroup\markoverwith{\dashrule[-0.5ex]{5 2 2 2 5}{1}}\ULon}
\myunderline{Souligné style perso}

```

2.3.5 Taille du texte

Pour spécifier la taille du texte, il faut utiliser des commandes qui vont vous permettre de changer la taille du texte. La taille de base est fixée grâce à une option de la commande `\documentclass`. On va par exemple écrire :

```
\documentclass[11pt]{report}
```

Les valeurs possibles sont 10pt, 11pt ou 12pt, la valeur par défaut étant 10pt. Les commandes de taille sont :

<code>\tiny</code>	(minuscule)	<code>\large</code>	(grand)
<code>\scriptsize</code>	(très petit)	<code>\Large</code>	(plus grand)
<code>\footnotesize</code>	(assez petit)	<code>\LARGE</code>	(très grand)
<code>\small</code>	(petit)	<code>\huge</code>	(énorme)
<code>\normalsize</code>	(normal)	<code>\Huge</code>	(géant)

Une autre solution pour modifier la taille de la police est de passer par la commande `\fontsize`.

ABC

La commande prend deux paramètres qui sont la taille du texte en points et l'espace interligne. La valeur du second paramètre doit environ être 20% supérieur au premier.



Code

```
\fontsize{30}{36}
\selectfont
ABC
```


La commande `\selectfont` permet de valider le choix de la nouvelle police. De plus, \LaTeX impose des limites sur la taille maximale que vous pouvez avoir. Pour contourner cette limite, il suffit d'insérer le package `fix-cm`.

2.3.6 Couleur

Grâce au package `color`, on va pouvoir modifier la couleur du texte inséré. La commande `\color` est de type déclarative tandis que la commande `\textcolor` prend en paramètre le texte dont il faut changer la couleur.

Noir, rouge, bleu, encore noir et gris.

Les commandes prennent en paramètre la couleur qu'on souhaite. Différentes valeurs sont prédéfinies : `black`, `white`, `gray`, `red`, `green`, `blue`, `cyan`, `magenta` et `yellow`.




Code

```
\bf Noir, {\color{red} rouge, \color{blue} bleu},
encore noir et \textcolor{gray}{gris}.
```

color

On peut également définir ses propres couleurs en utilisant la commande `\definecolor`. Celle-ci prend trois paramètres : un nom, un modèle de couleur et les valeurs de la couleur (nombre compris entre 0 et 1). Il existe trois modèles différents :

- `gray` (niveau de gris) une valeur
- `rgb` (RGB) trois valeurs pour rouge, vert et bleu
- `cmyk` (CMYK) quatre valeurs pour cyan, magenta, jaune et noir



Code color

```

\definecolor{darkgray}{gray}{0.2}
\definecolor{lightpurple}{rgb}{0.7,0,0.7}
\definecolor{darkgreen}{cmyk}{0.7,0,0.7,0.5}

\textcolor{lightpurple}{Texte en mauve clair}

```

Enfin, notez qu'on peut également spécifier une couleur par ses valeurs avec les commandes `\color` et `\textcolor`. Il suffit de préciser le modèle de couleur en option. On peut par exemple écrire :

```
\textcolor[rgb]{0.99,0.69,0.07}{Texte en orange clair}
```

2.4 Mise en page avancée

Dans cette section, on va voir quelques commandes permettant de faire des mises en page avancées de texte. On verra comment mettre en page des poèmes, faire des paragraphes avec des formes particulières, du texte sur plusieurs colonnes et on terminera par voir comment présenter des nombres avec des unités et comment insérer des URLs.


2.4.1 Citation

Pour insérer des citations ou des extraits de roman, on peut utiliser les environnements `quote` ou `quotation`. Les marges sont plus larges et la largeur occupée par le texte sur la page est réduite.

Il l'observa, occupée à tailler ses rosiers. Elle avait une silhouette magnifique, toute en longueur, mais pourtant, on la devinait sportive et tonique. Il prit le temps de la détailler.

Sa robe rouge, décontractée mais à la coupe parfaite épousait la moindre des formes de son corps de rêve. Le décolleté était simplement ensorcelant. [...]

La différence entre les deux environnements est que `quotation` ajoute un alinéa devant chaque paragraphe.



Code

```

\begin{quote}\small
Il l'observa, occupée à tailler ses rosiers. Elle avait ...
\end{quote}

```

2.4.2 Poème

Pour insérer un poème, on peut utiliser l'environnement `verse`. Chacune des lignes du poème doit se terminer par `\\`.

Demain, dès l'aube, à l'heure où blanchit la campagne,
Je partirai. Vois-tu, je sais que tu m'attends.
J'irai par la forêt, j'irai par la montagne.
Je ne puis demeurer loin de toi plus longtemps.



Code

```
\begin{verse}\small
Demain, dès l'aube, à l'heure où blanchit la campagne, \\
Je partirai. Vois-tu, je sais que tu m'attends. \\
J'irai par la forêt, j'irai par la montagne. \\
Je ne puis demeurer loin de toi plus longtemps.
\end{verse}
```

2.4.3 Forme des paragraphes

On peut obtenir des paragraphes qui ont des formes particulières en utilisant le package `shapepar`. Celui-ci propose des formes prédéfinies, mais il permet également d'en définir des nouvelles.

Une seule
chose est certaine. Une
fois que vous aurez goûté
à \LaTeX , vous ne pourrez
plus vous en passer.
Vous en serez litté-
ralement amou-
reux!
♡



Code

`shapepar`

```
\heartpar{\small Une seule chose est certaine. Une fois que vous
aurez goûté à \LaTeX, vous ne pourrez plus vous en
passer. Vous en serez littéralement amoureux !}
```

Les autres formes possibles prédéfinies dans le package sont :

- `\squarepar` : carré (□)
- `\circlepar` : cercle (○)

- `\diamondpar` : diamant (◇)
- `\heartpar` : cœur (♡)
- `\starpar` : étoile à cinq branches (★)
- `\hexagonpar` : hexagone (⬡)
- `\nutpar` : écrou hexagonal

2.4.4 Texte sur plusieurs colonnes

Pour avoir un texte sur plusieurs colonnes, il suffit d'utiliser l'environnement `multicols` disponible avec le package `multicol`. L'environnement prend un paramètre qui est le nombre de colonnes désirées. En option, on peut préciser un texte qui s'étendra sur toutes les colonnes.

Chapeau du texte en colonnes

Ce texte s'étend sur plusieurs colonnes qui ont été obtenues grâce à l'environne-	ment <code>multicols</code> disponible en important le package <code>multicol</code> . Un filet de séparation	est obtenu en définissant la longueur <code>columnseprule</code> .
---	---	--

On peut définir l'espacement entre deux colonnes avec la longueur `\columnsep` et la largeur du trait de séparation avec `\columnseprule`.

Code
multicol

```

\setlength{\columnsep}{7mm}
\setlength{\columnseprule}{1pt}

\begin{multicols}{3}[\textbf{Chapeau du texte en colonnes}]
\small Ce texte s'étend sur plusieurs colonnes ...
\end{multicols}

```

Pour forcer un saut de colonne, il suffit d'utiliser la commande `\columnbreak`.

2.4.5 Nombre et quantité

Pour mettre en forme proprement des nombres, on peut utiliser la commande `\numprint` du package `numprint`.

```


\numprint{150000.7829}      150 000,782 9
\numprint{2e-15}           2 · 10-15

```

La commande `\numprint` prend une option qui est une unité de mesure. Le nombre et l'unité seront placés dans une seule boîte inséparable. Enfin, on peut également utiliser les commandes `\cntprint` et `\lenprint` pour afficher respectivement des compteurs et des longueurs.

12,5 kg de pommes et 1,5 de lait
 1 pt = 0,351 46 mm = 0,013 84 in

Remarquez que pour l’affichage d’une longueur, l’option permet de définir dans quelle unité on souhaite afficher la longueur (la valeur par défaut étant le point). Par contre, pour `\numprint`, l’option indique juste quelle unité on souhaite afficher.

 **Code**
numprint

```

\numprint[kg]{12.5} de pommes et \numprint{1.5} de lait

\newlength{\one}
\setlength{\one}{1cm}

\lenprint{\one} = \lenprint[mm]{\one} = \lenprint[in]{\one}

```

Système International d’unités (SI)

Le package `siunitx` est plus récent et offre des possibilités plus riches. Il permet de représenter les données selon les normes du système international d’unités (SI). Les commandes essentielles sont `\num`, `\si`, `\ang`, `\SI`, `\numrange` et `\SIrange`.

La commande `\num` permet de représenter les nombres.

10 000.123 45	<code>\num{10000.12345}</code>
1.2×10^{-15}	<code>\num{1.2e-15}</code>
$5 \times 2 \times 10^{10}$	<code>\num{5 x 2e10}</code>

Les unités sont obtenues avec la commande `\si`. On peut les écrire soit directement, soit en utilisant des commandes. On peut changer le style avec l’option `per-mode`.

kg m/s^2	<code>\si{kg.m/s^2}</code>
kg m s^{-2}	<code>\si{\kilogram\meter\per\second\squared}</code>
kg m/s^2	<code>\si[per-mode=symbol]{\kilogram\meter\per\second\squared}</code>

On formate les angles en utilisant la commande `\ang`. On peut également préciser les minutes et secondes.

12.7°	<code>\ang{12.7}</code>
$1^\circ 2' 3''$	<code>\ang{1;2;3}</code>
$1' 10''$	<code>\ang{;1;10}</code>

Enfin, pour insérer un nombre avec une unité associée, on peut soit utiliser `\num` suivi de `\si`, soit utiliser la commande `\SI` qui garantit que le

nombre de l'unité forment un bloc indissociable. L'option de la commande permet de préciser une unité qui vient avant le nombre.


```
15.5 N/m2      \SI{15.5}{N/m^2}
$5.99/kg       \SI[per-mode=symbol]{5.99}[\$/\per\kilogram]
```

Enfin, on peut représenter des intervalles avec les commandes `\numrange` et `\SIrange`.

```
11 to 42          \numrange{11}{42}
120 cm to 180 cm  \SIrange{120}{180}{\centi\meter}
```

Grâce à ce package, on pourra par exemple écrire :

Si vous achetez 200 g de pommes à \$4.99/kg,
vous devrez payer un total de ≈ 1 €.



sunitx, eurosym

Code

```
Si vous achetez \SI{200}{g} de pommes à \SI[per-mode=symbol]{4.99}[\$/\per\kilo\gram], \\
vous devrez payer un total de \SI{1}[\$\approx$]{\euro}.
```

2.4.6 URL

On peut insérer des URL en utilisant le package `url`. Celui-ci définit une commande `\url` qui permet d'insérer une URL qui pourra être coupée et s'étendre sur plusieurs lignes. On peut définir le style des URL en utilisant la commande `\urlstyle`.

Site web officiel : `\url{http://www.latex-howto.be}`
Donald Knuth : `\url{http://www-cs-faculty.stanford.edu/~uno/}`

Notez que si l'URL contient les caractères `%` ou `#`, la commande `\url` ne peut pas être utilisée dans le paramètre d'une autre commande.



url

Code

```
\urlstyle{sf}

Site web officiel : \url{http://www.latex-howto.be} \\
Donald Knuth : \url{http://www-cs-faculty.stanford.edu/~uno/}
```

Pour pouvoir utiliser une URL avec % ou # dans le paramètre d'une autre commande, il va falloir définir une nouvelle URL avec `\urldef`. Par exemple :

```
\urldef{\mywebsite}\url{http://dom.ext/d%20}
```

Ensuite, on peut utiliser la commande `\mywebsite` n'importe où. On pourra par exemple écrire :

```
\section{Mon site web : \mywebsite}
```

2.4.7 Liens hypertextes

Vous pouvez ajouter des liens hypertextes dans un document en utilisant le package `hyperref`. En ajoutant simplement ce package, vous verrez que toute une série de liens seront automatiquement créés pour les entrées de la table des matières, pour les références croisées, pour les notes de bas de page, les URL ...

Il est également possible d'insérer des liens explicitement avec la commande `\href`. Celle-ci prend deux paramètres que sont l'URL du lien et le texte qui doit apparaître dans le texte.

Vous pouvez faire des recherches avec Google par exemple.


Code
hyperref

Vous pouvez faire des recherches avec
`\href{http://www.google.be}{Google}` par exemple.

2.5 Dimensions et format de la page

Dans cette section, on va voir comment modifier le format de la page, ainsi que les dimensions qui définissent la position du texte, des entêtes et pieds de page ...

2.5.1 Maquette

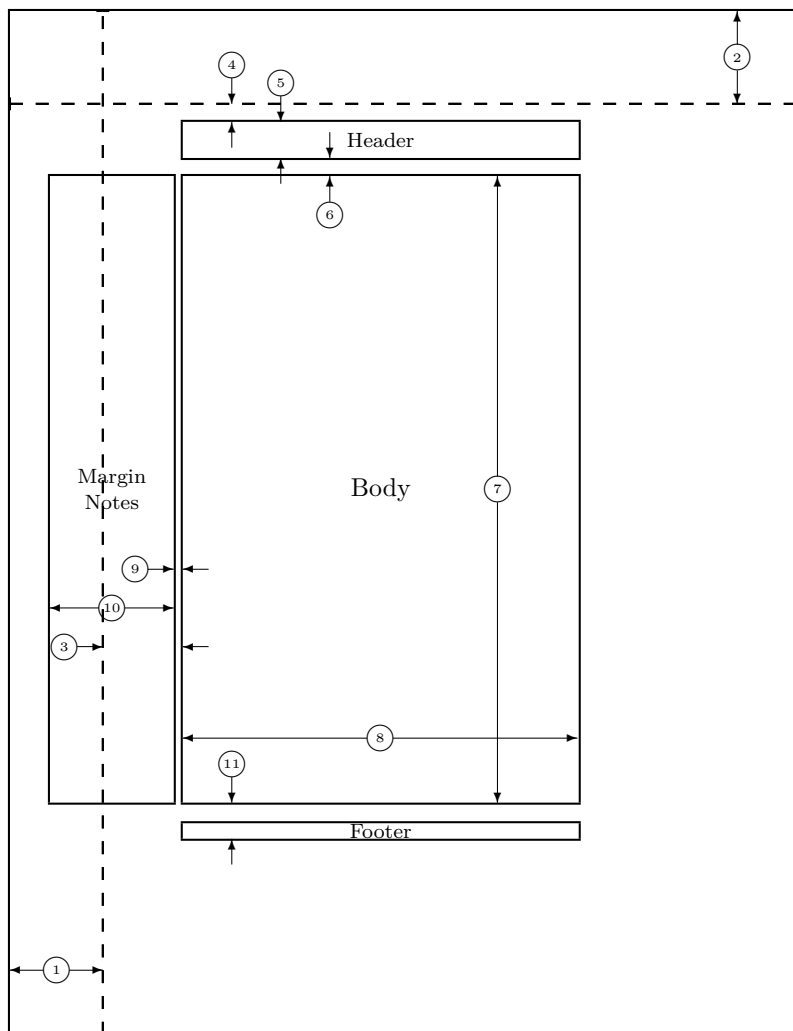
En utilisant la commande `\layout` du package de même nom, on va pouvoir insérer la maquette du document. Il s'agit d'une visualisation d'une page et des différentes dimensions attenantes. On y voit les quatre zones que sont le corps, l'entête, le pied de page et la zone des notes de marge. Toutes les dimensions et espacements sont présentés sur ce dessin. La maquette de ce livre est présentée ci-contre.

Voici les principales longueurs qui définissent un document :

- `\oddsidemargin` : marge sur le côté des pages impaires
- `\topmargin` : marge en haut de page, avant l'entête
- `\headheight` : hauteur de l'entête
- `\headsep` : distance entre l'entête et le haut du corps
- `\textheight` : hauteur du corps du document
- `\textwidth` : largeur du corps du document
- `\marginparsep` : distance entre le corps et les notes de marge
- `\marginparwidth` : largeur des notes de marge
- `\footskip` : distance entre le corps et le pied de page

Enfin, les deux longueurs `\paperwidth` et `\paperheight` contiennent la largeur et la hauteur du papier.

Toutes ces longueurs peuvent être modifiées avec la commande `\setlength`, afin de modifier la mise en page du document. L'utilisation de la maquette s'avère utile pour voir d'un coup d'œil la disposition des éléments sur la page. On verra qu'il y a des packages qui vont rendre ce travail plus facile.



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 62pt	4	\topmargin = 14pt
5	\headheight = 28pt	6	\headsep = 14pt
7	\textheight = 486pt	8	\textwidth = 307pt
9	\marginparsep = 7pt	10	\marginparwidth = 96pt
11	\footskip = 28pt		\marginparpush = 0pt (not shown)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 614pt		\paperheight = 794pt

2.5.2 Format prédéfini


En insérant le package `fullpage`, les marges vont être fixées de manière à ce que le texte occupe un maximum de place sur la page. Lors de l'import du package, il faut lui préciser en option le style du document (comme défini par la commande `\pagestyle`, voir section 2.3.1).

On peut également utiliser le package `savetrees` qui, comme son nom l'indique, permet d'économiser le papier. En plus de réduire au maximum les marges, il va également réduire les espacements, la taille de la police, utiliser des abréviations ...

2.5.3 Taille du papier

Voyons tout d'abord comment modifier la taille du papier. Par défaut, le papier est au format US letter (un peu plus petit que A4). Pour avoir du papier A4, il suffit de préciser l'option `a4paper` à la commande `\documentclass`.

Une autre manière de préciser le format du papier consiste à utiliser la commande `\setpapersize` du package `vmargin`. En option, on précise l'orientation de la page et en paramètre le format du papier : A0, A1 ..., A9, B0 ..., B9, C0 ..., C9, USletter, USlegal et USexecutive. On peut également donner trois paramètres pour définir une taille personnalisée.


Code
vmargin

```


\setpapersize{A4} % papier A4
\setpapersize[landscape]{A5} % papier A5, en mode paysage
\setpapersize{custom}{10cm}{2cm} % 10cm de large sur 2cm de haut

```

On peut également utiliser le package `geometry` décrit à la section suivante.

2.5.4 Définition des marges

Pour facilement définir les marges d'un document, on peut utiliser la commande `\setmargins` du package `vmargin`. Cette commande prend huit paramètres qui sont respectivement la marge gauche, la marge du haut, la largeur et hauteur du corps, la hauteur de l'entête, l'écart entre l'entête et le corps, la hauteur du pied de page et l'écart entre le corps et le pied de page.



Code vmargin

```
\setmargins{1cm}{1cm}{17cm}{25cm}{1cm}{5mm}{1cm}{5mm}
```

Des variantes de la commande existent, vous les retrouverez dans la documentation du package.

On peut également définir les marges avec le package `geometry`. Les valeurs des différentes longueurs sont spécifiées en option lors du chargement du package. Il y a tout d'abord le corps du document avec son entête, son pied de page et la zone pour les notes de marge. Viennent ensuite les marges gauche, droite, d'en haut et d'en bas. Voici les options les plus courantes :

- `paper` : spécifie la taille du papier (nom)
- `paperwidth` et `paperheight` : spécifient la taille du papier (longueur)
- `width` et `height` : dimension totale du corps du document
- `textwidth` et `textheight` : dimension de la zone du texte
- `includehead`, `includefoot`, `includeheadfoot` et `includemp` : sont utilisés pour indiquer si on souhaite ou non l'entête ou le pied de page ou la zone pour les notes de marge
- `headheight` et `headsep` : spécifient la hauteur de l'entête et la distance entre celle-ci et le texte
- `foot` : spécifie la distance entre la fin du texte et le pied de page
- `left`, `right`, `top` et `bottom` : spécifient les marges gauche, droite, d'en haut et d'en bas
- `hcentering`, `vcentering` et `centering` : permettent de centrer le contenu horizontalement et/ou verticalement

L'exemple donné en haut de page peut donc être obtenu en important simplement le package suivant :



Code geometry

```
\usepackage[paper=a4paper,includeheadfoot,Lleft=1cm,top=1cm,
textwidth=17cm,textheight=25cm,headheight=1cm,headsep=5mm,
foot=5mm]
```


2.5.5 Composer un PDF

Le package `pdfpages` permet d'effectuer facilement des opérations sur des fichiers PDF. La première chose qu'on peut faire consiste à insérer certaines pages d'un document PDF dans un document \LaTeX . On utilise pour cela la commande `\includepdf`. On spécifie les pages que l'on souhaite insérer avec la possibilité de définir des intervalles ou d'insérer une page blanche avec `{}`, avec l'option `pages`.

L'exemple suivant insère la page 1, suivie d'une page blanche, suivie des pages 5 à 9, suivies de la page 15 du document `monDocument.pdf`.

```
\includepdf[pages={1, {}, 5-9, 15}]{monDocument.pdf}
```

Il est également possible d'obtenir plusieurs pages par feuille. On utilise pour cela l'option `nup`. On définit ensuite l'espacement à mettre entre les pages logique avec l'option `delta` et on peut avoir une bordure autour des pages logiques avec l'option `frame`.

Par exemple, pour insérer toutes les pages du document `monDocument.pdf`, avec 3×2 pages par feuille, séparées par 5mm et une bordure, il faut écrire :

```
\includepdf[pages=-,nup=3x2,frame]{monDocument.pdf}
```

Enfin, on peut également insérer des pages en provenance de plusieurs fichiers PDF en utilisant la commande `\includepdfmerge` qui possède les mêmes options que `\includepdf` sauf l'option `pages`.

La commande suivante insère la première page du document `docA.pdf`, les pages 5 à 10 du document `docB.pdf` et enfin les pages 3 et 9 du document `docC.pdf`.

```
\includepdfmerge{docA.pdf, 1, docB.pdf, 5-10, docC.pdf, 3, 9}
```


CHAPITRE 3

Listes

Ce chapitre décrit comment insérer des listes d'éléments dans un document. On verra tout d'abord les listes à puces, numérotées et de description. On verra ensuite comment personnaliser les puces ou le style de la numérotation. On terminera en regardant comment définir ses propres listes.

3.1 Insérer une liste


Les listes d'éléments sont représentées par les environnements `itemize`, `enumerate` et `description`. Chaque élément de la liste est ensuite inséré grâce à la commande `\item`.

3.1.1 Liste à puces

On insère une liste à puces avec l'environnement `itemize`. On peut changer les puces via l'option de la commande `\item`, ou de manière plus générale en redéfinissant la commande `\labelitemi`.

- Premier
- Second
- Troisième

Comme vous pouvez le voir sur l'exemple, vous pouvez redéfinir la commande `\labelitemi` localement afin que le changement ne s'applique qu'à cette liste.

 **Code**

```

\begin{itemize}\renewcommand{\labelitemi}{\bullet}
  \item Premier
  \item Second
  \item[-] Troisième
\end{itemize}


```

3.1.2 Liste numérotée

On insère une liste numérotée avec l'environnement `enumerate`. On peut changer le style de la numérotation des éléments de la liste en redéfinissant la commande `\theenumi` et en utilisant le compteur `enumi`.

- a. Premier
- b. Second
- c. Troisième

La numérotation par défaut utilise les chiffres arabes. Dans l'exemple ci-dessus, on a changé le compteur pour avoir des lettres romaines minuscules (`\alph`).

 **Code**

```

\begin{enumerate}\renewcommand{\theenumi}{\alph{enumi}}
  \item Premier
  \item Second
  \item Troisième
\end{enumerate}


```

3.1.3 Liste de définitions

Enfin, on insère une liste de description avec l'environnement `description`. Les termes sont indiqués par l'option de la commande `\item` et les définitions suivent.

- Émétophobie** Peur de vomir
- Villophobie** Peur des animaux velus
- Pantophobie**
Peur de tout

Comme vous pouvez le voir sur l'exemple, la définition se trouve par défaut sur la même ligne que le mot défini. Si vous souhaitez l'avoir sur une ligne séparée, il suffit de commencer la définition par `\hfill \\\` comme on l'a fait pour le troisième mot.

 **Code**

```

\begin{description}
  \item[Émétrophobie] Peur de vomir
  \item[Villophobie] Peur des animaux velus
  \item[Pantophobie] \hfill \\ Peur de tout
\end{description}

```

3.2 Style des listes

Dans cette section, on va voir comment personnaliser différents aspects des listes. On va commencer par s'intéresser au style des listes numérotées. Ensuite, on verra comment personnaliser l'aspect des listes (espacement, indentation ...).

3.2.1 Numérotation


On a déjà vu, à la section 3.1.2, comment on pouvait modifier le style de la numérotation en redéfinissant la commande `\theenumi`. Une autre possibilité consiste à utiliser le package `enumerate` qui ajoute une option à l'environnement `enumerate`. Cette dernière décrit la numérotation à appliquer, où on utilise les lettres A, a, I, i et 1 pour représenter le compteur dans différents styles.

Art. 1. Il est strictement interdit de fumer.

Art. 2. Il est autorisé d'aller une fois aux toilettes.

Art. 3. Il est strictement interdit de regarder par la fenêtre.

Vous remarquerez que pour avoir la lettre A, et pas le compteur en style lettre romaine majuscule, il a fallu l'entourer d'accolades.

 **Code** enumerate


```

\begin{enumerate}[\bf{A}rt. 1.]
  \item Il est strictement interdit de fumer.
  \item Il est autorisé d'aller une fois aux toilettes.
  \item Il est strictement interdit de regarder par la fenêtre.
\end{enumerate}

```

On peut obtenir exactement le même effet en utilisant le package `enumitem` qui permet également de faire beaucoup d'autres choses comme on le verra dans la suite du chapitre.

Ce dernier ajoute la possibilité de spécifier des options aux environnements `itemize`, `enumerate` et `description`. Il est incompatible avec le package `enumerate`. Les options prennent la forme `clé=valeur` et, pour modifier la numérotation, il faut utiliser l'option `label=desc`. Notez on utilise les commandes `\Alph*`, `\alph*`, `\Roman*`, `\roman*` et `\arabic*` pour représenter le compteur dans différents styles.

 **Code** enumitem


```
\begin{enumerate}[label=\bf{A}rt. \arabic*.]
  \item Il est strictement interdit de fumer.
  \item Il est autorisé d'aller une fois aux toilettes.
  \item Il est strictement interdit de regarder par la fenêtre.
\end{enumerate}
```

3.2.2 Espacement

On peut modifier diverses dimensions des listes. On peut modifier l'espacement entre éléments d'une liste en redéfinissant la commande de longueur `\itemsep`.

- Premier
- Second
- Troisième

On peut écrire `\setlength{\itemsep}{5mm}` ou utiliser la notation raccourcie `\itemsep=5mm`.


 **Code**

```
\begin{itemize}\setlength{\itemsep}{3mm}
  \item Premier
  \item Second
  \item Troisième
\end{itemize}
```

On peut modifier beaucoup d'autres choses en utilisant le package `enumitem`. L'exemple suivant montre une liste qui a été modifiée grâce à plusieurs options.

- ▷ Premier
- ▷ Second
- ▷ Troisième

L'option `label` permet de changer l'étiquette de chaque élément de la liste. Avec `itemsep`, on change l'espacement entre les éléments. Ensuite, `labelsep` permet de changer la distance entre l'étiquette et le texte des éléments, tandis que `font` change la police des étiquettes. Enfin, `leftmargin` spécifie la marge à gauche de la liste.

 **Code** enumitem

```

\begin{itemize}[label=${\triangleright}$,itemsep=0pt,
font=\bf\Large,labelsep=5mm,leftmargin=2cm]
  \item Premier
  \item Second
  \item Troisième
\end{itemize}

```

Si vous utilisez également le package `[french]{babel}`, vous devrez ajouter la commande suivante après ajout du package `babel` pour pouvoir utiliser le package `enumitem` :

```
\frenchbsetup{CompactItemize=false}
```


Notez également qu'on peut spécifier des options qui seront appliquées à toutes les listes en utilisant les commandes `\setitemize`, `\setenumerate`, `\setdescription` et `\setlist`. Par exemple, pour fixer l'espacement entre les éléments de toutes les listes d'un document à cinq millimètres, il suffit d'ajouter :

```
\setlist{itemsep=5mm}
```

3.2.3 Liste dans le texte

Parfois, on souhaite insérer une liste numérotée, mais qui se trouve dans la continuité du texte. Pour faire cela, on peut utiliser l'environnement `inparaenum` disponible dans le package `paralist`. Voici un exemple d'une telle liste :

Pour installer le programme, il faut (a) désinstaller toute ancienne version, (b) défragmenter le disque et (c) lancer l'installateur.

 **Code** paralist

```

Pour installer le programme, il faut
\begin{inparaenum}[a]
  \item désinstaller toute ancienne version,
  \item défragmenter le disque et
  \item lancer l'installateur.
\end{inparaenum}


```

3.3 Gérer la numérotation

Les numéros des éléments des listes numérotées sont contrôlés par le compteur `enumi`. Vous pouvez changer sa valeur afin de personnaliser les numéros qui seront affichés.

1. Premier
2. Second
6. Troisième
- 9. Quatrième
- 8. Cinquième

Dans l'exemple ci-dessus, on a changé deux fois la valeur du compteur. Entre le deuxième et le troisième élément, on a changé la valeur du compteur en 5, ce qui fait que le troisième élément est numéroté 6. Entre le troisième et le quatrième élément, on a changé le compteur en -10 .



Code

```

\begin{enumerate}
  \item Premier
  \item Second

  \setcounter{enumi}{5}
  \item Troisième


  \setcounter{enumi}{-10}
  \item Quatrième
  \item Cinquième
\end{enumerate}

```

Le package `enumitem` propose des options pour modifier facilement la numérotation. L'option `start` indique le numéro auquel il faut commencer la liste. L'option `resume` permet de fixer la numérotation de manière à ce qu'elle continue celle de l'`enumerate` précédent.

10. Premier
 11. Second
- Bla bla bla
12. Troisième
 13. Quatrième

Dans cet exemple, on veut commencer la première liste numérotée à 10 (`start=10`). Ensuite, on a du texte et la numérotation de la liste suivante doit être dans la continuité de la première liste (`resume`).



Code enumitem

```

\begin{enumerate}[start=10]
  \item Premier
  \item Second
\end{enumerate}

Bla bla bla

\begin{enumerate}[resume]
  \item Troisième
  \item Quatrième
\end{enumerate}


```

3.4 Définir une liste

En plus des environnements `itemize`, `enumerate` et `description`, il existe également l'environnement `list` qui permet d'insérer une liste générique. On va notamment l'utiliser pour définir de nouveaux types de liste.

- * Premier
- * Deuxième
- * Troisième

Dans l'exemple ci-dessus, on a simplement défini une liste à puces dont les puces sont définies par la commande `\star` (une étoile à cinq branches). Le premier paramètre de l'environnement indique l'étiquette à ajouter à chaque élément, et le second paramètre permet de définir les espacements à appliquer.



Code

```

\begin{list}{\star}{}
  \item Premier
  \item Deuxième
  \item Troisième
\end{list}

```


Pour définir un nouveau type de liste à partir de cela, il suffit d'utiliser la commande `\newenvironment`. Voici comment on définit une nouvelle liste `mylist` :

```

\newenvironment{mylist}{\begin{list}{\star}{}{\end{list}}

```

On peut maintenant écrire :

 **Code**

```
\begin{mylist}
  \item Premier
  \item Deuxième
  \item Troisième
\end{mylist}
```

On peut également définir des listes numérotées. Pour cela, il faut d'abord définir un nouveau compteur dans le préambule du document avec :

```
\newcounter{mycounter}
```


Ensuite, on doit indiquer qu'on souhaite utiliser ce compteur avec la commande `\usecounter`. Enfin, pour faire apparaître la valeur du compteur, on utilise `\Alph` `\alph` `\Roman` `\roman` ou `\arabic` en fonction du style désiré. On peut par exemple définir :

```
\newenvironment{mynumberedlist}{
  \begin{list}{\alph{mycounter}}{\usecounter{mycounter}}
}{
  \end{list}
}
```

Vous remarquerez que le compteur utilisé doit être précisé dans le second paramètre de l'environnement `list`. Avec cette nouvelle liste, on peut par exemple obtenir :

- a) Premier
- b) Deuxième
- c) Troisième

Voici le code qu'on a utilisé pour obtenir cette liste :

 **Code**

```
\begin{mynumberedlist}
  \item Premier
  \item Deuxième
  \item Troisième
\end{mynumberedlist}
```

On peut également modifier différents paramètres d'espacement. Il y a tout d'abord les espacements verticaux qui peuvent être modifiés avec

`\parskip`, `\topsep`, `\parsep`, `\itemsep` et `\partopsep`. En ce qui concerne les espacements horizontaux, on peut utiliser `\leftmargin`, `\rightmargin`, `\labelwidth`, `\labelsep`, `\listparindent` et `\itemindent`. Consultez la documentation pour avoir plus d'informations sur ces longueurs.


3.5 Mise en page avancée

Dans cette section, on va voir quelques mises en page avancées de listes. On verra tout d'abord comment imbriquer des listes. Ensuite, on verra comment insérer des listes d'exercices.

3.5.1 Liste imbriquée

On peut imbriquer des listes les unes dans les autres. On pourrait par exemple insérer une liste de tâches pour la semaine :

- ★ Lundi
 - ◇ Faire les courses
 - ◇ Passer à la banque
- ★ Mercredi
 - ◇ Passer à la poste
- ★ Samedi
 - ◇ Chercher carte d'identité à la commune



Code

```

\renewcommand{\labelitemi}{\star}
\renewcommand{\labelitemii}{\diamond}
\begin{itemize}
  \item Lundi
    \begin{itemize}
      \item Faire les courses
      \item Passer à la banque
    \end{itemize}
  \item Mercredi
    \begin{itemize}
      \item Passer à la poste
    \end{itemize}
  \item Samedi
    \begin{itemize}
      \item Chercher carte d'identité à la commune
    \end{itemize}
\end{itemize>

```

Il s'agit ici de deux listes à puces qui ont été imbriquées. Remarquez qu'on a redéfini les étiquettes des éléments de la liste de premier niveau en changeant la commande `\labelitemi`. Pour les éléments des listes de second niveau, c'est la commande `\labelitemii` qui contrôle leurs étiquettes.

Les étiquettes des listes à puces et celles des listes numérotées sont contrôlées par une série de commandes, correspondant au niveau de la liste :


```
\labelitemi, \labelitemii, \labelitemiii et \labelitemiv
\labelenumi, \labelenumii, \labelenumiii et \labelenumiv
```

Comme vous pouvez le constater, il n'est pas possible d'avoir un niveau d'imbrication supérieur à 4 pour les listes. On peut dépasser cette limite en utilisant l'environnement `outline` du package de même nom. Celui-ci fonctionne comme l'environnement `enumerate` et permet un niveau d'imbrication de 6.

Enfin, la dernière solution consiste à utiliser l'environnement `easylist` du package de même nom. Avec cet environnement, la définition d'une liste imbriquée sera plus claire et on n'a plus la limite des niveaux.

1. un élément
 - 1.1. élément suivant
 - 1.2. encore un autre
 - 1.2.1. et un troisième
2. on revient au premier niveau

On peut modifier le style grâce à la commande `\ListProperties`. Le package est très riche et nous vous conseillons d'aller lire sa documentation pour en savoir plus.

 **Code**
easylist

```

\ListProperties(Progressive=5mm)
\begin{easylist}
  § un élément
  §§ élément suivant
  §§§ encore un autre
  §§§§ et un troisième
  § on revient au premier niveau
\end{easylist}

```


3.5.2 Liste en colonnes

L'environnement `tabenum` du package `tablists` permet d'insérer des listes d'éléments courts qui seront présentés en colonnes. Il est par exemple particulièrement adapté pour insérer une liste d'exercices comme le montre l'exemple suivant :

$$1) y = x^2 + 3 \quad 2) y = \sqrt{x-1} \quad 3) y = 2^{x+2}$$

$$4) y = \sin(2x-1) \quad 5) y = \frac{x+1}{x}$$

L'option de l'environnement est identique à celle de l'environnement `enumerate` une fois le package de même nom importé (voir section 3.2.1). Une ligne vide permet d'indiquer qu'il faut passer à la ligne suivante de la liste.

 **Code**
tablists

```

\begin{tabenum}[\bfseries 1]
  \tabenumitem $y = x^2 + 3$
  \tabenumitem $y = \sqrt{x - 1}$
  \tabenumitem $y = 2^{x + 2}$

  \tabenumitem $y = \sin (2x - 1)$
  \tabenumitem $y = \frac{x + 1}{x}$
\end{tabenum}


```

3.5.3 Mise en page des listes de définitions

On peut modifier la mise en page des listes de définitions en utilisant l'environnement `desclist` défini dans le package de même nom. L'environnement prend deux paramètres et une option. Les paramètres contiennent des commandes à insérer avant et après le mot défini. L'option de l'environnement contient un mot qui sera utilisé comme référence pour effectuer l'alignement des définitions.

Mot 1 — Définition
 Mot 2 — Définition
 Long mot — Définition
 Plus long mot — Définition

Dans notre cas, on a utilisé `Long mot` pour faire l'alignement. Les trois premières définitions sont donc alignées. Cependant, la dernière déborde, le mot défini étant plus long que `Long mot`.

**Code** desclist

```
\begin{desclist}{\sf}{\rm};\hfill---}[Long mot]
  \item[Mot 1] Définition
  \item[Mot 2] Définition
  \item[Long mot] Définition
  \item[Plus long mot] Définition
\end{desclist}
```

CHAPITRE 4

Images


Ce chapitre décrit comment insérer des images dans un document. On va commencer par voir la commande `\includegraphics` qui permet d'insérer une image. On verra ensuite comment ajouter une légende en utilisant l'environnement `figure`. Enfin, on verra comment personnaliser l'apparence : mettre deux images côte à côte, faire couler un paragraphe autour d'une image, insérer des parties d'image. On terminera par quelques commandes permettant de faciliter la gestion des images d'un document.

4.1 Insérer une image

On insère une image avec la commande `\includegraphics` disponible en chargeant le package `graphicx`. La commande prend en paramètre le chemin complet vers l'image à insérer et on peut spécifier toute une série de paramètres sous la forme `clé=valeur` en option.



Dans cet exemple, on insère l'image `running.png`, en spécifiant qu'elle doit avoir une largeur de deux centimètres.

 **Code****graphicx**


```
\includegraphics [width=2cm]{running.png}
```

4.1.1 Transformation de l'image

Grâce aux nombreuses options de la commande `\includegraphics`, on va pouvoir effectuer diverses transformations sur l'image.




Ici, on n'a pas précisé la taille de l'image en donnant sa largeur en centimètres, mais en spécifiant le facteur d'échelle à appliquer, par rapport à la taille originale de l'image. On utilise pour cela l'option `scale=factor`.

	Code	graphicx
<pre>\includegraphics[scale=0.05]{running.png} \includegraphics[scale=0.1]{running.png} \includegraphics[scale=0.2]{running.png}</pre>		

On a vu qu'on pouvait préciser la largeur de l'image avec `width=length`. On peut également préciser la hauteur avec `height=length`. Lorsque vous précisez les deux options, il se peut que l'image soit déformée. Pour forcer l'image à garder son aspect, il faut ajouter `keepaspectratio=true`.



Lorsque l'option `keepaspectratio=true` est présente, c'est toujours la dernière dimension demandée qui est prise en compte. Dans notre cas, c'est donc la hauteur de l'image qui sera fixée.

	Code	graphicx
<pre>\includegraphics[width=2cm,height=1cm]{running.png} \includegraphics[width=2cm,height=1cm,keepaspectratio=true] {running.png}</pre>		

Enfin, on peut également effectuer une rotation de l'image en utilisant l'option `angle=value`. Pour définir le centre de rotation, il faut utiliser

l'option `origin=label`. Le `label` se compose de lettres parmi `lrctbB` pour `left`, `right`, `center`, `top`, `bottom`, `Baseline`.



Vous remarquerez également que l'ordre des options est important. Dans le premier cas, on impose une largeur d'un centimètre, puis on effectue la rotation. Dans le deuxième cas, on procède dans l'autre sens. Dans le troisième cas, la rotation est effectuée par rapport au bas (b) de l'image.

Code
graphicx

```

\includegraphics[width=1cm,angle=45]{running.png}
\includegraphics[angle=45,width=1cm]{running.png}
\includegraphics[width=1cm,angle=45,origin=b]{running.png

```

4.1.2 Découpe d'une image

Grâce aux options `clip`, `trim` et `viewport`, on va pouvoir insérer une partie d'une image dans un document. Avec `trim`, on va pouvoir spécifier de combien il faut rogner l'image à gauche, en bas, à droite et en haut.



On a donc retiré 50 bp¹ de tous les côtés de l'image. Remarquez qu'il faut ajouter `clip=true` pour que le `trim` ait de l'effet.

Code
graphicx

```

\includegraphics[width=2cm]{images/running.png}
\includegraphics[width=2cm,trim=50 50 50 50,clip=true]
{running.png}

```

1. Un **bp** (big point) est une unité de mesure, avec 72 **bp** correspondant à un pouce.

Une autre technique pour découper une image consiste à utiliser l'option `viewport`. Cette fois-ci, on va préciser les coordonnées du coin inférieur gauche et supérieur droit de la zone de l'image qu'il faut afficher.



Ici, on a décidé de prendre la partie de l'image s'étendant de $(0,0)$ à $(500, 250)$. Remarquez qu'il faut également préciser `clip=true`.

Code
graphicx

```

\includegraphics[width=2cm]{running.png}
\includegraphics[width=2cm,viewport=0 0 500 250,clip=true]
{running.png}

```


4.1.3 Légende

Pour ajouter une légende à un image, il faut la placer dans un environnement `figure` et utiliser la commande `\caption`. Attention au fait que l'environnement `figure` définit un nouvel objet flottant : vous n'avez donc plus le contrôle absolu sur le placement de votre image. Voici un exemple :



Figure 4.1. Bonhomme qui court.

Afin de centrer l'image, il faut utiliser la commande `\centering`. Vous pouvez placer la légende au-dessus ou en-dessous de l'image en changeant la position de la commande `\caption`.

 **Code**
graphicx

```

\begin{figure}[!h]
  \centering
  \includegraphics[width=2cm]{running.png}
  \caption{Bonhomme qui court.}
\end{figure>


```

Parfois, on souhaite avoir la légende à côté de la figure. Une première solution consiste à le faire manuellement, grâce à deux `minipage`. L'autre solution consiste à utiliser le package `sidecap` qui propose un environnement `SCfigure`.



Figure 4.2. Bonhomme qui court.

La légende sera placée à gauche ou à droite en fonction des options utilisées lors du chargement du package. La première option de l'environnement `SCfigure` définit la proportion de place laissée pour la légende par rapport à la place prise par l'image.

 **Code**
graphicx, sidecap

```


\begin{SCfigure}[1.2][!h]
  \centering
  \includegraphics[width=2cm]{running.png}
  \caption{Bonhomme qui court.}
\end{SCfigure>

```

Enfin, on aimerait parfois pouvoir ajouter une légende à une image, sans toutefois la placer dans un environnement `figure` flottant. Pour cela, il faut utiliser la commande `\captionof` disponible en important le package `caption`.



Figure 4.3. Bonhomme qui court.

 **Code**
graphicx, caption


```
\includegraphics[width=2cm]{running.png}
\captionof{figure}{Bonhomme qui court.}
```

On utilisera notamment la commande `\captionof` dans les images placées dans la marge du document, et parfois également dans les documents à plusieurs colonnes, lorsque les environnements flottants sont interdits.

4.1.4 Image sur toute la largeur

On peut insérer une image sur toute la largeur de la page en utilisant l'option `width` et en lui donnant comme valeur la longueur `\textwidth` représentant la largeur de la page.



 **Code**
graphicx

```
\includegraphics[width=\textwidth,trim=0 0 0 650,clip=true]
{Tafoni_by_Endico.jpg}
```

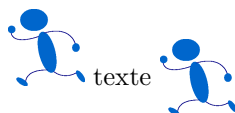
On peut également avoir une image qui occupe la moitié de la largeur avec `width=0.5\textwidth`, etc.

4.2 Positionnement

On va maintenant s'intéresser au positionnement de l'image dans le document, par rapport au texte ou par rapport à d'autres images. On va également s'intéresser aux sous-figures.

4.2.1 Alignement vertical

L'alignement vertical d'une image ne peut être défini via une option de la commande `\includegraphics`. La solution pour ajuster une image verticalement consiste à utiliser la commande `\raisebox` et de spécifier une hauteur négative.



Le seul souci avec cette technique est qu'il faut manuellement ajuster la valeur à donner à la commande `\raisebox` pour avoir l'alignement souhaité.

Code

graphicx

```

\includegraphics[width=1cm]{running.png}
texte
\raisebox{-4mm}{\includegraphics[width=1cm]{running.png}}

```

4.2.2 Images côte à côte

Pour avoir deux images l'une à côté de l'autre, il suffit de placer les `\includegraphics` les uns à la suite des autres, comme on a toujours fait depuis le début de ce chapitre. Maintenant, si on souhaite placer deux figures avec légende côte à côte, on va devoir utiliser des `minipage`.




Figure 4.4. Grr.



Figure 4.5. Youpie.

On utilise donc deux environnements `minipage`, chacun ayant sa propre légende. Le contenu de chaque `minipage` est centré horizontalement, ainsi que le contenu de la `figure`. Enfin, on aligne `minipage` verticalement avec l'option `t` afin d'avoir les deux légendes au même niveau.

 **Code** graphicx

```

\begin{figure}[!ht]
  \centering
  \begin{minipage}[t]{4cm}
    \centering
    \includegraphics[width=1.5cm]{angry.png}
    \caption{Grr.}
  \end{minipage}
  \begin{minipage}[t]{4cm}
    \centering
    \includegraphics[width=1.5cm]{happy.png}
    \caption{Youpie.}
  \end{minipage}
\end{figure}

```


4.2.3 Images dans la marge

On peut facilement placer une image dans la marge d'un document en utilisant la commande `\marginpar`. Lorsque l'on fait cela, on ne peut pas utiliser l'environnement `figure` et il faut donc utiliser la commande `\captionof` vue à la section 4.1.3 pour ajouter une légende.



Figure 4.6.
Youpie.

La largeur du texte a été localement modifiée afin que vous puissiez voir l'image dans la marge. On verra plus plus tard comment faire cela. Ici, on s'intéresse aux images dans la marge.

 **Code** graphicx, caption


```

\marginpar{
  \centering
  \includegraphics[width=1cm]{happy.png}
  \captionof{figure}{Youpie.}
}

```

4.2.4 Rotation d'image

On a vu précédemment qu'on pouvait effectuer une rotation sur une image en utilisant l'option `angle=value`. Maintenant, si on désire ajouter une légende à l'image, soit on fait comme on a vu à la section 4.1.3, mais alors la légende ne sera pas pivotée avec l'image, soit on utilise l'environnement `sidewaysfigure` du package `rotating`. L'image sera automatiquement tournée de 90 ou 270 degrés selon qu'elle se trouve sur une page paire ou impaire, et elle sera forcément placée seule sur une page.



Code

graphicx, rotating

```


\begin{sidewaysfigure}
  \centering
  \includegraphics[width=2cm]{running.png}
  \caption{Bonhomme qui court.}
\end{sidewaysfigure>

```

4.2.5 Texte coulant autour d'une image

Parfois, on désire insérer une image qu'on fait flotter avec le texte environnant qui coule autour. Ici par exemple, on a une image sur la droite et le texte de ce paragraphe coule autour. Pour cela, on va utiliser l'environnement `wrapfigure` du package `wrapfig`. On précise en option le nombre de lignes de texte se trouvant à la même hauteur que l'image et ensuite, en paramètres, on précise de quel côté il faut placer l'image et quelle est la largeur de la zone autour de laquelle le texte doit couler.





Code

graphicx, wrapfig

```

\begin{wrapfigure}[5]{r}{2.5cm}
  \vspace{-7mm}
  \includegraphics[width=2cm]{running.png}
\end{wrapfigure>

```

Pour ajouter une légende, il suffit d'utiliser la commande `\captionof` qu'on a vue à la section 4.1.3.

4.2.6 Sous-figures

Enfin, pour terminer cette section, voyons maintenant comment avoir des sous-figures. Il suffit d'utiliser la commande `\subfigure` disponible en important le package `subfigure`. La commande prend en option la légende et en paramètre l'image.

Chacune des images peut donc avoir sa propre légende et on peut également conserver une légende pour la figure dans sa globalité.

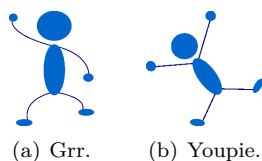


Figure 4.7. Humeurs.

Code
graphicx, subfigure

```

\begin{figure}[!h]
  \centering
  \subfigure[Grr.]{
    \includegraphics[height=1.5cm]{angry.png}
  }
  \quad
  \subfigure[Youpie.]{
    \includegraphics[height=1.5cm]{happy.png}
  }
  \caption{Humeurs.}
\end{figure>

```

4.3 Gestion des images

Le type des images qu'il est possible d'inclure dans un document L^AT_EX dépend du compilateur utilisé. Si vous travaillez avec `latex` pour créer des fichiers DVI, vous pourrez insérer des images postscript. En utilisant `pdflatex`, on peut insérer des images PNG, JPG et PDF.

Vous pouvez spécifier le chemin complet ou relatif à la commande `\includegraphics`. Une autre solution consiste à donner la liste des chemins où chercher les images avec la commande `\graphicspath`.

Il est également possible de déclarer les extensions à chercher en utilisant la commande `\DeclareGraphicsExtensions`. Cela permet d'écrire les noms de manière courte. On va ainsi par exemple pouvoir écrire :

Code
graphicx

```

\graphicspath{{images/}{/home/img/}}
\DeclareGraphicsExtensions{.png,.jpg}

```


Ainsi, si on utilise la commande

```
\includegraphics{filename}
```

l'image sera d'abord cherchée dans le répertoire courant, puis dans `images/` et enfin dans `/home/img/`. Le compilateur cherchera d'abord le fichier `filename.png`, puis `filename.jpg`.

4.4 Insérer une photo

Pour insérer une photo, on peut profiter de la commande `\putphoto` ou de l'environnement `Photo` du package `photo`. On peut ainsi préciser une légende ainsi que le nom du photographe. La seule différence entre la commande ou l'environnement est que ce dernier est un flottant. On peut obtenir la liste des photos d'un document avec la commande `\listofphotos`.

Photo 1 – Tafoni, des sculptures dans le sable.



Dawn Endico

Le positionnement de la photo, de la légende et du photographe sont contrôlés par l'option de la commande et de l'environnement. La première lettre vaut `l`, `r`, `i` ou `o` (left, right, in ou out) et permet de placer la photo. La seconde vaut `t`, `c` ou `b` (top, center ou bottom) et définit l'alignement vertical de la légende. Enfin, la troisième vaut `u` ou `s` (under ou side) et définit la position du photographe.

Code	photo, graphicx
<pre>\begin{Photo}[obu]{photo:book-cover}{Dawn Endico} {Tafoni, des sculptures dans le sable.} \includegraphics[width=4cm]{Tafoni_by_Endico.jpg} \end{Photo}</pre>	

CHAPITRE 5

Tableaux

Ce chapitre décrit comment insérer des tableaux de données dans un document. On va commencer par voir les environnements `tabbing` et `tabular` qui sont les plus utilisés. On verra ensuite comment ajouter une légende à un tableau en utilisant l'environnement `table`. Après cela, on verra comment personnaliser l'aspect des tableaux : épaisseur des filets, alignement du texte, largeur des tableaux, couleurs et grands tableaux. Enfin, on terminera avec quelques notions de mise en page de tableaux.

5.1 Insérer un tableau

On insère un tableau avec l'environnement `tabular` qui prend un paramètre qui décrit les colonnes du tableau. Une nouvelle colonne est définie à l'aide d'une lettre qui décrit l'alignement horizontal du texte dans cette colonne. Les valeurs possibles sont :


- `l` texte aligné à gauche ;
- `c` texte est centré ;
- `r` texte est aligné à droite.

On définit ensuite les données à l'intérieur de l'environnement. On remplit le tableau de gauche à droite et de haut en bas. On passe à la colonne suivante avec `&` et on passe à la ligne suivante avec `\\`.

Voyons par exemple comment construire le tableau suivant :

Article	Prix
Livres	12
DVD	17

Il s'agit d'un tableau qui comporte deux colonnes. Le texte de la première est aligné à gauche et celui de la seconde est centré. Le tableau contient trois lignes de texte. Le code pour l'obtenir est :

 **Code**


```
\begin{tabular}{lc}
Article & Prix \\
Livres & 12 \\
DVD & 17
\end{tabular}
```

5.1.1 Filets

On peut ajouter des filets verticaux et horizontaux à un tableau. Les filets verticaux sont définis dans l'option de l'environnement `tabular`. Il suffit d'insérer `|` partout là où on souhaite un filet vertical. Les filets horizontaux sont définis avec les données du tableau. La commande `\hline` insère un filet horizontal. Elle doit toujours être placée avant les données de la ligne. Voyons comment obtenir le tableau suivant :

Article	Prix
Livres	12
DVD	17

On voit sur l'exemple qu'il y a à la fois des filets simples et des filets doubles. Ce n'est pas bien difficile à obtenir, il suffit de placer deux filets.

 **Code**


```
\begin{tabular}{||l|c||}
\hline
\bf Article & \bf Prix \\
\hline\hline
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular}
```

Il est également possible d'avoir des filets horizontaux partiels qui ne s'étendent que sur certaines colonnes. On les définit à l'aide de la commande `\cline` qui prend en paramètre les colonnes sur lesquelles le filet doit s'étendre.

La commande prend un paramètre qui est composé de deux numéros de colonnes séparés par un tiret -. On peut utiliser plusieurs fois la commande si on veut plusieurs filets partiels.

A	B	C	D
---	---	---	---

Voici le code de ce tableau :



Code

```

\begin{tabular}{1111}
  A & B & C & D \\
\cline{1-2}\cline{4-4}
\end{tabular}

```


5.1.2 Légende

Pour ajouter une légende à un tableau, il faut le placer dans un environnement `table` et utiliser la commande `\caption`. Attention au fait que l'environnement `table` définit un nouvel objet flottant et que vous n'avez donc plus le contrôle absolu sur le placement de votre tableau. Voici un exemple :

Article	Prix
Livres	12
DVD	17

Table 5.1. Liste des prix.

Afin de centrer le tableau, il faut utiliser la commande `\centering`. Vous pouvez placer la légende au-dessus ou en-dessous du tableau en changeant la position de la commande `\caption`.



Code

```

\begin{table}[!h]
  \centering
  \begin{tabular}{|l|c|}
    \hline
    \bf Article & \bf Prix \\
    \hline\hline
    Livres & 12 \\
    DVD & 17 \\
    \hline
  \end{tabular}
  \caption{Liste des prix.}
\end{table}


```

5.1.3 Cellule paragraphe

Par défaut, les largeurs des colonnes dépendent de leur contenu. On peut insérer une colonne dont on fixe la largeur en utilisant le descripteur `p{length}`. Voici un exemple avec la seconde colonne qui fait deux centimètres de large.

Article	Description	Prix
Livres	Livres A5 avec couverture souple	12
DVD	DVD vierge vendus par pack de cinq	17

Comme vous pouvez le voir, le texte de ce type de cellule est par défaut justifié. On verra plus loin comment modifier cela.



Code

```

\begin{tabular}{|l|p{2.1cm}|c|}
\hline
\bf Article & \bf Description & \bf Prix \\
\hline
Livres & Livres A5 avec couverture souple & 12 \\
DVD & DVD vierge vendus par pack de cinq & 17 \\
\hline
\end{tabular>


```

5.1.4 Séparateur de colonne

Entre deux colonnes, jusqu'à présent, on avait soit rien du tout, soit un filet vertical (avec `|`). On peut définir son propre séparateur de colonne avec `@{text}`. Voyons par exemple comment avoir le tableau suivant dont chaque ligne contient deux nombres séparés par un `/`.

12/736
9231/1
9/49

De plus, on voit que le texte de la première colonne est aligné à droite et celui de la seconde colonne à gauche. Pour avoir le slash entre les deux colonnes, il suffit d'utiliser le séparateur `@{/}`.



Code

```
\begin{tabular}{r@{/}l}
  12 & 736 \\
  9231 & 1 \\
  9 & 49 \\
\end{tabular}
```

5.1.5 Répéter un descripteur

Pour répéter plusieurs fois un même descripteur, par exemple pour avoir un tableau de 10 colonnes centrées, au lieu d'écrire `|c|c|c|...`, on peut utiliser le descripteur `*{int}{desc}` qui va répéter `int` fois le descripteur `desc`. Ainsi, pour avoir un tableau à 10 colonnes centrées, toutes séparées par un filet vertical, on peut écrire :

```
\begin{tabular}{*{10}{|c|}}
```

ou

```
\begin{tabular}{|*{10}{c|}}
```

5.2 Cellules spéciales

On va voir qu'il est possible d'avoir des cellules qui s'étendent sur plusieurs colonnes ou lignes. Pour cela, il va falloir utiliser respectivement les commandes `\multicolumn` et `\multirow`. Pour la seconde commande, il faudra importer le package `multirow`. On verra également qu'on peut avoir des cellules coupées par une barre oblique.

5.2.1 Cellule sur plusieurs colonnes


La commande `\multicolumn` prend trois paramètres :

- Le nombre de colonnes sur lesquelles la cellule s'étend ;
- La description de la cellule ;
- Le texte de la cellule.

On va par exemple pouvoir définir un tel tableau :

Produit	Année		
	2008	2009	2010

La seconde cellule de la première ligne s'étend sur trois colonnes. Il ne lui faut pas de filet vertical sur la gauche, mais bien sur la droite.

 **Code**

```
\begin{tabular}{|l|c|c|c|}
\hline
\bf Produit & \multicolumn{3}{c}{\bf Année} \\
\hline
& 2008 & 2009 & 2010 \\
\hline
\end{tabular}
```

On verra à la fin de ce chapitre qu'on va également utiliser `\multicolumn` pour supprimer certains filets verticaux pour des mises en page avancées.

5.2.2 Cellule sur plusieurs lignes


Pour avoir une cellule qui s'étend sur plusieurs lignes, il faut utiliser la commande `\multirow` du package `multirow`. Elle prend trois paramètres :

- Le nombre de lignes sur lesquelles la cellule s'étend ;
- La largeur de la cellule ;
- Le texte de la cellule.

On va par exemple pouvoir définir un tel tableau :

		Produit
Année	2008	
	2009	
	2010	

Il faut définir le `\multirow` à l'endroit où se trouverait la première cellule. Pour toutes les lignes suivantes, on va devoir passer à la colonne suivante avec `&`, sans quoi du texte viendrait encombrer la cellule étendue.

 **Code** **multirow**


```
\begin{tabular}{|l|l|l|l|}
\hline
& & \bf Produit \\
\hline
\multirow{3}{12mm}{\bf Année} & 2008 & \\
& 2009 & \\
& 2010 & \\
\hline
\end{tabular}
```


5.2.3 Cellule coupée par une barre oblique

Grâce à la commande `\backslashslashbox` du package `slashbox`, on va pouvoir diviser une cellule en deux parties séparées par une barre oblique.

	B		
A	/	1	2
1		✓	
2		✓	✓

Les ✓ ont été obtenus avec la commande `\checkmark` disponible en important le package `amssymb`.


Code
slashbox, amssymb

```

\begin{tabular}{|l|l|l|l|}
\hline
\backslashslashbox{A}{B} & 1 & 2 & \\
\hline
1 & \checkmark & & \\
\hline
2 & \checkmark & \checkmark & \\
\hline
\end{tabular}

```


La commande `\slashbox` permet d'avoir une cellule divisée par rapport à l'autre diagonale.

5.2.4 Rotation du texte

On souhaite parfois effectuer une rotation du texte dans une cellule. Il suffit d'utiliser la commande `\rotatebox` du package `graphicx`.

Bla	Bla	Bla
-----	-----	-----

Remarquez qu'on a dû ajouter une espace fine avec `\,` pour éviter que le texte ne colle à la bordure supérieure du tableau.


Code
graphicx

```

\begin{tabular}{|c|c|c|}
\hline
Bla & \rotatebox{90}{Bla\,} & Bla \\
\hline
\end{tabular}

```

5.3 Descripteurs de colonne avancés

Grâce au package `array`, de nombreux descripteurs de colonne complémentaires sont ajoutés. On va également pouvoir définir ses propres descripteurs très facilement.


5.3.1 Nouveaux descripteurs

On a vu le descripteur `p{length}` qui permet d'indiquer que les cellules d'une colonne seront des paragraphes de largeur fixe. C'est comme si le contenu de chacune de ces cellules était dans une `\parbox[t]{length}`, c'est-à-dire que le contenu de la cellule est aligné en haut par rapport au reste de la ligne.

Les deux descripteurs `m{length}` et `b{length}` correspondent respectivement à une `\parbox{length}` et à une `\parbox[b]{length}` (voir section 6.1.4 pour `\parbox`).

un long texte	a	a
a	un long texte	a
a	a	un long texte

Ces descripteurs définissent donc comment le paragraphe doit être placé dans la cellule, par rapport aux textes des cellules se trouvant dans la même ligne.

 **Code**
array

```

\begin{tabular}{|p{1cm}|m{1cm}|b{1cm}|}
\hline
un long texte & a & a \\
\hline
a & un long texte & a \\
\hline
a & a & un long texte \\
\hline
\end{tabular>

```

Un autre nouveau descripteur disponible est `!{desc}`. Celui-ci est identique à `|` si ce n'est que `desc` est inséré au lieu d'un filet vertical. Il fait

la même chose que `@{desc}`, mais sans supprimer les blancs. Voyons cela sur un exemple :

A/A / A

On voit bien que les deux premiers A sont collés au /, tandis qu'entre les deux derniers, il y a de l'espace.

Code
array

```

\begin{tabular}{|l@{/}!{/}l|}
\hline
A & A & A \\
\hline
\end{tabular>

```

5.3.2 Insérer des commandes

Il est possible d'insérer une série de commandes avant ou après le contenu de toutes les cellules d'une colonne. On va respectivement utiliser les descripteurs `>{decl}` et `<{decl}` avant et après le descripteur de la colonne qu'on souhaite affecter.

Article	Prix €
Livres	12 €
DVD	17 €

Ici, on a mis les textes des cellules de la première colonne en gras et on a ajouté le symbole € derrière les textes des cellules de la seconde colonne. Notez que pour avoir le symbole €, il faut importer le package `eurosym`.

Code
array

```

\begin{tabular}{|>{\bf}l|c<{\euro}|}
\hline
Article & Prix \\
\hline
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular>

```

On peut par exemple utiliser ces descripteurs pour avoir une colonne dont toutes les cellules sont passées en mode mathématique. Pour cela, il suffit de déclarer la colonne comme suit :

```
>{\$}l<{\$}
```

Dans ce cas-ci, on pourrait également écrire :

```
>{\ensuremath}1
```

5.3.3 Définir ses propres descripteurs

Lorsqu'on souhaite personnaliser les cellules d'une colonne, on va utiliser les descripteurs `>{decl}` et `<{decl}`. Néanmoins, lorsqu'on doit le faire pour plusieurs tableaux d'un même document, il devient plus pratique de définir un nouveau descripteur. On va pour cela utiliser la commande `\newcolumntype` qui prend deux paramètres, qui sont un nom et la description de la colonne.

Voici comment on peut réécrire l'exemple précédent en définissant deux nouveaux descripteurs qui sont D et P.

Code
array

```

\newcolumntype{D}{>{\bf}1<{}}
\newcolumntype{P}{>{ }1<{-\euro{}}

\begin{tabular}{|D|P|}
\hline
Article & Prix \\
\hline
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular>

```

Notez qu'on peut également définir des descripteurs qui prennent des paramètres. On spécifie le nombre de paramètres en option et on utilise le i^e paramètre avec `#i` (voir chapitre 12 pour plus de détails).


```
\newcolumntype{M}[1]{>{ }1<{-#1}}
```

5.3.4 Alignement sur un point décimal

Parfois, lorsqu'on a des tableaux de données, on souhaite aligner les nombres d'une colonne sur le séparateur décimal. Le package `dcolumn` définit le descripteur D qui prend trois paramètres.

Article	Prix
Livres	9,5
DVD	17,99
Tondeuse	129,25

Le premier paramètre indique le caractère séparateur (“,” dans notre exemple). Le second paramètre indique le séparateur qu’il faut produire (également “,” dans notre exemple). Enfin, le dernier paramètre indique le nombre maximal de chiffres décimaux. Une valeur négative fait en sorte de centrer le séparateur dans la colonne.

 Code
dcolumn

```

\begin{tabular}{|l|D{,}{,}{2}|}
\hline
\bf Article & \bf Prix \\
\hline
Livres      & 9,5 \\
DVD         & 17,99 \\
Tondeuse    & 129,25 \\
\hline
\end{tabular>

```

5.4 Apparence


Dans cette section, on va voir différentes choses qu’il est possible de faire pour personnaliser l’apparence des tableaux. On verra d’abord comment modifier l’alignement du texte des cellules, ensuite on jouera sur le style des filets, puis sur les couleurs et enfin sur l’espacement.

5.4.1 Alignement dans les cellules

On définit l’alignement du texte dans les cellules grâce aux descripteurs de colonnes. On peut néanmoins vouloir modifier l’alignement pour certaines cellules. Pour ce faire, on utilise les commandes `\raggedleft`, `\raggedright` et `\centering`.

A	B	C
left	right	center

Le problème est que ces commandes redéfinissent `\` qu’on ne peut donc plus utiliser pour terminer la ligne du tableau. Il faut donc utiliser `\tabularnewline` à la place, pour toutes les lignes contenant une des trois commandes `\raggedleft`, `\raggedright` ou `\centering`.

 **Code**


```
\begin{tabular}{|*{3}{p{2cm}|}}
\hline
A & B & C \\
\hline
\raggedright left & \raggedleft right &
\centering center \tabularnewline
\hline
\end{tabular}
```

5.4.2 Épaisseur des filets

On a vu qu'on pouvait avoir deux filets horizontaux ou verticaux qui se suivent. Ils sont alors séparés par un certain espace, contrôlé par la commande de longueur `\doublerulesep` qu'on va pouvoir fixer à zéro.

	Année
Produit	

Vous pourriez vous demander pourquoi on ne change pas directement la valeur de `\arrayrulewidth`. On peut le faire, mais le résultat sera inesthétique.


 **Code**

```
\setlength{\doublerulesep}{\arrayrulewidth}
\begin{tabular}{|1|1|1|}
\hline
& Année \\
\hline\hline
Produit & \\
\hline
\end{tabular}
```

En ce qui concerne les filets verticaux, il suffit d'utiliser une barre verticale épaisse en tant que séparateur de colonne. On pourra ainsi avoir :

	 Année
Produit	

Remarquez qu'on a utilisé `\`, pour insérer une espace fine insécable. Sans cela, le texte aurait été complètement collé au filet vertical. Pour ce dernier, on a utilisé la commande $\text{T}_{\text{E}}\text{X}$ `\vrule` en spécifiant une certaine largeur.


 **Code**

```
\begin{tabular}{|l@{\,}\vrule width 1mm\,|l|}
\hline
& Année \\
\hline
Produit & \\
\hline
\end{tabular}
```

En ce qui concerne les filets horizontaux, on peut utiliser les commandes `\Xhline` et `\Xcline` du package `makecell` qui prennent l'épaisseur du trait en paramètre.

	Année
Produit	

Il y a une petite incompatibilité avec le package `arydshln` qu'on va voir juste après. Pour contrer cela, il faut insérer la commande `\ADLinactivate` avant l'environnement `tabular`.

 **Code** **makecell**

```
\begin{tabular}{|l|l|}
\hline
& Année \\
\Xhline{2.5pt}
Produit & \\
\hline
\end{tabular}
```


On verra également plus loin dans le chapitre une autre technique pour changer l'épaisseur des filets horizontaux.

5.4.3 Style des filets

On peut également vouloir changer le style des filets. Par exemple, on pourrait vouloir des filets pointillés. Pour cela, on va faire appel au package `arydshln`. Pour avoir un filet vertical pointillé, il suffit d'utiliser `:` et pour les filets horizontaux, on utilise les commandes `\hdashline` et `\cdashline` qui s'utilisent comme `\hline` et `\cline`.

Article	Prix
Livres	12
DVD	17

Vous voyez qu'on va bien entendu pouvoir combiner les filets « normaux » avec ceux en pointillés.


arydshln

Code

```


\begin{tabular}{|l:c|}
\hline
\bf Article & \bf Prix \\
\hdashline
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular>

```

Les longueurs des traits pleins et des zones vides sont contrôlées par les deux commandes de longueur `\dashlinedash` et `\dashlinegap`. On peut également personnaliser ces valeurs pour chaque trait en utilisant l'option des commandes `\hdashline` et `\cdashline` et en utilisant `;
{length/length}` pour les filets verticaux.

Article	Prix
Livres	12
DVD	17

La première longueur correspond au trait plein et la seconde à la longueur de la zone vide.


arydshln

Code

```

\begin{tabular}{|l;{5pt/2pt}c|}
\hline
\bf Article & \bf Prix \\
\hdashline[5pt/10pt]
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular>

```

Il est possible de paramétrer encore plus ces filets. Pour plus d'informations, je vous recommande la documentation du package `arydshln`.

Comme dit dans la section précédente, on peut localement désactiver et activer la gestion des filets pointillés grâce aux commandes `\ADLinactivate` et `\ADLactivate`. On utilise cela notamment pour résoudre les conflits qu'il pourrait y avoir avec d'autres packages.

5.4.4 Couleur de fond des cellules

Il est possible d'ajouter une couleur de fond à une cellule, ou à toutes les cellules d'une ligne ou d'une colonne. Pour cela, on va utiliser des commandes définies dans le package `colortbl`.

Article	Prix
Livres	12
DVD	17

Pour avoir une cellule avec couleur de fond personnalisée, il suffit d'utiliser la commande `\cellcolor` qui prend en paramètre le nom d'une couleur. On peut également utiliser les autres possibilités de définir la couleur, comme avec la commande `\color` du package du même nom. On peut par exemple écrire `\cellcolor[gray]{0.5}` ou alors `\cellcolor[rgb]{1,1,0.2}`.

Code
colortbl

```

\begin{tabular}{|l|c|}
\hline
\bf Article & \bf Prix \\
\hline
Livres & \cellcolor{red} 12 \\
DVD & 17 \\
\hline
\end{tabular>

```

Pour changer la couleur de fond de toutes les cellules d'une ligne, il faut utiliser la commande `\rowcolor` dans la première cellule de la ligne.

Article	Prix
Livres	12
DVD	17

Vous devez utiliser la commande dans la première cellule de la ligne, sans quoi vous ferez face à une erreur de compilation.

Code
colortbl

```

\begin{tabular}{|l|c|}
\hline
\rowcolor[gray]{0.7} \bf Article & \bf Prix \\
\hline
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular>

```

Pour changer la couleur de fond de toutes les cellules d'une colonne, il suffit d'utiliser le descripteur de colonne `>` avec la commande `\rowcolor`.

Article	Prix
Livres	12
DVD	17

On va ainsi affecter toutes les cellules d'une colonne. Si on veut par exemple avoir toutes les cellules d'une colonne avec un fond gris, mais pas la cellule de la première ligne, on met toutes les cellules de la colonne en gris, et puis, on met celle de la première ligne avec un fond blanc.

Code
colortbl

```


\begin{tabular}{|l|>{\cellcolor[gray]{0.7}}c|}
\hline
\bf Article & \cellcolor{white} \bf Prix \\
\hline
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular>

```

On peut également vouloir colorer les lignes d'un tableau avec deux couleurs différentes, une pour les lignes paires et l'autre pour les lignes impaires. Pour cela, on va utiliser la commande `\rowcolors` disponible en chargeant `\usepackage[table]{xcolor}`. Le premier paramètre indique le numéro de la première ligne qu'il faut colorer, ensuite les deux paramètres suivants donnent respectivement la couleur des lignes impaires et paires.

Article	Prix
Livres	12
DVD	17
Chaussettes	3
Vodka	15

La commande `\rowcolors` doit être utilisée en dehors de l'environnement `tabular`. Sachez aussi qu'on peut laisser les deux derniers paramètres vides. Cela signifie simplement qu'il ne faut pas de couleur de fond.



Code xcolor

```

\rowcolors{2}{lightgray}{gray}

\begin{tabular}{|l|c|}
\hline
\bf Article & \bf Prix \\
\hline
Livres      & 12 \\
DVD         & 17 \\
Chaussettes & 3 \\
Vodka      & 15 \\
\hline
\end{tabular}

```

La commande `\rowcolors` s'appliquera à tous les tableaux se trouvant entre le moment où vous avez écrit la commande et la fin du bloc la contenant.


Enfin, lorsque vous avez des tableaux avec des cellules qui ont été créées avec `\multicolumn` ou `\multirow`, des problèmes peuvent survenir. Vous devrez à chaque fois utiliser `\cellcolor` pour colorer ces cellules étendues.

5.4.5 Espacement

Afin de modifier l'espacement qui se trouve entre le texte d'une cellule et sa bordure, il faut redéfinir la commande `\arraystretch`.

Article	Prix
Livres	12
DVD	17

La valeur de la commande est un nombre qui représente le facteur d'agrandissement de l'espacement. Dans notre exemple, on a mis 1.5, c'est-à-dire que l'espacement vaut 150% de l'espacement normal.



Code

```

\renewcommand{\arraystretch}{1.5}


\begin{tabular}{|l|c|}
\hline
\bf Article & \bf Prix \\
\hline
Livres      & 12 \\
DVD         & 17 \\
\hline
\end{tabular}

```

Parfois, on ne souhaite modifier l'espacement que pour certaines cellules. Une solution pour cela consiste à insérer une ligne verticale de largeur nulle dans les cellules qu'il faut modifier, avec la commande `\rule`.

Article	Prix
Livres	12
DVD	17

Dans cet exemple, on a ajouté une ligne verticale de largeur nulle dans la première cellule. La ligne fait 1 cm de haut, et est descendue de 0,4 mm.



Code

```

\begin{tabular}{|l|c|}
\hline
\bf\rule[-0.4cm]{0mm}{1cm}Article & \bf Prix \\
\hline
Livres & 12 \\
DVD & 17 \\
\hline
\end{tabular}

```

5.5 Dimensions et positionnement


Dans cette section, on va tout d'abord voir comment modifier les dimensions d'un tableau. On va commencer par s'intéresser à fixer la largeur d'un tableau et puis on verra comment gérer les longs tableaux qui s'étendent sur plus d'une page. Ensuite, on verra comment gérer la position d'un tableau par rapport au texte l'entourant, et comment pivoter un tableau.

5.5.1 Largeur fixe

On peut vouloir fixer la largeur totale d'un tableau. Pour cela, on peut utiliser l'environnement `tabular*` qui prend en paramètre la largeur totale du tableau.

A	B
---	---

Il reste un problème avec l'environnement `tabular*`. Même si on contrôle la largeur du tableau, on n'a pas de contrôle sur les largeurs des colonnes. Afin d'avoir un tableau esthétique, il faut ajouter un séparateur de colonne élastique avec `!\extracolsep{\fill}`.


 **Code**

```
\begin{tabular*}{4cm}{|!{\extracolsep{\fill}}1|1|}
\hline
A & B \\
\hline
\end{tabular*}
```

Une meilleure solution consiste à utiliser l'environnement `tabularx` disponible dans le package de même nom. Les paramètres sont les mêmes que pour `tabular*`. La différence est qu'un nouveau descripteur de colonne est disponible : `X`. Les colonnes de type `X` vont automatiquement se partager de manière équilibrée tout l'espace horizontal restant.

A	B	C	D
---	---	---	---

Dans cet exemple, on a un tableau de huit centimètres de large et deux colonnes `1` et deux colonnes `X`. Les deux colonnes `X` vont se partager la largeur restante.

 **Code** **tabularx**

```
\begin{tabularx}{8cm}{|1|X|1|X|}
\hline
A & B & C & D \\
\hline
\end{tabularx}
```

5.5.2 Long tableau

Parfois, on doit insérer des tableaux qui ne tiennent pas sur une seule page. Deux solutions sont à envisager : il faut soit modifier la taille du tableau avec `\scalebox` par exemple, soit autoriser le tableau à s'étendre sur plusieurs pages en utilisant l'environnement `longtable` disponible dans le package de même nom. Voyons un exemple :

TABLE 5.2 – Liste des prix.

Article	Prix
Livres	12
DVD	17
Chaussettes	3
Vodka	15
Portable	93

TABLE 5.2 – Liste des prix (suite).

Article	Prix
Chocolat	4
GSM	31
Piles	15

Vous voyez donc que le tableau commence au bas de la page précédente et s'étend jusqu'au début de celle-ci. Pour insérer un tel tableau, il faut tout d'abord définir les parties d'entête et de bas de tableau qui se répètent sur toutes les pages ; on délimite ces parties avec les commandes `\endfirsthead`, `\endhead`, `\endfoot` et `\endlastfoot` qui définissent respectivement l'entête à avoir sur la première page, sur les autres pages, le bas à avoir sur toutes les pages et sur la dernière page.

De plus, comme un long tableau n'est pas un flottant, on ne peut le placer dans un environnement `table` et on utilise donc directement la commande `\caption` pour ajouter une légende.

 **Code**
longtable

```

\begin{longtable}[c]{|l|c|}
  \caption{Liste des prix.} \\
  \hline
  \bf Article & \bf Prix \\
  \hline
\endfirsthead
  \caption[]{{Liste des prix (suite).} \\
  \hline
  \bf Article & \bf Prix \\
  \hline
\endhead
  \hline
\endfoot
  Livres & 12 \\
  DVD & 17 \\
  Chaussettes & 3 \\
  Vodka & 15 \\
  Portable & 93 \\
  Chocolat & 4 \\
  GSM & 31 \\
  Piles & 15
\end{longtable}

```

Vous voyez également que l'environnement `longtable` admet une option qui indique l'alignement horizontal du tableau. On a ici mis `c` pour centrer le tableau. Pour plus d'informations sur cet environnement, nous vous renvoyons à la documentation du package `longtable`.

5.5.3 Hauteur du tableau


On peut spécifier comment le tableau est aligné verticalement par rapport au texte environnant en précisant une option à l'environnement `tabular`.

texte

 texte

 texte

Les trois valeurs possibles pour l'option sont `t`, `c` ou `b` pour aligner le tableau sur le haut, pour le centrer ou l'aligner sur le bas. La valeur par défaut est `c`.


Code

```

texte \begin{tabular}[t]{|l|}
      \hline A \ B \\\hline
      \end{tabular}
texte \begin{tabular}[c]{|l|}
      \hline A \ B \\\hline
      \end{tabular}
texte \begin{tabular}[b]{|l|}
      \hline A \ B \\\hline
      \end{tabular}
texte

```


Lorsque l'on aligne le tableau sur le haut ou le bas, le résultat obtenu peut vous paraître inesthétique. Grâce aux commandes `\firstline` et `\lastline` disponibles dans le package `array`, on va obtenir un résultat plus joli. On utilise ces commandes à la place de `\hline` pour le premier et dernier filet horizontal du tableau.

texte

 texte

 texte

On voit maintenant que c'est le contenu de la première ou dernière ligne qui est utilisé pour effectuer l'alignement.


Code

`array`

```

texte \begin{tabular}[t]{|l|}
      \firstline A \ B \\\hline
      \end{tabular}
texte \begin{tabular}[b]{|l|}
      \hline A \ B \\\lastline
      \end{tabular}
texte


```

5.5.4 Rotation d'un tableau

Les tableaux sont parfois trop larges pour tenir sur la largeur de la page et une solution consiste à effectuer une rotation de 90 degrés au tableau. Pour faire cela, on peut utiliser l'environnement `turn` du package `rotating`.

Y	Âge
1984	25
1982	27

L'environnement prend un angle en degrés en paramètre. Notez qu'on peut avoir le même résultat avec la commande `\rotatebox` du package `graphicx`.



Code
rotating

```

\begin{turn}{90}
  \begin{tabular}{|c|c|}
    \hline
    \bf Y & \bf Âge \\
    \hline
    1984 & 25 \\
    1982 & 27 \\
    \hline
  \end{tabular}
\end{turn>

```

Maintenant, si on désire ajouter une légende au tableau, soit on fait comme on a vu à la section 5.1.2, mais alors la légende ne sera pas pivotée avec le tableau, soit on utilise l'environnement `sidewaystable` du package `rotating`. Le tableau sera automatiquement tourné de 90 ou 270 degrés selon qu'il se trouve sur une page paire ou impaire, et il sera forcément placé seul sur une page.


Code
rotating

```

\begin{sidewaystable}
  \centering
  \begin{tabular}{|c|c|}
    \hline
    \bf Y & \bf Âge \\
    \hline
    1984 & 25 \\
    1982 & 27 \\
    \hline
  \end{tabular}
  \caption{Âges en fonction de l'année de naissance.}
\end{sidewaystable>

```


5.6 Mise en page avancée

Dans cette dernière section sur les tableaux, on va voir quelques mises en page avancées comme les sous-tableaux, la gestion fine des filets horizontaux ainsi que des tableaux plus complexes.

5.6.1 Sous-tableaux

On peut insérer des sous-tableaux, tout comme on avait des sous-figures à la section 4.2.6. Il suffit d'utiliser la commande `\subtable` du package `subfigure`.

(a) 2009	(b) 2010								
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px 10px;">120</td> <td style="padding: 2px 10px;">80</td> <td style="padding: 2px 10px;">60</td> <td style="padding: 2px 10px;">90</td> </tr> </table>	120	80	60	90	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px 10px;">110</td> <td style="padding: 2px 10px;">90</td> <td style="padding: 2px 10px;">65</td> <td style="padding: 2px 10px;">86</td> </tr> </table>	110	90	65	86
120	80	60	90						
110	90	65	86						

Table 5.3. Prix moyens saisonniers.

On insère donc chaque sous-tableau avec la commande `\subtable` qui prend la légende en paramètre.

Code
subfigure

```

\begin{table}[!h]
  \centering
  \subtable[2009]{
    \begin{tabular}{|*{4}{1}|}
      \hline 120 & 80 & 60 & 90 \\ \hline
    \end{tabular}
  }
  \subtable[2010]{
    \begin{tabular}{|*{4}{1}|}
      \hline 110 & 90 & 65 & 86 \\ \hline
    \end{tabular}
  }
  \caption{Prix moyens saisonniers.}
\end{table>


```

5.6.2 Style des filets horizontaux

Pour présenter un tableau joliment, il est conseillé de ne pas utiliser de filets verticaux, mais rien que des horizontaux. De plus, le premier et le dernier filet doivent être plus épais et écartés du texte. Afin de satisfaire cela, on peut utiliser le package `booktabs` qui définit les commandes `\toprule`, `\midrule`, `\bottomrule` et `\cmidrule`.

Article	Prix
Livres	12
DVD	17

On utilise donc `\toprule` et `\midrule` pour le premier et dernier filet horizontal. Pour tous les autres filets, on utilise `\midrule` ou `\cmidrule` pour avoir un filet partiel.


Code
booktabs

```

\begin{tabular}{lc}
\toprule
\bf Article & \bf Prix \\
\midrule
Livres      & 12 \\
DVD        & 17 \\
\bottomrule
\end{tabular>

```

5.6.3 Exemple d'un tableau complexe

		Année	Bilan	
			Recettes	Dépenses
Département	Informatique	2009	42 k€	36 k€
		2010	42 k€	34 k€
	Électricité	2009	130 k€	125 k€
		2010	120 k€	122 k€
	Mathématique	2009	37 k€	29 k€
		2010	36 k€	27 k€

Cet exemple de tableau vous permet d'apprendre deux choses. Tout d'abord, il est possible d'avoir des filets verticaux partiels en les supprimant pour certaines lignes. On fait cela en insérant un `\multicolumn` d'une seule cellule. C'est ce qu'on a fait pour les deux premières cellules des deux premières lignes.

Ensuite, lorsqu'on utilise les descripteurs `>{...}` et `<{...}`, et qu'on ne souhaite pas qu'ils s'appliquent sur toutes les lignes, il suffit d'utiliser le même truc et de placer la cellule dans un `\multicolumn`.

Code
multirow

```

\renewcommand{\arraystretch}{1.5}
\begin{tabular}{|l|l|c|*{2}{r}<{\;k\euro}}|}
%% Ligne 1
\cline{3-5}
\multicolumn{1}{l}{} & & \multirow{2}{1cm}{Année}
& \multicolumn{2}{c|}{Bilan} \\
%% Ligne 2
\cline{4-5}
\multicolumn{1}{l}{} & & & \multicolumn{1}{c}{Recettes}
& \multicolumn{1}{c|}{Dépenses} \\
%% Lignes 3 et 4
\hline
\multirow{6}{3mm}{\rotatebox{90}{Département\;}}
& \multirow{2}{2.5cm}{Informatique} & 2009 & 42 & 36 \\
& & 2010 & 42 & 34 \\
%% Lignes 5 et 6
\cline{2-3}
& \multirow{2}{2.5cm}{Électricité} & 2009 & 130 & 125 \\
& & 2010 & 120 & 122 \\
%% Lignes 7 et 8
\cline{2-3}
& \multirow{2}{2.5cm}{Mathématique} & 2009 & 37 & 29 \\
& & 2010 & 36 & 27 \\
\hline
\end{tabular>

```

5.6.4 Feuille de calcul

En utilisant l'environnement `spreadtab` du package de même nom, il est possible d'avoir des fonctionnalités de type tableur directement dans un document \LaTeX . Le paramètre de l'environnement correspond au type de tableau qu'il faut construire avec les données.

A	B	Total
12	15	27
21	17	38
TotA 33	TotB 32	65
MoyA 16.5	MoyB 16	

On identifie une cellule par une lettre pour la colonne (a, b ...) et un nombre pour la ligne. Les cellules qui commencent par @ contiennent du texte non-interprété. On peut mixer du texte et un calcul dans une cellule en mettant le calcul entre `:=`.

Code
spreadtab

```

\begin{spreadtab}{{\tabular}{|c|c|}}
\hline
@A      & @B      & @Total \\
\hline
12      & 15      & a2+b2 \\
21      & 17      & a3+b3 \\
\hline
TotA :={a2+a3} & TotB :={b2+b3} & a4+b4 \\
MoyA :={a4/2}  & MoyB :={b4/2}  & \\
\hline
\end{spreadtab>

```

Le package est très riche et nous vous conseillons de lire sa documentation pour en savoir plus.

5.6.5 Importer un fichier CSV

Pour facilement insérer un fichier `csv` dans un document \LaTeX , on peut utiliser le package `datatool` qui permet de gérer des bases de données. Ce package est très riche et nous n'allons pas le traiter en détail.

Soit le fichier `csv` suivant :

```
A,B,Somme
12,23,35
21,43,64
```

On va pouvoir facilement l'insérer dans un document et obtenir le tableau suivant :

A	B	Somme
12	23	35
21	43	64

Pour cela, il faut tout d'abord créer une nouvelle base de données en important le fichier `csv` avec la commande `\DTLloaddb`. Ensuite, on peut afficher les données sous forme d'un tableau avec la commande `\DTLdisplaydb`.

Code
datatool

```

\DTLloaddb{mydb}{data.csv}
\DTLdisplaydb{mydb}

```

CHAPITRE 6

Boite

Ce chapitre décrit comment utiliser les boites. On commencera par voir les boites basiques et leur utilité, ainsi que les boites avec une bordure. Ensuite, on verra comment insérer des boites plus riches, notamment avec de l'ombre ou avec des formes non-rectangulaires. On verra également comment effectuer des transformations et jouer avec les couleurs. Enfin, on terminera le chapitre en voyant quelques boites plus sophistiquées, comme celles utilisées pour présenter le code des exemples de ce livre.

6.1 Insérer une boite


On peut insérer une nouvelle boite avec les commandes `\mbox` et `\makebox`. La différence entre les deux commandes est que la première insère une boite dont la largeur dépend de son contenu, alors qu'on va pouvoir spécifier la largeur avec la seconde commande. Une boite ne sera jamais découpée et restera donc toujours sur la même ligne. On les utilise notamment pour empêcher L^AT_EX d'effectuer une césure.

```
Une boite
  Une belle boite
Une belle boite
  Une belle boite
Une    belle    boite
```

La commande `\makebox` prend une première option qui est la largeur de la boite. Si vous ne spécifiez rien, la commande se comporte comme `\mbox`. Il y a également une seconde option qui permet de préciser l'alignement horizontal du texte à l'intérieur de la boite.

Par défaut, le texte est centré. Les autres valeurs possibles sont :

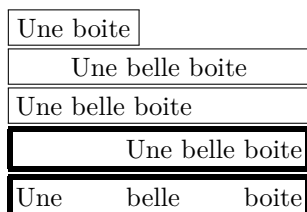
- l texte aligné à gauche;
- r texte aligné à droite;
- s texte justifié.


 **Code**

```
\mbox{Une boite} \\
\makebox[4cm]{Une belle boite} \\
\makebox[4cm][l]{Une belle boite} \\
\makebox[4cm][r]{Une belle boite} \\
\makebox[4cm][s]{Une belle boite}
```

6.1.1 Boite avec bordure

On peut également vouloir une bordure autour de la boite. Pour cela, il suffit d'utiliser les commandes `\fbox` et `\framebox` qui se comportent exactement comme les deux commandes `\mbox` et `\makebox` qu'on vient de voir. L'épaisseur de la bordure est contrôlée par la commande de longueur `\fboxrule`.



 **Code**

```
\fbox{Une boite} \\
\framebox[4cm]{Une belle boite} \\
\framebox[4cm][l]{Une belle boite}

\setlength{\fboxrule}{2pt}
\framebox[4cm][r]{Une belle boite} \\
\framebox[4cm][s]{Une belle boite}
```

6.1.2 Dimensionnement des boites

En \LaTeX , tout est représenté sous forme de boites. Celles-ci sont définies par trois longueurs qui sont leur largeur, hauteur et profondeur. Chaque

boite possède également un point de référence situé sur sa gauche. Les différentes boîtes des éléments qui constituent une ligne sont placées de manière à ce que leurs points de référence se situent tous sur une même ligne appelée la ligne de base. Tous ces éléments sont illustrés sur la figure 6.1.

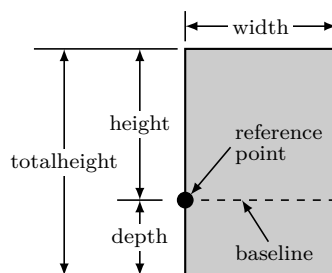


Figure 6.1. Une boîte \LaTeX .

Les valeurs de ces différentes longueurs peuvent être obtenues via les commandes `\width`, `\height`, `\depth` et `\totalheight`. Ainsi, on peut par exemple aisément faire une boîte dont la largeur est le triple de celle occupée par son contenu.

Une belle boîte

En utilisant le package `calc`, on peut même faire des calculs plus complexes. On verra d'autres exemples plus tard.

Code

```
\framebox[3\width]{Une belle boîte}
```

6.1.3 Espacement des boîtes

On peut modifier l'espace ajouté entre le contenu de la boîte et sa bordure grâce à la commande de longueur `\fboxsep`. Pour ne modifier que la hauteur de la boîte, on va utiliser la même technique que celle vue à la section 5.4.5, à savoir insérer une ligne verticale de largeur nulle.

Une belle boîte
Une belle boîte

**Code**

```
\fbox{\rule[-0.4cm]{0mm}{1cm}Une belle boite}
\setlength{\fboxsep}{3mm}
\fbox{Une belle boite}
```

6.1.4 Boite paragraphe

Enfin, dans les boites qu'on vient de voir, il n'est possible de mettre qu'une seule ligne de texte. Pour insérer un paragraphe dans une boite, il convient d'utiliser la commande `\parbox` dont le premier paramètre est la largeur de la boite et le second son contenu.

Une belle boite
contenant
beaucoup de texte

Comme montré sur cet exemple, on contrôle l'alignement du texte dans la boite avec les commandes d'alignement `\raggedleft`, `\raggedright` et `\centering`. Le contenu est justifié par défaut.

**Code**

```
\fbox{\parbox{3cm}{
\centering Une belle boite contenant beaucoup de texte
}}
```

On peut également contrôler l'alignement vertical de la boite par rapport à la ligne de base via une option de la commande. On peut aligner en haut (t) ou en bas (b), sachant que par défaut, la boite est centrée.

texte

A
B

 texte

A
B

 texte

A
B

L'alignement en haut se fait par rapport à la première ligne de texte de la boite et l'alignement en bas par rapport à la dernière ligne de texte.

**Code**

```
texte \fbox{\parbox[t]{3mm}{A\B}}
texte \fbox{\parbox{3mm}{A\B}}
texte \fbox{\parbox[b]{3mm}{A\B}}
```


6.2 Boites de transformation


On va maintenant voir toute une série de boites qui permettent d'effectuer des transformations sur leur contenu. On verra comment bouger une boite en hauteur, effectuer un redimensionnement ou une mise à l'échelle et faire un effet miroir.

6.2.1 Déplacement vertical

On peut déplacer une boite verticalement en utilisant la commande `\raisebox`. Celle-ci prend en premier paramètre la distance de laquelle il faut déplacer la boite et en second argument son contenu.

```
Bonjour, comment vas-tu ? Bien, merci !
```

Une valeur positive fait monter la boite tandis qu'une négative la fait descendre.



Code


```
Bonjour, \raisebox{3mm}{comment} vas-tu ?
Bien, \raisebox{-3mm}{merci} !
```

6.2.2 Mise à l'échelle et déformation

La commande `\scalebox` du package `graphicx` permet de faire une mise à l'échelle d'une boite. La commande prend en paramètre le facteur d'échelle à appliquer. Avec l'option, on peut spécifier un facteur différent pour la mise à l'échelle verticale.

```
Texte normal
agrandi 2 fois
rétréci 1/2 fois verticalement
```

Remarquez que l'option de la commande arrive après le premier paramètre. Le texte de la troisième ligne n'est pas redimensionné horizontalement, mais écrasé verticalement.



Code **graphicx**

```
Texte normal \\
\scalebox{2}{agrandi 2 fois} \\
\scalebox{1}[0.5]{rétréci 1/2 fois verticalement}
```


6.2.3 Redimensionnement

La commande `\resizebox` du package `graphicx` permet de redimensionner une boite en spécifiant en paramètre une nouvelle largeur et hauteur.

Texte

 texte
 texte

On peut utiliser un `!` à la place de la largeur ou hauteur lorsque l'on veut un redimensionnement qui conserve les proportions de la boite.


Code
graphicx

```

Texte \\
\resizebox{4mm}{10mm}{texte} \\
\resizebox{10mm}{4mm}{texte} \\
\resizebox{10mm}{!}{texte}


```

6.2.4 Rotation

La commande `\rotatebox` du package `graphicx` permet d'effectuer une rotation sur une boite. Le premier paramètre de la commande est un angle donné en degrés et le second est le contenu de la boite.

Tourné de 30 degrés
 et de -20 degrés

Un angle positif correspond à une rotation dans le sens anti-horloger et un angle négatif correspond à une rotation dans le sens horloger.


Code
graphicx

```

Tourné de \rotatebox{30}{30 degrés}
et de \rotatebox{-20}{-20 degrés}


```

6.2.5 Effet miroir

La commande `\reflectbox` du package `graphicx` permet d'effectuer un effet de miroir vertical. La commande prend en paramètre le contenu de la boite.

rioriM Miroir

Notez qu'on peut avoir exactement le même effet de miroir vertical avec la commande `\scalebox{-1}[1]{Miroir}`.


Code
graphicx


```
\reflectbox{Miroir} Miroir
```

6.2.6 Composition de transformation

On peut composer les transformations en utilisant plusieurs types de boîtes. Par exemple, on peut produire un effet de miroir horizontal en combinant trois transformations.

Miroir


La première chose à faire est d'appliquer un effet de miroir vertical, ensuite, on applique une rotation de 180 degrés et enfin, il faut remonter la boîte d'une fois sa hauteur (qu'on obtient avec `\depth`).


Code
graphicx


```
Miroir \\  

\raisebox{\depth}{\rotatebox{180}{\reflectbox{Miroir}}}
```

Lorsque vous composez des transformations, il faut bien faire attention à l'ordre dans lequel elles sont faites. Ceci peut changer le résultat obtenu comme le montre l'exemple suivant.

rioriM rioriM

Dans le premier exemple, on fait d'abord un miroir vertical suivi d'une rotation de 30 degrés. Dans le second exemple, on fait les transformations dans l'autre sens.


Code
graphicx

```
\rotatebox{30}{\reflectbox{Miroir}}  

\reflectbox{\rotatebox{30}{Miroir}}
```

6.3 Styles de boites

On va maintenant voir différents styles de cadre. On va commencer par voir des boites dont on peut spécifier le type de bordure. Ensuite, on verra des boites plus riches auxquelles on peut notamment ajouter un logo. Enfin, on terminera avec des boites de couleur.

6.3.1 Bordures

On peut obtenir différents types de bordures en chargeant le package `fancybox`. On peut définir des cadres avec ombre, des cadres avec trait double et des cadres avec coins arrondis. On utilise respectivement les commandes `\shadowbox`, `\doublebox`, `\ovalbox` et `\Ovalbox`.



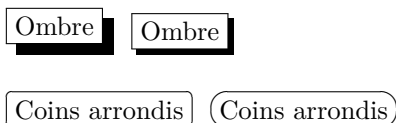
La différence entre les deux derniers cadres est que, avec `\Ovalbox`, la bordure est plus épaisse. Pour `\ovalbox`, elle est définie par `\thinlines` et pour `\Ovalbox`, elle est définie par `\thicklines`.

```

Code fancybox
\shadowbox{Ombre}
\doublebox{Double bordure}
\ovalbox{Ovale}
\Ovalbox{Ovale}

```

La largeur de l'ombre est contrôlée par la commande `\shadowsize` et les coins arrondis sont définis par la commande `\cornersize`.



```

Code fancybox
\shadowbox{Ombre}
\setlength{\shadowsize}{2pt}
\shadowbox{Ombre} \
\ovalbox{Coins arrondis}
\cornersize{2}
\ovalbox{Coins arrondis}


```

6.3.2 Boites de couleur

Le package `graphicx` propose la commande `\colorbox` qui permet d'avoir une boîte dont on peut spécifier la couleur de fond, et la commande `\fcolorbox`, qui permet en plus de spécifier la couleur de la bordure.

Une belle boîte Une belle boîte avec bordure Une boîte

Tout comme pour les couleurs, on peut soit spécifier un nom, soit un modèle et la valeur de la couleur pour le modèle choisi.


 **Code** graphicx, color

```
\colorbox{lightgray}{Une belle boîte}
\fcolorbox{black}{lightgray}{Une belle boîte avec bordure}
\fcolorbox[gray]{0.1}{0.9}{Une boîte}
```

Si on désire avoir une boîte de couleur contenant des paragraphes, il suffit de combiner une `\fcolorbox` avec une `\parbox` par exemple.

Bla
Bla
Bla

On peut également combiner la `\fcolorbox` avec une `minipage` pour obtenir le même résultat, en fonction de la situation et du contenu que vous voulez placer dans la boîte.

 **Code** graphicx, color

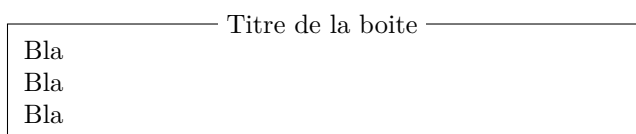
```
\fcolorbox{black}{lightgray}{
  \parbox{5cm}{
    Bla \\
    Bla \\
    Bla
  }
}
```

6.3.3 Superposition de boites

Avec ce qu'on a vu jusqu'à présent dans ce chapitre, on peut déjà faire beaucoup de constructions assez intéressantes. On va maintenant voir comment faire une boîte avec une bordure et un titre se situant par dessus cette dernière. Pour cela, on va utiliser la commande `\boxput` qui permet de superposer deux boîtes.

La commande prend deux boites en paramètres et place la première boite derrière la seconde (ou inversement si vous utilisez la commande `\boxput*`). Enfin, on doit spécifier une coordonnée qui indique où la première boite est placée par rapport à la seconde. Par exemple, avec $(0,0)$ elle est centrée, avec $(0,1)$ elle est centrée et en haut, avec $(-1,-1)$ elle est en bas à gauche ...

Voici par exemple comment on peut construire une boite avec un cadre et un titre :



On place donc une boite avec le titre et un fond blanc, obtenue avec `\colorbox`, par dessus une boite avec un cadre, obtenue avec `\fbox`.

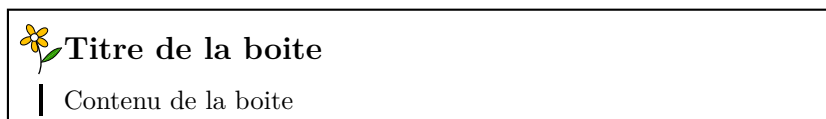
Code
fancybox

```


\boxput*(0,1){
  \colorbox{white}{Titre de la boite}
}{
  \setlength{\fboxsep}{6pt}
  \fbox{\begin{minipage}{8cm}
    Bla \\
    Bla \\
    Bla
  \end{minipage}}
}
```

6.3.4 Boite riche avec logo

Pour obtenir des boites riches, avec la possibilité d'avoir un logo, comme celles utilisées pour présenter les extraits de code dans ce livre, on utilise l'environnement `bclogo` défini dans le package de même nom. Dans la version de base, il suffit de spécifier le titre de la boite en paramètre. Voici ce que ça donne :



On a donc une boite avec un fond blanc, une bordure noire et un logo qui est une fleur. Notez que la boite est dessinée soit avec `PSTricks`, soit avec `PGF/TikZ` et qu'il faut faire son choix lorsque l'on importe le package.



Code bclogo

```
\begin{bclogo}{Titre de la boite}
  Contenu de la boite
\end{bclogo}
```

On peut personnaliser la boite en spécifiant des options à l'environnement. Pour plus d'informations, consultez la documentation du package. Pour information, les boites de ce livre sont obtenues avec les options suivantes :

```
[couleur=black!10,couleurBord=black!50,arrondi=0.1,logo=\bccrayon]
```


6.4 Grande boite

On peut vouloir insérer des boites larges pouvant contenir autre chose que simplement un mot ou une ligne de texte. Pour ce faire, il suffit de placer le contenu de la boite dans une `\parbox` comme on en a déjà vu auparavant. On peut également utiliser l'environnement `minipage`.

Paragraphe 1

Paragraphe 2

En utilisant cela, on peut redéfinir différents paramètres comme l'espace entre paragraphes par exemple.



Code

```
\fbox{\begin{minipage}{8cm}\setlength{\parskip}{5mm}
  Paragraphe 1

  Paragraphe 2
\end{minipage}}
```

6.4.1 Boite sur plusieurs pages


Toutes les boites qu'on a vues jusqu'à présent ne pouvaient pas s'étendre sur plusieurs pages. Pour pouvoir faire cela, on va devoir utiliser le package `framed`.

Le package `framed` propose quatre environnements :

- `framed`

- shaded
- snugshade
- leftbar

La boîte ci-dessus s'étend donc automatiquement sur deux pages. Les environnements `shaded` et `snugshade` permettent d'avoir une boîte avec une ombre et `leftbar` permet d'avoir un trait vertical dans la marge de gauche.

 **Code** **framed**

```
\begin{framed}
  Le package framed propose quatre environnements :
  \begin{itemize}
    \item framed
    \item shaded
    \item snugshade
    \item leftbar
  \end{itemize}
\end{framed}
```

Le package propose également des commandes pour définir de nouveaux types de boîtes s'étendant sur plusieurs pages. Nous vous recommandons de lire sa documentation.

CHAPITRE 7

Mathématique

Ce chapitre décrit tous les éléments de base qu'il vous faudra maîtriser pour utiliser des mathématiques dans un document. On va commencer avec les symboles et les constructions de base. Ensuite, on enchaînera avec l'insertion d'équations et de matrices. Enfin, on terminera en voyant comment faire de la mise en page comme des théorèmes par exemple.

7.1 Mode mathématique

Il y a trois modes mathématique différents. On peut insérer une formule dans le texte en utilisant `$... $` ou `\(... \)`. Pour insérer une formule séparée du texte et centrée, on peut utiliser `$$... $$` ou `\[... \]`. Enfin, pour insérer une équation numérotée, on va utiliser l'environnement `equation`.


Lorsque que l'on est en mode mathématique et qu'on souhaite repasser en mode texte, il suffit d'utiliser la commande `\text{rm}`. Dans ce mode, les espaces seront à nouveau pris en compte, contrairement à ce qui se passe en mode mathématique.

Soient x et y deux entiers. Leur somme vaut :

$$z = x + y$$

On a également la propriété suivante :

$$z^2 \leq x^2 + y^2 \tag{7.1}$$

 **Code**

```

Soient  $x$  et  $y$  deux entiers. Leur somme vaut :
\[
z = x + y
\]
On a également la propriété suivante :
\begin{equation}
z^2 \leq x^2 + y^2
\end{equation}

```

7.2 Symboles

Cette section rassemble les symboles les plus courants dont vous pourriez avoir besoin. La plupart sont disponibles directement en \LaTeX , tandis que pour d'autres, il faudra utiliser \mathcal{AMS} et donc importer les packages `amsmath` et `amssymb`.

7.2.1 Lettres grecques

Les lettres grecques minuscules sont obtenues avec des commandes, sauf pour la lettre *omicron* qui est obtenue avec le *o* minuscule romain.

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

Les lettres grecques majuscules sont obtenues soit avec une lettre majuscule romaine, soit avec une commande.

A	<code>A</code>	H	<code>H</code>	N	<code>N</code>	T	<code>T</code>
B	<code>B</code>	Θ	<code>\Theta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>
Γ	<code>\Gamma</code>	I	<code>I</code>	O	<code>O</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	K	<code>K</code>	Π	<code>\Pi</code>	X	<code>X</code>
E	<code>E</code>	Λ	<code>\Lambda</code>	P	<code>P</code>	Ψ	<code>\Psi</code>
Z	<code>Z</code>	M	<code>M</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>

Enfin, certaines lettres minuscules possèdent une variante. Pour la lettre *kappa*, il faut importer le package `amssymb`.

ε	<code>\varepsilon</code>	\varkappa	<code>\varkappa</code>	ϱ	<code>\varrho</code>	φ	<code>\varphi</code>
ϑ	<code>\vartheta</code>	ϖ	<code>\varpi</code>	ς	<code>\varsigma</code>		

7.2.2 Symboles divers

\aleph	<code>\aleph</code>	$'$	<code>\prime</code>	\forall	<code>\forall</code>	<code>\forall</code>	<code>\forall</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\exists	<code>\exists</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\neg	<code>\neg</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\flat	<code>\flat</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\natural	<code>\natural</code>
\wp	<code>\wp</code>	\perp	<code>\perp</code>	\sharp	<code>\sharp</code>	\sharp	<code>\sharp</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\clubsuit	<code>\clubsuit</code>	\clubsuit	<code>\clubsuit</code>
\Im	<code>\Im</code>	\sphericalangle	<code>\sphericalangle</code>	\diamond	<code>\diamond</code>	\diamond	<code>\diamond</code>
∂	<code>\partial</code>	\triangle	<code>\triangle</code>	\heartsuit	<code>\heartsuit</code>	\heartsuit	<code>\heartsuit</code>
∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\spadesuit	<code>\spadesuit</code>	\spadesuit	<code>\spadesuit</code>
$<$	<code><</code>	$>$	<code>></code>	\gg	<code>\gg</code>	\gg	<code>\gg</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\ll	<code>\ll</code>	\ll	<code>\ll</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	$=$	<code>=</code>	$=$	<code>=</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\neq	<code>\neq</code>	\neq	<code>\neq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\equiv	<code>\equiv</code>	\equiv	<code>\equiv</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\approx	<code>\approx</code>	\approx	<code>\approx</code>
\sqsubset^*	<code>\sqsubset^*</code>	\sqsupset^*	<code>\sqsupset^*</code>	\cong	<code>\cong</code>	\cong	<code>\cong</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\propto	<code>\propto</code>	\propto	<code>\propto</code>
\triangleleft	<code>\triangleleft</code>	\triangleright	<code>\triangleright</code>	\bowtie	<code>\bowtie</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\notin	<code>\notin</code>	\notin	<code>\notin</code>
\vee	<code>\vee</code>	\wedge	<code>\wedge</code>	\sim	<code>\sim</code>	\sim	<code>\sim</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>	\models	<code>\models</code>
\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>	\asymp	<code>\asymp</code>
$+$	<code>+</code>	$-$	<code>-</code>	\simeq	<code>\simeq</code>	\simeq	<code>\simeq</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\circ	<code>\circ</code>	\circ	<code>\circ</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\bullet	<code>\bullet</code>	\bullet	<code>\bullet</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\oplus	<code>\oplus</code>	\oplus	<code>\oplus</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\ominus	<code>\ominus</code>	\ominus	<code>\ominus</code>
\otimes	<code>\otimes</code>	\oslash	<code>\oslash</code>	\odot	<code>\odot</code>	\odot	<code>\odot</code>
\dagger	<code>\dagger</code>	\ddagger	<code>\ddagger</code>	\Rightarrow	<code>\Rightarrow</code>	\Rightarrow	<code>\Rightarrow</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	$*$	<code>\ast</code>	$*$	<code>\ast</code>
\diamond	<code>\diamond</code>	\wr	<code>\wr</code>	\star	<code>\star</code>	\star	<code>\star</code>
\lessdot^*	<code>\lessdot^*</code>	\gtrdot^*	<code>\gtrdot^*</code>	\doteq	<code>\doteq</code>	\doteq	<code>\doteq</code>
\bigtriangleup	<code>\bigtriangleup</code>	\bigtriangledown	<code>\bigtriangledown</code>	\bigcirc	<code>\bigcirc</code>	\bigcirc	<code>\bigcirc</code>
\uplus	<code>\uplus</code>	\amalg	<code>\amalg</code>	\Join^*	<code>\Join^*</code>	\Join^*	<code>\Join^*</code>
\lhd	<code>\lhd</code>	\rhd	<code>\rhd</code>				
\unlhd	<code>\unlhd</code>	\unrhd	<code>\unrhd</code>				

7.2.3 Flèches

On peut obtenir toute une série de flèches en mode mathématique. Voyons tout d'abord les flèches classiques :

\rightarrow	<code>\rightarrow</code>	\leftarrow	<code>\leftarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Leftarrow	<code>\Leftarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\longrightarrow	<code>\longrightarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Longrightarrow	<code>\Longrightarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\downarrow	<code>\downarrow</code>	\uparrow	<code>\uparrow</code>	\updownarrow	<code>\updownarrow</code>
\Downarrow	<code>\Downarrow</code>	\Uparrow	<code>\Uparrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\leadsto	<code>\leadsto*</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>		
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>		
\nwarrow	<code>\nwarrow</code>	\swarrow	<code>\swarrow</code>		
\rightharpoonup	<code>\rightharpoonup</code>	\leftharpoonup	<code>\leftharpoonup</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\rightharpoondown	<code>\rightharpoondown</code>	\leftharpoondown	<code>\leftharpoondown</code>		

Pour obtenir des flèches barrées, il suffit d'utiliser la commande `\not`. Par exemple :

\nrightarrow `\not\rightarrow` \nleftarrow `\not\leftarrow`

Cependant, pour avoir des flèches barrées plus esthétiques dans certaines situations, vous pouvez utiliser des commandes du package \mathcal{AMS} :

\nrightarrow `\nrightarrow` \nleftarrow `\nleftarrow` \nleftrightarrow `\nleftrightarrow`
 \nRightarrow `\nRightarrow` \nLeftarrow `\nLeftarrow` \nLeftrightarrow `\nLeftrightarrow`

7.2.4 Raccourcis

De nombreux symboles peuvent être obtenus avec plusieurs commandes, afin de faciliter leurs utilisations dans certaines situations.

		(équivalent à)
\rightarrow	<code>\to</code>	<code>\rightarrow</code>
\leftarrow	<code>\gets</code>	<code>\leftarrow</code>
\ni	<code>\owns</code>	<code>\ni</code>
\wedge	<code>\land</code>	<code>\wedge</code>
\vee	<code>\lor</code>	<code>\vee</code>
\neg	<code>\lnot</code>	<code>\neg</code>
$ $	<code>\vert</code>	<code> </code>
$\ $	<code>\Vert</code>	<code>\ </code>
\implies	<code>\implies</code>	<code>;\Longrightarrow\;</code>
\iff	<code>\iff</code>	<code>;\Longleftrightarrow\;</code>

7.2.5 Polices de caractères

On peut obtenir différentes polices de caractères en mode mathématique. Les lettres représentant les ensembles sont obtenues avec la commande `\mathbb` :

\mathbb{A} `\mathbb{A}` \mathbb{B} `\mathbb{B}` \mathbb{C} `\mathbb{C}` ...

On peut également obtenir des lettres rondes (calligraphiées) avec la commande `\mathcal` :

\mathcal{A} `\mathcal{A}` \mathcal{B} `\mathcal{B}` \mathcal{C} `\mathcal{C}` ...

Enfin, on peut également avoir une police de type gothique avec la commande `\mathfrak` :

\mathfrak{A} `\mathfrak{A}` \mathfrak{B} `\mathfrak{B}` \mathfrak{C} `\mathfrak{C}` ...

7.3 Constructions basiques

On va voir toutes les constructions mathématiques basiques dans cette section : fraction, racine carrée, fonctions, indice et exposant ...

Indice	x_{10}	<code>x_{10}</code>
Exposant	x^{10}	<code>x^{10}</code>
Fraction	$\frac{x}{y}$	<code>\frac{x}{y}</code>
Racine nième	$\sqrt[n]{x}$	<code>\sqrt[n]{x}</code>
Trait horizontal	\overline{AB}	<code>\overline{AB}</code>
	\underline{AB}	<code>\underline{AB}</code>
Vecteur	\overrightarrow{AB}	<code>\overrightarrow{AB}</code>
Angle	\widehat{AB}	<code>\widehat{AB}</code>
Tilde	\widetilde{AB}	<code>\widetilde{AB}</code>

7.3.1 Accolade

On peut avoir une accolade qui s'étend en-dessous ou au-dessus d'une formule, tout en ajoutant un texte sur l'accolade avec les commandes `\overbrace` et `\underbrace`.

$$S = \overbrace{\pi}^{3,14} \cdot \underbrace{r}_{\text{rayon}}^2$$



Code

```
\[
  S = \overbrace{\pi}^{3,14} \cdot \underbrace{r}_{\text{rayon}}^2
\]
```

7.3.2 Opérateurs de taille variable avec indice et exposant

On peut insérer des opérateurs de taille variable, pouvant prendre une valeur en indice et en exposant. Voyons tout d'abord comment insérer une somme :

$$\sum_{i=0}^{n-1} x_i^2 + 2$$

On utilise pour cela la commande `\sum` :




Code

```
\[
  \sum_{i = 0}^{n - 1} {x_i^2 + 2}
\]
```

On peut également obtenir le même effet en utilisant les commandes `\limits` et `\nolimits` qui permettent de spécifier les valeurs en exposant et indice à placer sur le symbole obtenu avec la commande qui précède.

$$\int_0^2 x^2 - 1 dx \quad \int_0^2 \limits x^2 - 1 dx \quad \int_0^2 \nolimits x^2 - 1 dx$$

Vous remarquerez qu'avec `\limits`, les valeurs sont positionnées au-dessus et en-dessous du symbole alors qu'avec `\nolimits`, ces dernières sont placées à côté du symbole.



Code

```
\[
\int_{0}^{2} {x^2 - 1 \, dx}
\qquad \int\limits_{0}^{2} {x^2 - 1 \, dx}
\qquad \int\nolimits_{0}^{2} {x^2 - 1 \, dx}
\]
```


Voici les opérateurs les plus courants :

\prod	<code>\prod</code>	\coprod	<code>\coprod</code>	\sum	<code>\sum</code>	\int	<code>\int</code>
\bigcap	<code>\bigcap</code>	\bigcup	<code>\bigcup</code>	\bigsqcup	<code>\bigsqcup</code>	\oint	<code>\oint</code>
\bigwedge	<code>\bigwedge</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>	\iint	<code>\iint</code>
\bigoplus	<code>\bigoplus</code>	\bigotimes	<code>\bigotimes</code>	\bigodot	<code>\bigodot</code>	\iiint	<code>\iiint</code>

7.3.3 Fonctions

Il y a toute une série de fonctions prédéfinies qui sont disponibles via des commandes. Celles-ci sont rendues dans une police de caractère droite, et suivies d'un espace. On peut par exemple écrire :

$$p = \Pr \left[|X| \leq \sin \frac{\pi}{4} \right]$$



Code

```
\[
p = \Pr\left[ |X| \leq \sin \frac{\pi}{4} \right]
\]
```

Voici les fonctions prédéfinies :

<code>\arccos</code>	<code>\cot</code>	<code>\exp</code>	<code>\lim</code>	<code>\min</code>	<code>\tan</code>
<code>\arcsin</code>	<code>\coth</code>	<code>\gcd</code>	<code>\liminf</code>	<code>\Pr</code>	<code>\tanh</code>
<code>\arctan</code>	<code>\csc</code>	<code>\hom</code>	<code>\limsup</code>	<code>\sec</code>	
<code>\arg</code>	<code>\deg</code>	<code>\inf</code>	<code>\ln</code>	<code>\sin</code>	
<code>\cos</code>	<code>\det</code>	<code>\ker</code>	<code>\log</code>	<code>\sinh</code>	
<code>\cosh</code>	<code>\dim</code>	<code>\lg</code>	<code>\max</code>	<code>\sup</code>	

On peut utiliser ces fonctions avec `\limits` et `\nolimits` :

$$\lim_{x \rightarrow \infty} \frac{1}{x} \quad \max_{x \in [0,5]} x^2$$

**Code**

```
\[
  \lim\nolimits_{x \to \infty} \frac{1}{x}
  \quad \max_{x \in [0,5]} x^2
\]
```

Il y a également la fonction `mod` utilisée pour l'arithmétique modulaire. On peut l'obtenir avec deux commandes différentes : `\bmod` et `\pmod`.

$$a \bmod b \quad x \equiv a \pmod{b}$$

**Code**

```
\[
  a \bmod b
  \quad x \equiv a \pmod{b}
\]
```

Définir une fonction

Pour définir une nouvelle fonction, il faut utiliser la commande `\DeclareMathOperator` du package `amsmath`. Si on souhaite redéfinir une fonction qui existait déjà, on va utiliser la commande `\operatorname`. Ces définitions doivent se faire dans le préambule du document.

$$\operatorname{acos} x = \operatorname{Arc} \cos x$$

**Code****amsmath**

```
\DeclareMathOperator{\acos}{acos}
\renewcommand{\arccos}{\operatorname{Arc-cos}}

\begin{document}
$\operatorname{acos} x = \operatorname{arccos} x$
```

7.3.4 Superposer deux symboles

On peut superposer deux symboles avec la commande `\stackrel`. Celle-ci prend deux paramètres. Le premier est le symbole qui va au-dessus du second.

$$P(f) \stackrel{\Delta}{=} \{x \mid f(x) \geq 0\}$$

Voici les délimiteurs les plus courants :

(())	↑	\uparrow	⇧	\Uparrow
[[]]	↓	\downarrow	⇩	\Downarrow
{	\{	}	\}	↕	\updownarrow	⇕	\Updownarrow
<	\langle	>	\rangle				\
⌊	\lfloor	⌋	\rfloor		\lceil		\rceil
/	/	\	\backslash				

7.4.1 Définition directe

En utilisant le package `delarray`, on va pouvoir définir directement les délimiteurs. Il faut par contre faire très attention aux conflits avec d'autres packages, dont notamment `dcolumn`, `colortbl`, `makecell` ou `tabularx`.

$$\begin{pmatrix} y & 3x^2 + x - 1 \\ 2y & -x + 3 \end{pmatrix}$$

Code
delarray

```
\[
  \begin{array}({cc})
    y & 3 x^2 + x - 1 \\
    2 y & -x + 3
  \end{array}
\]
```


7.5 Tableaux, matrices et équations

Dans cette section, on va voir comment définir des tableaux en mode mathématique. On verra également comment définir des matrices, des systèmes d'équations et des listes d'équations.

7.5.1 Tableaux

On peut définir des tableaux en mode mathématique en utilisant l'environnement `array`. Ce dernier fonctionne comme l'environnement `tabular` (section 5.1).

$$\begin{array}{l} 1 = 1 \\ 11 = 11 \\ 111 = 111 \end{array}$$

 **Code**

```


\l
  \begin{array}{rc1}
    1 & = & 1 \\
    11 & = & 11 \\
    111 & = & 111
  \end{array}
\l

```

7.5.2 Matrices et systèmes d'équations

On peut facilement construire des matrices et systèmes d'équations en utilisant l'environnement `array` et les délimiteurs. Pour une matrice, il suffit d'utiliser des parenthèses à gauche et à droite. Pour un système d'équations, on place une accolade à gauche et aucun délimiteur à droite.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{cases} a = b \\ c = d \end{cases}$$

 **Code**

```

\l
  \left(\begin{array}{11}
    a & b \\
    c & d
  \end{array}\right)
  \quad
  \left\{\begin{array}{11}
    a = b \\
    c = d
  \end{array}\right.
\l

```


7.5.3 Listes d'équations

Parfois, on souhaite faire une liste d'équations successives, de la forme $Y = X$. Pour ce faire, on peut utiliser l'environnement `eqnarray`.

$$y = x^2 - 1 \tag{7.2}$$

$$2y = -x^2 + x + 1 \tag{7.3}$$

Cet environnement ne doit pas être utilisé en mode mathématique, c'est implicite. Il correspond en fait à l'environnement `array` avec l'alignement `rc1`. De plus, chaque équation est automatiquement numérotée.

 **Code**


```
\begin{eqnarray}
y & = & x^2 - 1 \\
2y & = & -x^2 + x + 1
\end{eqnarray}
```

Pour ne pas avoir de numérotation sur les équations, il faut plutôt utiliser l'environnement `eqnarray*`. On peut également supprimer les numéros de certaines des lignes en utilisant la commande `\nonumber`.

$$y = x^2 - 1 \tag{7.4}$$

$$2y = -x^2 + x + 1$$

$$y = -x + 4 \tag{7.5}$$

 **Code**


```
\begin{eqnarray}
y & = & x^2 - 1 \\
2y & = & -x^2 + x + 1 \nonumber \\
y & = & -x + 4
\end{eqnarray}
```

7.6 Théorèmes

On peut très facilement insérer des nouveaux théorèmes (ou des définitions, propositions, lemmes ...) en définissant des nouveaux environnements. Pour cela, il faut utiliser la commande `\newtheorem` au début du document. Celle-ci prend deux paramètres qui sont le nom de l'environnement et l'étiquette à afficher. En option, on peut spécifier le niveau de sectionnement pour la numérotation et on peut spécifier un compteur pour la numérotation. Voici un exemple d'un tel théorème :

Théorème 7.1 (Pythagore). *Le carré de la longueur de l'hypothénuse d'un triangle rectangle est égal à la somme des carrés des longueurs des autres côtés.*

On a défini ici un environnement `theorem` dont les numéros dépendront du chapitre. De plus, on a donné un nom au théorème via l'option de l'environnement. Enfin, la numérotation des propositions se fera avec le même compteur que celui utilisé pour les théorèmes.

 **Code**

```

\newtheorem{theorem}{Théorème}[chapter]
\newtheorem{proposition}[theorem]{Proposition}

\begin{theorem}[Pythagore]
Le carré de la longueur de l'hypothénuse d'un triangle rectangle
est égal à la somme des carrés des longueurs des autres côtés.
\end{theorem}

```

7.6.1 Démonstration


On peut également insérer une démonstration en utilisant l'environnement `proof` disponible dans le package `amsthm`. Le symbole QED¹ est automatiquement ajouté grâce à cet environnement.

Démonstration. Soient x et y deux entiers tels que $x^2 + y^2 \geq 0$. De plus, soit l'entier positif z défini comme $\frac{x}{y}$. On peut écrire :

$$zy = x \tag{7.6}$$

□

Parfois, le symbole QED n'est pas placé comme il faut. C'est notamment le cas lorsque la preuve se termine par une équation. Dans ce cas, on peut utiliser la commande `\qedhere` pour indiquer où l'on souhaite le placer.

 **Code**

```

\begin{proof}
Soient  $x$  et  $y$  deux entiers tels que  $x^2 + y^2 \geq 0$ . De
plus, soit l'entier positif  $z$  défini comme  $\frac{x}{y}$ .
On peut écrire :

\begin{equation}
zy = x \qedhere
\end{equation}
\end{proof}

```

Notez qu'on peut également indiquer en option à l'environnement `proof` le nom du théorème qui est prouvé.

1. Quod erat demonstrandum, synonyme de CQFD.

CHAPITRE 8

Autour du texte

Dans ce chapitre, on va voir toute une série de constructions qu'il est possible de faire autour du texte. On va commencer avec les références croisées. On verra ensuite les entêtes et pieds de page et les notes de marge et de bas de page. On terminera par quelques autres constructions.


8.1 Références croisées

Les références croisées permettent de poser des étiquettes à certains endroits (une figure, un tableau, une équation ...) et puis d'y faire référence automatiquement. L'étiquette est définie avec la commande `\label`. On y fait ensuite référence avec `\ref`, ou avec `\pageref` pour faire référence à la page contenant cet élément.

$$y = ax^2 + bx + c \quad (8.1)$$

L'équation 8.1 présentée à la section 8.1 en page 117.

Dans cet exemple, on fait référence à une équation, à une section et au numéro de la page sur laquelle l'équation se trouve.

 **Code**

```
\section{Référence croisée\label{sec:crossref}}  
  
\begin{equation}  
  y = ax^2 + bx + c  
  \label{eq:2nd-deg-polynom}  
\end{equation}  
  
\it L'équation-\ref{eq:2nd-deg-polynom} présentée à la  
  section-\ref{sec:crossref} en page-\pageref{eq:2nd-deg-polynom}.
```

Remarquez comment on a nommé les étiquettes en préfixant celles-ci avec une abréviation indiquant de quel type d'élément il s'agit : `eq:` pour les équations, `sec:` pour les sections ...

Notez que pour les sections, on a placé le `\label` dans le paramètre de `\section`. On peut également le placer juste après la commande `\section`. De plus, on fait chaque fois précéder les commandes `\ref` et `\pageref` d'un tilde (`~`). Ceci permet d'éviter une césure entre le mot et le numéro.

8.1.1 Figures et tables

Pour les figures et les tables, la commande `\label` doit être utilisée après la commande `\caption`, sans quoi vous n'aurez pas le bon numéro puisque c'est `\caption` qui le génère.

A

B


Table 8.1. Données.

Table 8.2. Données 2.

Comme vous pouvez le voir dans le tableau 8.1

Il est également possible d'utiliser la commande `\label` dans le paramètre de la commande `\caption`. Ainsi, on aurait pu écrire :

```
\caption{Données\label{tab:data}}
```



Code

```


\begin{table}[!h]
  \centering
  \begin{tabular}{|l|}\hline A\\\hline\end{tabular}
  \caption{Données.}
  \label{tab:data}
\end{table}
% [...]
\it Comme vous pouvez le voir dans le tableau~\ref{tab:data}

```

8.1.2 Référence proche

Assez souvent, lorsque l'on inclut une référence, on fait également référence à la page. On va par exemple écrire : « Le tableau 8.1 de la page 118 ». Par contre, si l'élément référencé se situe sur la même page, on ne doit pas la préciser. De même, lorsque la page est proche, on préférerait « à la page suivante » ou « sur la page ci-contre » ...

Pour cela, on peut utiliser la commande `\vref` du package `varioref`. On obtient ainsi une référence comme : « Le tableau 8.1 page ci-contre ». Il ne faut pas oublier de sélectionner l'option `french` en important le package. De plus, les étiquettes ne peuvent pas comporter n'importe quel caractère. Les deux-points (:) sont notamment interdits.


 **Code** **varioref**

On obtient ainsi une référence comme :

```
\og Le tableau~\vref{tab.data} \fg.
```

La package propose également une commande `\vpageref` qui est l'équivalent de `\pageref`. Enfin, il y a également les commandes `\vrefrange` et `\vpagerefrange` qui permettent de référencer plusieurs éléments successifs. On va ainsi par exemple pouvoir écrire :

Voir les tableaux 8.1 à 8.2 page précédente.

 **Code** **varioref**

```
\it Voir les tableaux~\vrefrange{tab.data}{tab.data2}.
```

8.1.3 Référence intelligente

Le package `cleveref` propose également des nouvelles commandes permettant de gérer des références proches, mais il offre des fonctionnalités complémentaires par rapport à `varioref`. On peut afficher plusieurs références avec la commande `\cref` et celles-ci seront triées intelligemment. La commande `\crefrange` permet d'afficher un intervalle de références.

```

1.1 Historique

[LIGNE DU TEMPS]
FIGURE 1 – Historique.

1.2 Description
1.3 Progression et invasion

[CARTE DU MONDE]
FIGURE 2 – Pays envahis.

La section 1 ...
Les sections 1.1 à 1.3 ...
Comme montré aux sections 1, 1.2 et 1.3 et figures 1 et 2 ...

```

Code
cleveref

```

\section{Les abeilles tueuses\label{killer-bees}}
\subsection{Historique\label{history}}
\begin{figure}[!ht]
  \centering
  [LIGNE DU TEMPS]
  \caption{Historique.}
  \label{time-line}
\end{figure}
\subsection{Description\label{description}}
\subsection{Progression et invasion\label{invasion}}
\begin{figure}[!ht]
  \centering
  [CARTE DU MONDE]
  \caption{Pays envahis.}
  \label{invaded-countries}
\end{figure}

La \section-\ref{killer-bees} \dots \\\
Les \crefrange{history}{invasion} \dots \\\
Comme montré aux \cref{killer-bees,time-line,description,
                        invasion,invaded-countries} \dots

```

8.1.4 Texte de la référence

Assez souvent, lorsque l'on utilise la commande `\ref` pour avoir un numéro, on le fait précéder d'un texte. Par exemple, on veut pouvoir écrire « à la section 8.1.4 ». La commande `\labelformat` du package `varioref` permet d'éviter de chaque fois écrire la `section~\ref{sec:reftext}`.

Code
varioref

```

\labelformat{subsection}{la section~#1}
\labelformat{table}{le tableau~#1}

% [...]

\Ref{sec:reftext} traite de \dots{}
Un résumé est donné dans~\ref{tab:data}.

```

La commande `\Ref` ajoute une majuscule à la première lettre du texte. Le résultat obtenu est donc :

« La section 8.1.4 traite de ... Un résumé est donné dans le tableau 8.1. ».

8.1.5 Sous-figures et sous-tables

On a vu aux sections 4.2.6 et 5.6.1 qu'on pouvait définir des sous-figures ou sous-tables. On peut référencer une sous-figure en particulier si on le souhaite. On utilise pour cela la commande `\subref` du package `subfigure`.

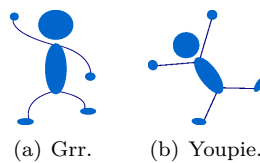


Figure 8.1. Humeurs.

La figure 8.1 présente deux humeurs. La colère est illustrée par le dessin 8.1(a). On peut également voir la joie sur la sous-figure (b).

Code
subfigure

```

\begin{figure}[!h]
  \centering
  \subfigure[Grr.]{
    \label{moods:angry}
    \includegraphics[height=1.5cm]{images/angry.png}
  }
  \quad
  \subfigure[Youpie.]{
    \label{moods:happy}
    \includegraphics[height=1.5cm]{images/happy.png}
  }
  \caption{Humeurs\label{fig:moods}.}
\end{figure}


```

La `figure-\ref{fig:moods}` présente deux humeurs. La colère est illustrée par le dessin-`\ref{moods:angry}`. On peut également voir la joie sur la sous-`figure-\subref{moods:happy}`.

8.1.6 Compteur de pages

Si vous souhaitez connaître le numéro de la page courante, il suffit d'utiliser la commande `\thepage`. Pour connaître le numéro de la dernière page d'un document, on peut utiliser le package `lastpage` qui va ajouter une étiquette `LastPage` sur la dernière page. On obtient ainsi le numéro de cette dernière avec la commande `\pageref{LastPage}`.

Vous êtes à la page 121 sur un total de 201 pages.

 **Code**
lastpage

Vous êtes à la page-`\thepage{}` sur un total de `\pageref{LastPage}`-pages.

Vous remarquerez néanmoins que, dans le cas d'un document avec plusieurs parties et différents styles de numérotation, le nombre total de page n'indique pas le nombre total de pages du document.

8.2 Légendes

Pour insérer la légende d'une figure ou d'un tableau, on utilise la commande `\caption`. On peut également utiliser la commande `\captionof` du package `caption` dans certaines situations (voir section 4.1.3).

8.2.1 Style des légendes


La manière la plus commode pour modifier le style des légendes consiste à utiliser les commandes du package `caption`. Pour rappel, une légende est composée d'un nom (figure, table ...), d'un numéro, d'un séparateur (un point par défaut) et enfin d'un titre. On va pouvoir modifier le style de ces différents éléments.

A

Figure 1 – La lettre A.

Les commandes `\captionnamefont` et `\captiontitlefont` permettent de définir le style du nom et du titre de la légende. La commande `\captiondelim` définit le séparateur entre le nom et le titre. Enfin, les commandes `\precaption` et `\postcaption` indiquent du code qui sera ajouté avant et après celui qui produit la légende.

Le style de la légende présentée ci-dessus a été défini avec les commandes suivantes, placées dans le préambule.

 **Code** ccaption

```

\captiondelim{ -- }
\captionnamefont{\small\sf\bfseries}
\captiontitlefont{\small\sf}
\precaption{\rule{\linewidth}{0.4pt}\}

```


8.2.2 Nom des légendes

On peut vouloir changer le nom des légendes, et par exemple remplacer « *Figure* » par « *Fig.* ». Pour cela, il suffit de redéfinir les commandes `\figurename`, `\tablename` ...

A

Fig. 1 – La lettre A.

Lorsque l'on utilise l'extension `babel`, il faut redéfinir les noms pour la bonne langue en faisant les appels dans `\addto\captionsfrench` dans le préambule du document.

 **Code** babel

```

\addto\captionsfrench{
  \renewcommand{\figurename}{Fig.}
  \renewcommand{\tablename}{Tab.}
}

```

8.3 Entête et pied de page


On a vu à la section 2.3.1 qu'il était possible d'avoir différents types de styles pour un document. Le style `headings` permet d'insérer le titre du chapitre courant et le numéro de page dans l'entête, tout en laissant le pied de page vide. On va voir comment on peut modifier l'entête et le pied de page.

8.3.1 Marque

En utilisant le style de document `myheadings`, on va pouvoir définir les éléments qui apparaissent dans l'entête de la page. Tout d'abord, le

numéro de page sera placé dans l'entête, à droite. On va pouvoir définir ce qu'on place à gauche dans l'entête. Cette marque est contrôlée par la commande `\rightmark` (marque des pages de droite). Dans le cas des livres, on peut définir une marque différente pour les pages de gauche. Elle sera contrôlée par `\leftmark`.

On ne peut pas modifier ces commandes soi-même. On doit le faire via `\markright`, qui permet de définir la marque des pages de droite, et via `\markboth`, qui permet de définir les marques des pages gauche et droite. Pour ce faire, on aimerait connaître le titre du chapitre ou de la section courante. On va devoir redéfinir les commandes `\chaptermark` et `\sectionmark` qui sont appelées à chaque nouveau chapitre et chaque nouvelle section. On peut utiliser les commandes `\thepage`, `\thechapter` et `\thesection` qui contiennent les numéros de la page, du chapitre et de la section courante.

 **Code**

```
\begin{document}
  \pagestyle{myheadings}
  \renewcommand{\sectionmark}[1]{%
    \markboth{\today}{\bf\thesection}~#1}}


  \section{Introduction}
  % ...
```

Voici un exemple de ce qui est obtenu avec la classe de document `book`. On voit bien qu'on a le titre de la section courante sur les pages impaires et la date sur les pages paires.

	<p>0.1 <i>Introduction</i> 1</p> <p>0.1 Introduction</p> <p>Hello</p>
<p>2</p> <p>World!</p> <p>26 juin 2010</p>	<p>0.2 <i>Conclusion</i> 3</p> <p>0.2 Conclusion</p> <p>Bye bye World!</p>

8.3.2 Définir l'entête et le pied de page

En utilisant le style `fancy` disponible en ajoutant le package `fancyhdr`, on va pouvoir modifier beaucoup plus facilement l'entête et le pied de page. On va pouvoir spécifier le texte à mettre à gauche, au centre et à droite de l'entête et du pied de page avec les commandes `\lhead`, `\chead`, `\rhead`, `\lfoot`, `\cfoot` et `\rfoot`. On peut également avoir un trait horizontal dont les longueurs sont représentées par les commandes `\headrulewidth` et `\footrulewidth`.

 **Code**
fancyhdr

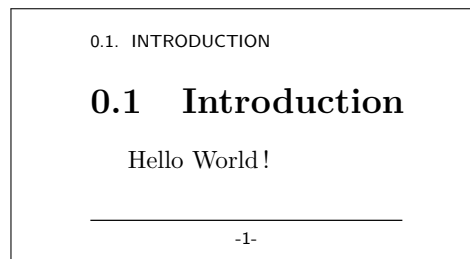
```

\pagestyle{fancy}

\lhead{\scriptsize\sf\rightmark}
\chead{}
\rhead{}
\lfoot{}
\cfoot{\scriptsize\sf-\thepage-}
\rfoot{}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0.5pt}

\begin{document}
% ...


```



On doit redéfinir toutes les parties de l'entête et du pied de page, pour vider le contenu par défaut. On peut le faire plus rapidement en écrivant :

```
\fancyhead{}
\fancyfoot{}
```

Ces deux commandes permettent de définir les entêtes et pieds de page de manière plus générale. On va utiliser leur option en spécifiant deux lettres. La première parmi `L` (gauche), `C` (centre) et `R` (droite) et la seconde parmi `E` (page paire) et `O` (page impaire). On pourrait redéfinir l'exemple précédent avec ces commandes.

 **Code** fancyhdr

```

\pagestyle{fancy}

\fancyhead{} % on efface l'entête
\fancyhead[LO,RE]{\scriptsize\sf\rightmark}
\fancyfoot{} % on efface le pied de page
\fancyfoot[C]{\scriptsize\sf-\thepage-}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0.5pt}


\begin{document}
% ...

```

La seule différence est que la section courante sera placée à gauche sur les pages impaires et à droite sur les pages paires. Vous pouvez également utiliser la commande `\fancyhf`, qui est encore plus générale. Il faut en plus préciser H (entête) ou F (pied de page) en option.

8.3.3 Pages spéciales

Certaines pages, notamment la première page des chapitres, sont automatiquement en style `plain`. On peut modifier ce comportement en redéfinissant le style `plain` avec la commande `\fancypagestyle` du package `fancyhdr` (section 8.3.2).

 **Code** fancyhdr

```

\fancypagestyle{plain}{%
  \fancyhf{} % on efface tout
  \fancyfoot[C]{\scriptsize\sf-\thepage-}
}

```


8.3.4 Style personnalisé

On peut complètement modifier l'apparence des entêtes et pied de page. Il suffit pour cela de combiner l'utilisation du package `fancyhdr` (section 8.3.2) avec la redéfinition des marques (section 8.3.1). On peut par exemple obtenir le résultat suivant :

1-INTRODUCTION

1 Introduction

Hello World!



Code fancyhdr


```

\pagestyle{fancy}
\renewcommand{\sectionmark}[1]{%
  \markright{\bf\thesection}--\textsc{#1}}
\fancyhf{}
\head{\scriptsize\sf\rightmark}

```

8.4 Notes de bas de page

On va maintenant voir comment ajouter des notes de bas de page dans un document. La manière la plus simple est d'utiliser la commande `\footnote` qui insère une note de bas de page¹.



Code

```

... insère une note de bas de page\footnote{Le paramètre de la
  commande est le texte de la note de bas de page.}.


```

8.4.1 Plusieurs références vers la même note

Pour faire plusieurs références vers une même note, il faut procéder en deux étapes. On définit tout d'abord le texte de la note avec `\footnotetext` et ensuite, on insère les marques avec `\footnotemark`.

Voici une note référencée deux fois : ici² et là².

Vous remarquerez qu'on peut définir soi-même les numéros de notes avec l'option des commandes. Attention cependant au fait que lorsque l'on utilise `\footnotetext`, le compteur des notes n'est pas augmenté.



Code

```


\footnotetext[2]{Une seule et unique note.} Voici une note
référéncée deux fois : ici\footnotemark[2] et là\footnotemark[2].

```

1. Le paramètre de la commande est le texte de la note de bas de page.
2. Une seule et unique note.

8.4.2 Style des notes

On peut modifier le type de numérotation des notes en redéfinissant la commande `\thefootnote`. Le compteur associé aux notes de bas de page est `footnote`^b.

 **Code**

```
\renewcommand{\thefootnote}{\alph{footnote}}

... notes de bas de page est footnote\footnote{Le compteur de
      cette note est une lettre romaine minuscule.}.
```

Les différents styles possibles sont `\arabic`, `\roman`, `\Roman`, `\alph`, `\Alph` et `\fnsymbol`. La dernière commande permet d'avoir des symboles comme compteur ; ils sont au nombre de 9 : *, †, ‡, §, ¶, ||, **, †† et ‡‡. Ce style ne peut pas être utilisé si vous avez plus de 9 notes de bas de page.

On peut modifier beaucoup d'autres aspects du style des notes de bas de page en important le package `footmisc`. Beaucoup de choses sont activées via des options du package.

- `para` : chaque note de bas de page va dans son propre paragraphe
- `side` : les notes sont placées dans la marge plutôt qu'en bas de document
- `bottom` : force les notes de bas de page à être poussées tout en bas de la page
- `norule` : supprime le trait au-dessus des notes de bas de page
- `multiple` : permet de gérer les multiples références vers des notes

8.4.3 Trait horizontal

Le trait horizontal qui sépare les notes de bas de page du corps du texte est contrôlé par la commande `\footnoterule`. On peut supprimer le trait en la redéfinissant :

```
\renewcommand{\footnoterule}{}.
```


Une autre solution consiste à importer le package `footmisc` en spécifiant l'option `norule`.

b. Le compteur de cette note est une lettre romaine minuscule.

8.4.4 Niveaux de notes de bas de page

On peut définir différents niveaux de notes de bas de page en utilisant le package `manyfoot`. On définit un nouveau niveau avec `\DeclareNewFootnote`³. On doit spécifier un suffixe et on peut donner un style de numérotation en option^A.

De plus, on peut définir un trait horizontal de séparation entre les différents niveaux de notes. Pour cela, on va utiliser la commande `\newfootnoterule` du package `nccrules` pour définir le trait et la commande `\SelectFootnoteRule` du package `manyfoot` pour appliquer ce trait.



manyfoot, nccrules

Code

```

\newfootnoterule{dotted}{\mboxfill [2\width] [s]{.}}
\SelectFootnoteRule{dotted}
\DeclareNewFootnote{sec}[Alph]

\begin{document}
... de numérotation en option\footnotesecc{Note de second niveau,
numérotation en lettres romaines majuscules.}.

```

La commande `\newfootnoterule` prend un nom en premier paramètre et la définition du trait en second paramètre. On a utilisé la commande `\mboxfill` pour remplir une boîte avec des points. On sélectionne ensuite ce type de trait avec la commande `\SelectFootnoteRule`.

Enfin, on définit un nouveau niveau de notes avec `\DeclareNewFootnote`. Ce niveau est appelé `sec` et le compteur est en lettres romaines majuscules. Une fois le nouveau niveau défini, on peut utiliser les commandes `\footnotesecc`, `\footnotemarksec` et `\footnotetextsec`.

Avec le package `nccrules`, on peut définir des traits complexes comme ceux vus à la section 2.1.3. Il suffit d'utiliser `\newfootnotedashrule`. On peut par exemple définir :

```
\newfootnotedashrule{dotdashed}{4 2 1 2 4 -4}{.6}
```

8.4.5 Notes de bas de page en fin de document


Parfois, notamment pour la rédaction d'articles, on souhaite pouvoir insérer toutes les notes de bas de page dans le corps du texte, toutes

3. Cette commande prend un paramètre.

.....

A. Note de second niveau, numérotation en lettres romaines majuscules.

rassemblées à la fin par exemple. Pour ce faire, il va falloir utiliser le package `endnotes`. Celui-ci définit les commandes `\endnote`, `\endnotemark` et `\endnotetext`. On insère ensuite les notes avec `\theendnotes`.

 **Code** **endnotes**

```
\section{L'histoire}
Et c'est là\endnote{à ce moment là.} qu'elle a dit qu'elle ne le
  \endnote{son frère.} voulait plus avec elle\endnote{son ex-
  copine.}.

\theendnotes
```

Remarquez que vous pouvez toujours utiliser les notes de bas de page classiques tout en utilisant les `\endnote`.

1 L'histoire

Et c'est là¹ qu'elle a dit qu'elle ne le² voulait plus avec elle³.


Notes

¹à ce moment là.
²son frère.
³son ex-copine.

1

8.5 Notes de marge

On peut facilement insérer une note dans la marge d'un document en utilisant la commande `\marginpar`. La note est placée dans la marge de droite. Pour les documents recto-verso, les notes sont placées dans la marge extérieure. La largeur de la marge est contrôlée par la longueur `\marginparwidth`. Avec l'option de la commande, on peut placer un élément dans la marge opposée.

 **Code** **bbding**

```
... utilisant la commande \marginpar\marginpar{\PencilLeftDown}.
```

8.5.1 Notes « todo »

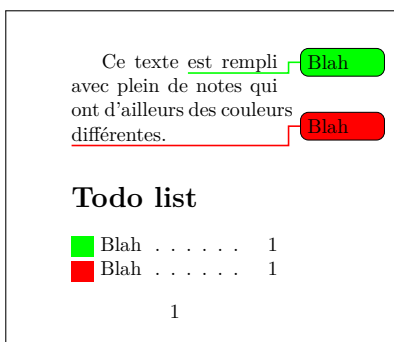
En utilisant le package `todonotes`, on va pouvoir insérer des notes de type « todo » dans la marge du document. On utilise pour cela la commande `\todo` qui ajoute le todo dans la marge et le relie avec l'endroit dans le texte où la commande a été utilisée.

blah

Code
todonotes

```
... cela la commande \todo \todo{blah} qui ajoute le todo ...
```

De nombreuses options sont disponibles pour modifier le style des notes, la couleur des boîtes, etc. On peut également insérer une liste des todos avec la commande `\listoftodos`. Voici un exemple plus riche :



On aurait pu changer la langue du titre et éviter « Todo list » en précisant l'option `french` en chargeant le package `todonotes`.

Code
todonotes

```
\usepackage[colorinlistoftodos]{todonotes}
\begin{document}
  Ce texte \todo[color=green]{Blah} est rempli avec plein de
  notes qui ont d'ailleurs des couleurs \todo[color=red]{
  Blah} différentes.
  \listoftodos
\end{document}
```

8.5.2 Notes depuis des flottants


La commande `\marginpar` ne fonctionne pas dans les flottants, ni dans les notes de bas de page. Pour cela, il faut utiliser la commande `\marginnote`

définie dans le package de même nom. Il y a néanmoins quelques soucis de positionnement avec ce package.


8.6 Modification

Un des avantages de \LaTeX est la possibilité de travailler à plusieurs sur un même document. Il est parfois utile d'indiquer les changements qui ont eu lieu entre plusieurs versions d'un même document. Le package `changebar` propose des environnements et commandes pour cela.

On peut placer les barres dans la marge gauche ou droite en spécifiant l'option `leftbars` ou `rightbars` en chargeant le package. La commande `\cbstart` indique le début d'un changement et `\cbend` la fin de celui-ci. La commande `\cbdelete` indique qu'un texte a été supprimé. Enfin, on peut également placer un texte changé dans l'environnement `changebar`.


 Ce texte a été ajouté
 Ici, quelque chose a été supprimé .
 Texte inséré

On peut changer la couleur des barres avec la commande `\cbcolor`. Pour cela il faut ajouter l'option `color` en chargeant le package.



Code

changebar

```

\cbstart Ce texte a été ajouté \cbend
Ici, quelque chose a été supprimé \cbdelete.
\cbcolor{red} \begin{changebar}
Texte inséré
\end{changebar>

```

CHAPITRE 9

Codes source

Dans ce chapitre, on va voir comment faire pour insérer du texte brut dans un document. On utilise notamment cela pour insérer des codes source, des sorties de programme, ou toute autre texte qui ne doit pas être interprété comme du code \LaTeX . On verra également comment présenter joliment des algorithmes.


9.1 Insertion brute

Pour insérer du texte brut dans un document, on peut soit utiliser la commande `\verb`, soit l'environnement `verbatim`. La commande `\verb` est particulière en ce sens qu'on peut également l'utiliser avec des délimiteurs autres que des accolades ouvrantes et fermantes.

On obtient le logo \LaTeX avec la commande `\LaTeX`

La commande `\verb` permet d'insérer du texte brut

Dans cet exemple, on a utilisé le caractère `=` pour délimiter le paramètre de la commande `\verb`.




Code

```
On obtient le logo \LaTeX{} avec la commande \verb=\LaTeX=  
\begin{verbatim}  
  La commande \verb permet d'insérer du texte brut  
\end{verbatim}
```

Notez que la commande `\verb` ne peut pas être utilisée dans un paramètre d'une autre commande.

9.1.1 Raccourci d'insertion brute

Cela peut vite devenir ennuyeux d'écrire `\verb` à chaque fois. Grâce au package `shortvrb` et sa commande `\MakeShortVerb`, il est possible de définir un caractère de telle sorte que tout texte entouré par ce caractère sera en mode brut.



Code


```
\MakeShortVerb{$}

On obtient le logo \LaTeX{} avec la commande $\LaTeX$
```

9.1.2 Conservation des tabulations

L'environnement `verbatimtab` du package `moreverb` permet d'insérer du texte brut, mais en préservant les tabulations. L'option de l'environnement permet de spécifier la largeur des tabulations, en nombre d'espaces.

```
if (x > 10)
  print "Examen réussi"
```



Code `moreverb`


```
\begin{verbatimtab}[3]
if (x > 10)
  print "Examen réussi"
\end{verbatimtab}
```

9.1.3 Ajout des numéros de ligne

Enfin, on peut ajouter les numéros des lignes en utilisant l'environnement `listing` du package `moreverb`. L'option de l'environnement indique l'écart entre les numéros, et son paramètre la première ligne qu'il faut numéroter.

```
1  for (i = 0 to 10)
2    if (i mod 2 = 0)
3      print i
```

Ici, pour modifier la largeur des tabulations, il faut manuellement redéfinir la commande `\verbatimtabsize`.

 **Code** moreverb

```

\renewcommand{\verbatimtabsize}{3}
\begin{listing}[1]{1}
for (i = 0 to 10)
  if (i mod 2 = 0)
    print i
\end{listing}

```


L'environnement `listingcont` permet d'insérer un texte brut avec numérotation, cette dernière continuant celle du dernier listing inséré.

```

4     else
5     print i + 1

```

Cet environnement est très pratique lorsque vous devez présenter un programme par morceaux, séparés par des commentaires explicatifs.

 **Code** moreverb

```

\begin{listingcont}
  else
    print i + 1
\end{listingcont}

```

9.1.4 Insertion depuis un fichier

On peut directement inclure un texte brut depuis un fichier texte. Pour cela, on va utiliser la commande `\verbatiminput` du package `verbatim`. Cette commande prend le chemin du fichier en paramètre.


Voici le contenu du fichier `data.csv` :

```

A,B,Somme
12,23,35
21,43,64

```

Vous pouvez également utiliser la commande `\verbatimtabinput` du package `moreverb` pour insérer un texte brut en préservant les tabulations. Pour avoir les numéros de ligne, vous pouvez utiliser `\listinginput`.

 **Code** verbatim

```

Voici le contenu du fichier \texttt{data.csv} :
\verbatiminput{data.csv}

```


9.1.5 Présenter un exemple

On peut présenter des exemples de code \LaTeX avec le résultat produit en utilisant l'environnement `SideBySideExample` du package `fvr-b-ex`.

\LaTeX , c'est fun :-)

1 `\LaTeX{}`, c'est fun :-)

Il faut définir la place disponible pour le résultat et le code avec l'option `xrightmargin`. On peut ensuite ajouter une bordure et la numérotation avec les options `frame` et `numbers`.

 **Code** **fvr-b-ex**


```
\begin{SideBySideExample}
  [xrightmargin=0.5\linewidth,frame=single,numbers=left]
  \LaTeX{}
```

9.2 Codes source d'un programme

Pour insérer le code source d'un programme, on peut utiliser le package `listings`. Ce dernier est très riche et offre de nombreuses options dont la coloration syntaxique. On utilise l'environnement `lstlisting` pour insérer un code source.

```
for (int i = 0; i < 5; i++)
{
    System.out.println ("i : " + i);
}
```

Comme on va le voir tout au long de cette section, ce package est très riche. Nous n'allons néanmoins pas tout explorer en détail, nous vous renvoyons à la documentation pour en savoir plus.

 **Code** **listings**

```
\begin{lstlisting}[language=java]
for (int i = 0; i < 5; i++)
{
    System.out.println ("i : " + i);
}
\end{lstlisting}
```

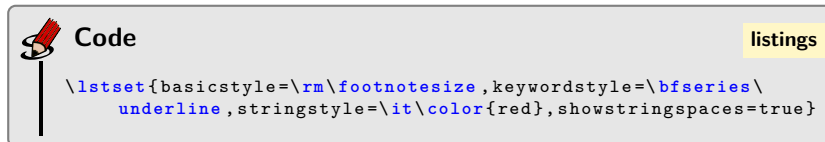
En plus de l'environnement `lstlisting`, on peut également utiliser la commande `\lstinline` qui fonctionne comme la commande `\verb`, en ce sens qu'on peut utiliser n'importe quel caractère comme délimiteur. Enfin, la commande `\lstinputlisting` permet de charger un code source depuis un fichier externe.

Il y a deux manières de préciser des options : soit via l'option de l'environnement ou de la commande, soit en utilisant la commande `\lstset` qui permet de définir des options de manière globale.

9.2.1 Style du texte

On peut modifier le style du texte en précisant la police, la couleur, la taille, etc. grâce aux options `basicstyle` (style de base), `keywordstyle` (mot réservé), `identifiestyle` (identificateur), `commentstyle` (commentaire) et `stringstyle` (chaîne de caractères). L'option `showstringspaces` permet de rendre les espaces dans les chaînes de caractères visibles.


```
for (int i = 0; i < 5; i++)
{
    System.out.println ("i:u" + i);
}
```



9.2.2 Numérotation

La numérotation est contrôlée par l'option `numbers` qu'on définit à `left`, `right` ou `none` (valeur par défaut). Le style des numéros est fixé par `numberstyle`, et le premier numéro est fixé avec `firstnumber` qui est soit un entier, soit `last` pour reprendre la numérotation. Enfin, `numbersep` représente l'écart entre les numéros affichés et `numberfirstline` indique s'il faut ou non numéroter la première ligne.

```
11 for (int i = 0; i < 5; i++)
12 {
    System.out.println ("i : " + i);
14 }
```


listings

Code

```
\lstset{numbers=left,numberstyle=\tiny\bfseries\underline,
stepnumber=2,firstnumber=11,numberfirstline=true}
```

9.2.3 Coloration syntaxique

On peut choisir le langage de programmation du code source présenté afin d'avoir la coloration syntaxique automatique. On utilise pour cela l'option `language`. Pour certains langages, il faut spécifier le dialecte en option. Par exemple, pour un programme en `C#`, on va écrire :

```
\lstset{language={ [Sharp]C}}
```


Notez qu'il faut englober le tout entre des accolades afin d'éviter des problèmes de compilation avec l'extension `keyval`.

9.2.4 Positionnement, dimensions et marge

On peut faire en sorte que le listing soit un flottant avec l'option `float`. On peut contrôler les marges avant et après le listing avec les options `aboveskip` et `belowskip`. Les options `xleftmargin` et `xrightmargin` spécifient les marges à gauche et à droite du listing. Enfin, l'option `linewidth` définit la largeur du listing.

On a ajouté une bordure dans l'exemple suivant, afin que vous puissiez voir les marges. On verra plus loin dans cette section comment faire cela.

```
for (int i = 0; i < 5; i++)
{
    System.out.println ("i : " + i);
}
```


listings

Code

```
\lstset{linewidth=9cm,xleftmargin=2cm,aboveskip=5mm,belowskip=1
cm,float=!h}
```


9.2.5 Cadre et couleur

On peut ajouter un cadre autour du listing avec l'option `frame`. On va utiliser les lettres `tblr` pour avoir une bordure en haut, en bas, à gauche et à droite. Les mêmes lettres en majuscule donneront une double bordure. Avec `rulesep`, on contrôle la distance entre les traits des bordures doubles. L'option `framesep` donne la distance entre le code et la bordure tandis que `framerule` donne l'épaisseur de cette dernière. On peut avoir des coins arrondis avec `frameround` qui prend quatre lettres (une par coin) en paramètre, ces lettres étant `t` ou `f` selon qu'on veuille ou non arrondir le coin.

On peut également jouer avec de la couleur. L'option `backgroundcolor` définit une couleur de fond. Les options `rulecolor`, `fillcolor` et `rulesepcolor` définissent la couleur de la bordure, du remplissage et de l'écart entre les doubles bordures.

```
for (int i = 0; i < 5; i++)
{
    System.out.println ("i : " + i);
}
```

Notez que certaines options ne sont pas compatibles. Il n'est par exemple pas possible d'utiliser `framerule` si `frameround` a été défini.

 **Code**
listings

```
\lstset{frame=tBlR,rulesep=1mm,framesep=5mm,framerule=2pt,
xrightmargin=5mm,xleftmargin=5mm,rulecolor={\color[gray]
}{0.6}},rulesepcolor={\color[gray]{0.9}}
```


9.2.6 Légende, étiquette et liste des listings

On peut ajouter une légende avec l'option `caption` et une étiquette pour faire référence au listing avec `label`. Enfin, il est possible d'insérer la liste des listings d'un document avec la commande `\lstlistoflistings`.

```
for (int i = 0; i < 5; i++)
{
    System.out.println ("i : " + i);
}
```

Listing 9.1. Une boucle for en Java.

La légende est par défaut placée au dessus. On peut modifier cette position avec l'option `captionpos` qui vaut `b` (bas) ou `t` (haut).


Code
listings


```
\lstset{frame=lines,caption={Une boucle for en Java.},label=lst:
  java_for_loop,captionpos=b}
```

9.3 Algorithmes

Pour insérer des algorithmes, il est possible d'utiliser l'environnement `lstlisting` du package `listings` vu à la section précédente. Une solution plus pratique consiste à utiliser l'environnement `algorithm` du package `algorithm2e`.

```
x ← 5 ;
while x > 0 do
  | print x ;
  | x ← x + 1
end
```

Dans sa version basique, il suffit de placer le code de l'algorithme dans l'environnement `algorithm`. La commande `\gets` indique une affectation et la commande `\While` permet d'insérer une boucle. Chaque instruction simple doit se terminer par `\;`.


Code
algorithm2e

```
\begin{algorithm}
$x \gets 5$ \;
\While{$x > 0$}{
  print $x$ \;
  $x \gets x + 1$
}
\end{algorithm}
```

9.3.1 Instructions prédéfinies

De nombreuses instructions sont prédéfinies. Voici les plus courantes :

- Entrées/Sorties : `\KwData`, `\KwResult`, `\KwIn` et `\KwOut`

- Intervalle : `\KwTo`
- Renvois de valeur : `\KwRet` ou `\Return`
- Conditions : `\If`, `\ElseIf`, `\Else` et `\eIf`
- Choix : `\Switch`, `\Case` et `\Other`
- Boucles : `\For`, `\While`, `\ForEach`, `\ForAll` et `\Repeat`

Algorithme 1: Un exemple d'algorithme.


Input : N un entier positif

```

sum ← 0 ;
for  $x$  ← 0 to  $N$  do
  | if  $x$  est pair then
  | | sum ← sum/2 ;
  | else
  | | sum ← sum + 1 ;
  | end
end
return  $x$ 

```

Cet exemple utilise quelques-unes des instructions prédéfinies. De plus, on a déjà appliqué quelques modifications de style. On verra en détails ces modifications à la section 9.3.3. Remarquez également la commande `\BlankLine` qui permet d'insérer un petit espace vertical, ainsi que la commande `\caption` qui est utilisée pour définir la légende. Pour que le mot « *Algorithme* » apparaisse en français, il faut ajouter l'option `french` en important le package `algorithm2e`.


Code

algorithm2e

```

\begin{algorithm}
\caption{Un exemple d'algorithme.}
\KwIn{$N$ un entier positif}
\BlankLine
$sum \gets 0$ \;
\For{$x \gets 0$ \KwTo $N$}{
  \eIf{$x$ est pair}{
    $sum \gets sum / 2$ \;
  }{
    $sum \gets sum + 1$ \;
  }
}
\Return{$x$}
\end{algorithm}

```


9.3.2 Nouvelles instructions

On peut définir des nouvelles instructions avec différentes commandes. Les principales sont `\SetKw` et `\SetKwInput`.

Precondition : S un ensemble fini d'entiers

```
sum ← 0 ;
foreach  $i \in S$  do
  |  $sum \leftarrow sum + i$  ;
end
print  $sum$  ;
```

La commande `\SetKw` permet de définir un nouveau mot réservé et la commande `\SetKwInput` permet de définir une nouvelle instruction de type input.

 **Code**
algorithm2e

```

\SetKw{KwPrint}{print}
\SetKwInput{KwPre}{Precondition}

\begin{algorithm}
\KwPre{$$$ un ensemble fini d'entiers}
\BlankLine
$sum \gets 0$ \;
\ForEach{$i \in S$}{
  $sum \gets sum + i$ \;
}
\KwPrint{$sum$} \;
\end{algorithm>
```

9.3.3 Style

On peut modifier le style des algorithmes via les options du package lorsqu'on l'importe. Les options principales sont :

- `boxed` ou `boxruled` pour avoir un cadre autour de l'algorithme, avec la légende dedans ou pas ;
- `ruled` ou `algoruled` pour avoir des traits horizontaux avant et après l'algorithme, avec plus ou moins d'espace ;
- `lined`, `vlined` ou `noline` pour avoir des traits verticaux pour les instructions composées, ou rien du tout ;
- `linesnumbered` pour avoir les numéros de ligne ;

- `longend`, `shortend` ou `noend` pour des fins d'instructions longues ou courtes, ou aucune fin.

Enfin, on peut supprimer les points-virgules affichés par `\;` avec la commande `\DontPrintSemicolon`. On peut modifier la bordure avec la commande `\RestyleAlgo`. Les traits verticaux sont modifiables avec `\SetLine`, `\SetAlgoVlined` et `\SetNoline`. On peut obtenir les numéros de ligne avec `\LinesNumbered`.

Input : n un entier positif

Output : la valeur de la somme $1 + 2 + \dots + n$

```

1  $sum \leftarrow 0$ 
2 for  $i \leftarrow 1$  to  $n$  do
3    $sum \leftarrow sum + i$ 
4 return  $sum$ 

```

Code

algorithm2e

```

\RestyleAlgo{boxed}

\begin{algorithm}
\DontPrintSemicolon
\SetAlgoVlined
\LinesNumbered

\KwIn{$n$ un entier positif}
\KwOut{la valeur de la somme $1 + 2 + \cdots + n$}
\BlankLine
$sum \gets 0$ \;
\For{$i \gets 1$ \KwTo $n$}{
  $sum \gets sum + i$ \;
}
\Return{$sum$} \;
\end{algorithm}

```


CHAPITRE 10

Bibliographies

Dans ce chapitre, on va voir comment insérer une bibliographie en utilisant l'outil `BIBTEX`. On verra qu'il y a deux manières d'en insérer une dans un fichier. La seconde permet de gérer sa bibliographie dans un fichier séparé et nécessite l'outil `BIBTEX` pour faire la mise en page. Cela permet de constituer des bases de données bibliographiques et de les partager entre documents et avec d'autres personnes.

10.1 Insérer une bibliographie


Une bibliographie est insérée avec l'environnement `thebibliography`. Chaque élément est inséré avec la commande `\bibitem` qui prend comme paramètre une étiquette (pour faire référence à l'élément bibliographique).

Références

- [1] Christian ROLLAND. *L^AT_EX par la pratique*. O'Reilly, 1999.
- [KNU90] Donald E. KNUTH. *The T_EXbook*. Addison-Wesley, 1990.
- [2] Frank MITTELBACH and Michel GOOSENS. *The L^AT_EX Companion*. Addison-Wesley, 2004.

L'environnement `thebibliography` prend en option la plus longue étiquette. Celle-ci sera utilisée pour aligner les différents éléments de la bibliographie. Par défaut, ces différents éléments sont numérotés avec des chiffres romains. On peut choisir autre chose avec l'option de la commande `\bibitem`.

Comme vous pouvez le remarquer, il faut soi-même définir toute la mise en page de l'entrée de la bibliographie. On verra plus loin dans le chapitre que l'outil `BIBTEX` fera cette mise en page automatiquement.

 **Code**

```

\begin{thebibliography}{KNU90}
  \bibitem[latexpratique]{Christian \textsc{Rolland}. \emph{\LaTeX{}} par la pratique}. O'Reilly, 1999.
  \bibitem[KNU90]{texbook} Donald E. \textsc{Knuth}. \emph{The \TeX book}. Addison-Wesley, 1990.
  \bibitem[latexcompanion]{Frank \textsc{Mittelbach} and Michel \textsc{Goosens}. \emph{The \LaTeX{}} Companion}. Addison-Wesley, 2004.
\end{thebibliography}


```

10.1.1 Citer une référence bibliographique

Une fois une bibliographie insérée, on peut citer une référence en utilisant la commande `\cite`. Celle-ci prend en paramètre l'étiquette de l'entrée que vous souhaitez référencer. On peut également en donner plusieurs séparées par des virgules.

Notre bibliothèque propose trois livres [1, KNU90, 2] :

- Les livres [1, 2] traitent de `LATEX`.
- Le livre [KNU90] traite de `TEX`.

 **Code**

```


Notre bibliothèque propose trois livres-\cite{latexpratique, texbook, latexcompanion} :
\begin{itemize}
  \item Les livres-\cite{latexpratique, latexcompanion} traitent de \LaTeX.
  \item Le livre-\cite{texbook} traite de \TeX.
\end{itemize}

```

10.2 Utiliser Bib_TE_X

Une autre manière de gérer une bibliographie consiste à définir une base de données dans un fichier `.bib`. Ensuite, on utilise l'outil `BIBTEX` qui va analyser les entrées référencées et les insérer dans le document tout en faisant leur mise en page.

Une telle base de données est un fichier texte constitué d'enregistrements. Chaque enregistrement correspond à une entrée bibliographique et est composé de champs renseignant le titre, les auteurs ... Voici un exemple avec deux enregistrements :


 **Code**

```
@book{latexpratique,
  author={Christian Rolland},
  title={\LaTeX{} par la pratique},
  year={1999},
  publisher={O'Reilly}
}

@book{texbook,
  author={Donald E. Knuth},
  title={The \TeX book},
  year={1990},
  publisher={Addison-Wesley}
}
```

Cette base de données doit être enregistrée comme un fichier texte avec l'extension `.bib`. Supposons qu'on l'enregistre dans le fichier `mabiblio.bib`.

On insère ensuite la bibliographie avec la commande `\bibliography` qui prend en paramètre le nom du fichier `.bib`. On peut également définir le style avec la commande `\bibliographystyle`. Tous les éléments qui auront été référencés avec `\cite` seront inclus dans la bibliographie insérée, à l'endroit où les deux commandes suivantes auront été insérées.

 **Code**


```
\bibliography{mabiblio} % chargement du fichier mabiblio.bib
\bibliographystyle{alpha}
```

Parmi les différents styles par défaut, on trouve `plain`, `unsrt`, `alpha` et `abbrv`. On peut également insérer un élément de la base de données, sans le citer en utilisant la commande `\nocite`.

Si on désire que toutes les entrées de la base de données soient insérées, il suffit de faire un `\nocite{*}` avant d'insérer la base avec `\bibliography`.

10.2.1 Bases de données bibliographique


Les enregistrements d'une base de données bibliographique ont la forme suivante :

 **Code**

```
@TypeDocument{Étiquette,
  Champ1={Valeur 1},
  Champ2="Valeur 2",
  ...
  Champn={Valeur n}
}
```

On peut voir que les valeurs des champs sont soit placées entre accolades, soit entre guillemets doubles. Les nombres, ainsi que les mots spéciaux ne doivent pas être entre accolades ou guillemets. Tout ce qui n'est pas dans un enregistrement est considéré comme un commentaire.

On peut définir des constantes avec des enregistrements de la forme `@string`. On peut concaténer des chaînes avec le caractère `#`. Enfin, il existe des constantes prédéfinies pour les différents mois de l'année : `jan`, `feb`, `mar`, `apr`, `may`, `jun`, `jul`, `aug`, `sep`, `oct`, `nov` et `dec`.

 **Code**

```
@string{intro="Une introduction à "}

@book{introLaTeX,
  year=2010,
  month=dec,
  title=intro # "'LaTeX"
}
```

10.2.2 Types de document

Les différents types de documents sont repris ci-dessous. La première liste indique les champs obligatoires et la seconde les champs optionnels.

article (article d'un journal ou d'une revue)
author, title, journal, year volume, number, pages, month, note, key
book (livre avec éditeur)
author (ou editor), title, publisher, year volume, series, address, edition, month, note, key, pages
booklet (document imprimé sans éditeur)
title author, howpublished, address, month, year, note, key
conference (comme <code>inproceedings</code>)

inbook (partie d'un livre, chapitre ou ensemble de pages) author (ou editor), title, chapter (ou pages), publisher, year volume, series, address, edition, month, note, key
incollection (partie d'un livre qui possède son propre titre) author, title, booktitle, year editor, pages, organization, publisher, address, month, note, key
inproceedings (article dans actes de conférence) author, title, booktitle, year editor, pages, volume (ou number), series, organization, publisher, address, month, year, note
manual (documentation technique) title author, organization, address, edition, month, year, note
masterthesis (thèse de maîtrise) author, title, school, year type, address, month, note
misc (autre document) — author, title, howpublished, month, year, note, key
phdthesis (thèse de maîtrise) author, title, school, year type, address, month, note
proceedings (actes de conférence) title, year editor, publisher, organization, address, month, note, key
techreport (rapport technique) author, title, institution, year type, number, address, month, note, key
unpublished (document avec auteur et titre, mais non publié) author, title, note month, year, key

10.2.3 Champs

Voici une description des différents champs :

- `address` : adresse de l'éditeur
- `annotate` : une annotation
- `author` : les auteurs, séparés par `and`
- `booktitle` : titre du livre
- `chapter` : numéro de chapitre
- `crossref` : étiquette d'une référence croisée
- `edition` : l'édition du livre
- `editor` : l'éditeur du livre
- `eprint` : publication électronique
- `howpublished` : comment il a été publié
- `institution` : institution impliquée dans la publication
- `journal` : journal ou magazine où il a été publié
- `key` : champ caché pour définir l'ordre alphabétique
- `month` : mois de publication
- `note` : information diverse
- `number` : numéro du journal ou du magazine
- `organization` : sponsor d'une conférence
- `pages` : numéros des pages, séparés par des virgules
- `publisher` : maison d'édition
- `school` : école dans laquelle la thèse a été écrite
- `series` : série de livres dans laquelle le livre a été publié
- `title` : titre du document
- `type` : le type
- `url` : l'adresse URL
- `volume` : numéro du volume
- `year` : année de publication

Pour les intervalles de page, on sépare la première et la dernière page avec un tiret moyen obtenu avec `--`.


10.2.4 Site web

BIB_TE_X n'étant pas tout jeune, il n'y a pas de type de document représentant les pages web. Pour faire référence à un site web, on va utiliser le type `@misc` avec le champ `howpublished`.

Références

[Com10] Sébastien Combéfis. Tutoriel L^AT_EX. <http://www.ukonline.be/programmation/latex/tutoriel/>, July 2010.

Il faut bien entendu charger le package `url` dans le document qui utilise cette bibliographie. Dans cet exemple, on a modifié le style des URL avec `\urlstyle{sf}`.

 **Code**

```
@misc{latex-tutorial,
  author={S\'ebastien Comb\'efis},
  title={Tutoriel {\LaTeX}},
  howpublished={\url{http://www.ukonline.be/programmation/latex/
    /tutoriel/}},
  year={2010},
  month=jul
}
```


CHAPITRE 11

Dessin

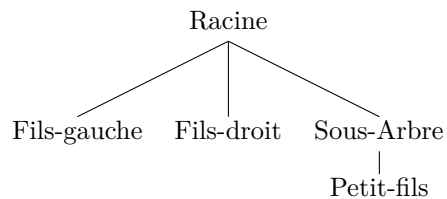
On pourrait rédiger plus d'un livre sur le dessin avec \LaTeX . Dans ce chapitre, on va rapidement passer sur les deux packages prédominants que sont \PSTricks et \TikZ/PGF . On verra également différents packages permettant de faire des dessins plus ciblés comme des arbres grammaticaux pour les linguistes, des molécules pour les chimistes, des circuits pour les électriciens ...

11.1 Arbres

Pour dessiner des arbres, on peut utiliser le package spécialisé \qtree . Ce dernier permet de facilement dessiner des arbres dont on donne la description avec la commande \Tree . On décrit un arbre par un simple mot pour une feuille ou par une liste de nœuds séparés par des espaces et mis entre crochets pour un sous-arbre, la racine étant précédée d'un point.

```
 $\text{\Tree}$  [.Racine Fils-gauche Fils-droit [.Sous-Arbre Petit-fils ] ]
```

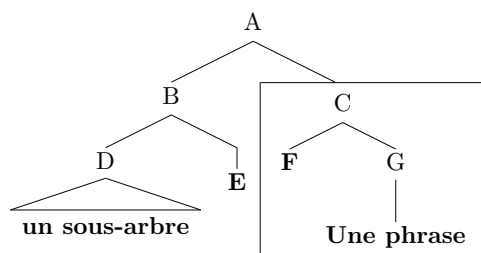
Il faut bien veiller à ce que chaque crochet soit précédé d'une espace.



Les arbres sont automatiquement centrés. Pour annuler ce comportement, il suffit de spécifier l'option `nocenter` au moment de charger le package. On peut modifier le style en redéfinissant les commandes vides suivantes :

- `qtreeinithook` est appelée au début de chaque arbre
- `qtreefinalhook` est appelée à la fin de chaque arbre
- `qleafhook` est appelée pour chaque feuille
- et `qlabelhook` est appelée pour chaque nœud interne

Enfin, on peut obtenir tout un sous-arbre sous forme d'un triangle avec la commande `\qroof`. On peut également encadrer un sous-arbre avec la commande `\qframesubtree`. Remarquez aussi qu'on n'est pas obligé d'avoir un texte pour chaque nœud interne. Voici un exemple plus élaboré :



Remarquez que `\qframesubtree` ne fonctionnera pas si vous avez le package `babel` qui est chargé.

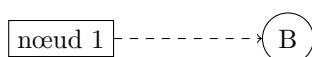
Code
qtree

```

\newcommand{\qleafhook}{\bf}
\Tree [.A
  [.B
    \qroof{un sous-arbre}.D
    [E ]
  ]
  [.C
    [.F ]
    [.G [{Une phrase} ] ]
  ] !{\qframesubtree}
]
```

11.2 TikZ/PGF

Un package de dessin très connu et fonctionnant avec pdfTeX est le package TikZ/PGF. Nous n'allons pas le détailler ici car cela nécessiterait un livre complet, mais on va simplement regarder un ou deux exemples. L'idée de base est de créer des nœuds qu'on va ensuite relier par des arêtes. On insère une nouvelle figure avec l'environnement `tikzpicture`. Voyons un premier exemple.



Code

tikz,pgf

```

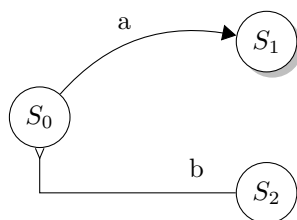
\begin{tikzpicture}
  \node[draw,shape=rectangle] (A) at (0,0) {nœud 1};
  \node[draw,shape=circle] (B) at (3,0) {B};
  \draw (A) edge[dashed,->] (B);
\end{tikzpicture}

```

La commande `\node` permet d'ajouter un nouveau nœud. On définit d'abord des options, ensuite un nom entre parenthèses suivi de `at` et de la coordonnée du nœud et enfin, on place le texte qui doit figurer dans le nœud entre accolades.

La commande `\draw` permet de faire un dessin. Dans notre cas, on dessine une arête entre le nœud A et le nœud B. Pour cela, on utilise `edge` qui permet notamment de définir des options.

Il existe de nombreuses extensions pour TikZ/PGF. L'exemple suivant montre l'utilisation de deux extensions. La première permet d'avoir des styles de flèches différents et la seconde permet d'avoir des ombres. Vous pourrez également voir sur l'exemple suivant qu'on peut définir des options par défaut et qu'on peut ajouter des étiquettes sur les arêtes.



Code
tikz,pgf

```

\begin{tikzpicture}[shape=circle,auto]
  \node[draw] (A) at (0,0) {$S_0$};
  \node[draw,fill=white,drop shadow] (B) at (3,1) {$S_1$};
  \node[draw] (C) at (3,-1) {$S_2$};
  \draw (A) edge[-triangle 60,bend left] node {a} (B);
  \draw[open reversed 45-] (A) |- node[pos=0.9] {b} (C);
\end{tikzpicture>

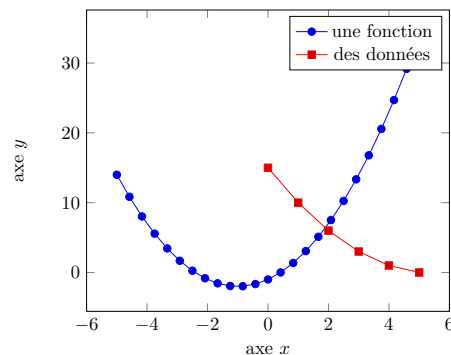
```

Il faut préalablement importer les deux extensions avec la commande `\usetikzlibrary` dans le préambule du document :

```
\usetikzlibrary{arrows,shadows}
```

11.2.1 Dessiner des données

On peut dessiner des données en utilisant le package `pgfplots` qui se base sur le package TikZ/PGF.



Code
pgfplots

```

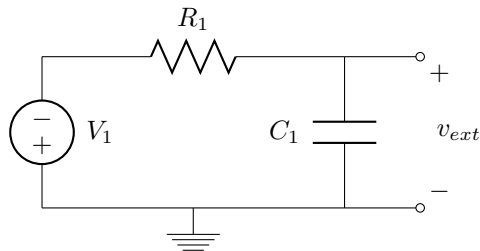
\begin{tikzpicture}[scale=0.7]
  \begin{axis}[xlabel=axe $x$,ylabel=axe $y$]
    \addplot {x^2 + 2*x - 1};
    \addlegendentry{une fonction}
    \addplot coordinates {
      (0,15)
      (1,10)
      (2,6)
      (3,3)
      (4,1)
      (5,0)
    };
    \addlegendentry{des données}
  \end{axis}
\end{tikzpicture>

```

On utilise l'environnement `axis` pour définir un nouveau graphe. Ensuite, on ajoute une courbe grâce à la commande `\addplot`. On peut soit définir une équation, soit donner une série de points. Enfin, on peut obtenir une légende avec `\addlegendentry`. Il existe de nombreuses autres possibilités offertes par ce package et nous vous invitons à consulter sa documentation détaillée.

11.2.2 Dessiner des circuits électriques

Le package `circuitikz` permet de dessiner des circuits électriques facilement. Il se base, comme son nom le suggère, sur TikZ/PGF. Il y a essentiellement des monopoles et des dipôles. Chacun de ces éléments est représenté par un nœud. On va utiliser `node` pour les monopoles et `to` pour les dipôles.



Code
circuitikz

```

\begin{circuitikz}[american]
  \draw (-2,1) to[voltage source,l=$V_1$] (-2,-1);
  \draw (-2,1) to[R,l=$R_1$] (2,1);
  \draw (2,1) to[short] (3,1);
  \draw (2,1) to[C,l=$C_1$] (2,-1);
  \draw (-2,-1) to[short] (3,-1);
  \draw (0,-1) node[ground] {};
  \draw (3,1) to[open,o-o,v^=$v_{ext}$] (3,-1);
\end{circuitikz

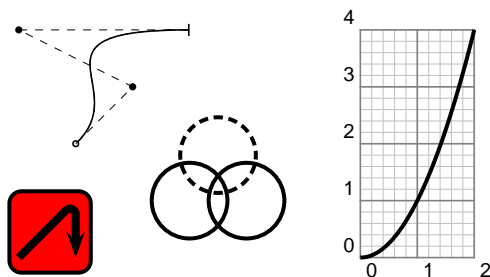
```

N'hésitez pas à consulter la documentation officielle pour avoir la liste des tous les composants qu'il est possible de dessiner.

11.3 PSTricks

PSTricks est un package de dessin qui permet d'exploiter les possibilités de PostScript avec L^AT_EX. Ce package n'est pas compatible avec pdfT_EX et vous devrez dès lors passer par un document en PostScript à convertir en un PDF.

Les dessins se font grâce à des commandes à qui il faut passer, en plus d'options et de paramètres, des coordonnées sous la forme (x,y).



Code
pstricks, pst-all

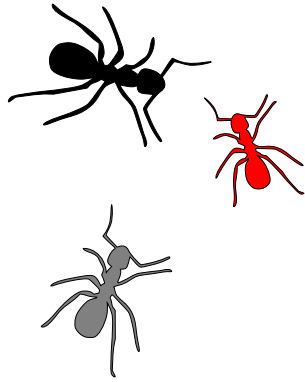
```

\pspicture(0,0)(8,5)
  \psframe[linewidth=2pt,framearc=0.3,fillstyle=solid,
           fillcolor=red](-0.2,-0.3)(1.3,1.2)
  \psline[linewidth=4pt,linearc=0.1]{->}(0,0)(1,1)(1,0)
  \pscicle[linewidth=2pt](3,1){0.7}
  \pscicle[linewidth=2pt,linestyle=dashed](3.5,1.8){0.7}
  \pscicle[linewidth=2pt](4,1){0.7}
  \psbezier[showpoints=true]{o-|}(1,2)(2,3)(0,4)(3,4)
  \psgrid[origin={6,0},gridcolor=gray,
          subgridcolor=lightgray](0,0)(0,0)(2,4)
  \psplot[origin={6,0},plotpoints=100,
          linewidth=2pt]{0}{2}{x x mul}
\endpspicture

```

Il existe bien d'autres commandes ainsi que de nombreux autres packages qui ont été construits sur base de PSTricks. On peut notamment dessiner des montages pour des expériences de chimie, faire des dessins 3D ...

Voici par exemple une utilisation du package `pst-fun` qui permet notamment d'insérer des fourmis avec la commande `\pstAnt`.

**Code**

pstricks, pst-all, pst-fun

```
\pspicture(0,0)(8,5)
  \rput(1,1){\psscalebox{0.5}{\psAnt}}
  \rput{45}(3,3){\psscalebox{0.4}{\psAnt[fillcolor=red]}}
  \rput{-80}(1,4){\psscalebox{0.6}{\psAnt[fillcolor=black]}}
\endpspicture
```


CHAPITRE 12

Personnaliser L^AT_EX


Dans ce chapitre, on va voir comment définir ses propres commandes et environnements. On verra également quelques packages et commandes permettant d'effectuer des tâches répétitives et de manière automatique. Enfin, on terminera en voyant comment définir ses propres environnements flottants et comment écrire dans des fichiers externes.

12.1 Commandes et environnements

On définit une nouvelle commande avec `\newcommand`. En option, on spécifie le nombre de paramètres qu'elle prend. Elle peut en avoir au maximum neuf.

Arachnophobie : Peur des araignées.

Dans la définition de la nouvelle commande, on utilise `#i` avec *i* étant un chiffre entre 1 et 9, pour obtenir la valeur du *i*^e paramètre.

 **Code**

```
\newcommand{\definition}[2]{\textbf{#1 : } #2.}
\definition{Arachnophobie}{Peur des araignées}
```

On définit un nouvel environnement avec la commande `\newenvironment`. Cette commande prend trois paramètres qui sont le nom de l'environnement, le code à placer avant et celui à placer après. En option, on peut préciser le nombre de paramètres que l'environnement prend.

————— Le titre —————

Le contenu de la boîte qui est délimitée
par un trait horizontal en haut, mais
également en bas.

—————

On a donc ici défini un environnement `titledbox` qui permet d’avoir une boîte d’une certaine largeur avec un titre. La boîte est délimitée en haut et en bas par un trait horizontal.



Code

```
\newenvironment{titledbox}[2]({\begin{minipage}{#1}%
\hrulefill-\raisebox{-0.4ex}{#2}-\hrulefill\par\smallskip}%
{\par\hrulefill\end{minipage}}

\begin{titledbox}{6cm}{Le titre}
Le contenu de la boîte qui est délimitée par un trait horizontal
en haut, mais également en bas.
\end{titledbox}
```

On désire parfois redéfinir une commande ou un environnement qui existait préalablement. Pour cela, il faut utiliser `\renewcommand` et `\renewenvironment`.

12.2 Tests et répétitions

Voyons une série de packages qui proposent des commandes qui vont permettre de faire un peu de programmation. On va pouvoir faire des tests, des boucles et des traitements plus sophistiqués sur du texte.


12.2.1 Conditions

Le package `ifthen` définit la commande `\ifthenelse` qui permet d’écrire du code L^AT_EX seulement si une certaine condition est vérifiée. La commande prend trois paramètres qui sont respectivement une condition, le code à insérer si la condition est vraie et celui à insérer si elle est fausse.

12 est plus grand que 7
42 est plus petit que 69

Pour définir la condition, on peut utiliser les opérateurs logiques `\AND`, `\OR` et `\NOT`, les parenthèses `\(` et `\)`, les opérateurs de comparaison `<`, `=` et `>` pour comparer des nombres. La commande `\isodd` teste si un nombre est impair. La commande `\isundefined` teste si une commande

n'est pas définie. La commande `\equal` teste si deux chaînes de caractères sont identiques. Pour comparer des longueurs, on ne peut pas utiliser directement `<`, `=` et `>`, mais il faut le faire dans la commande `\lengthtest`.


Code
ifthen

```

\newcommand{\greater}[2]{#1 est plus \ifthenelse{#1>#2}{grand}{
  petit} que #2}

\greater{12}{7} \\
\greater{42}{69}

```

12.2.2 Boucles

On a parfois besoin de répéter plusieurs fois le même code \LaTeX . Pour ce faire, on peut utiliser la commande `\multido` définie dans le package de même nom.


```

blablablabla
1864679sp 2051146sp 2237613sp 2424080sp 2610547sp
4 3 2 1 0
0.0 0.15 0.29999 0.44998 0.59998
0.00 0.15 0.30 0.45 0.60

```

On peut utiliser la commande pour répéter un certain nombre de fois du code \LaTeX . On peut également définir un compteur, le type de ce dernier dépendant de la première lettre du nom utilisé. La déclaration du compteur a toujours la forme `\nom=initial+increment`.

- `d` pour une longueur
- `i` pour un entier
- `r` pour un réel
- `n` pour un nombre avec un nombre fixé de chiffres après la décimale


Code
multido

```

\multido{}{5}{bla} \\
\multido{\d=1cm+1mm}{5}{\d, } \\
\multido{\i=4+-1}{5}{\i, } \\
\multido{\r=0+0.15}{5}{\r, } \\
\multido{\n=0.00+0.15}{5}{\n, }

```

Le nombre d'itérations est accessible par `\multidocount`. On peut arrêter de boucler avec la commande `\multidostop`. Enfin, remarquez que pour faire un décompte à l'envers, il faut écrire `\i=4+-1` car `\i=4-1` ne fonctionnerait pas.


12.2.3 Manipuler des strings

Le package `coolstr` définit trois commandes qui permettent de tester si une chaîne de caractères correspond à un nombre décimal, à un nombre ou à un nombre entier : `\isdecimal`, `\isnumeric` et `\isint`.

Le package définit également la commande `\substr` qui permet d'extraire une sous-chaîne d'une chaîne. Le premier paramètre correspond à la chaîne à traiter, le second à l'indice du caractère de début et le dernier au nombre de caractères à extraire.

Né en 1983.

Ici, on part donc du septième caractère et on en prend quatre. Cela permet donc d'extraire l'année de naissance. Il faut savoir que le résultat de la commande `\substr` est un string et ne pourra donc par exemple pas être utilisé dans un calcul.

	Code	coolstr
<pre>\newcommand{\printinfo}[1]{Né en \substr{#1}{7}{4}.} \printinfo{06-11-1983}</pre>		

Le package `xstring` propose beaucoup plus de commandes. On va pouvoir faire des tests, des recherches et des remplacements dans une chaîne et, enfin, compter. Nous n'allons pas toutes les détailler ici, mais voici la liste des essentielles :

1. Commandes qui effectuent un test

- `\IfSubStr` teste si une chaîne est une sous-chaîne d'une autre
- `\IfBeginWith` et `\IfEndWith` testent si une chaîne commence ou se termine par une autre
- `\IfInteger` et `\IfDecimal` testent si une chaîne est un entier ou un nombre décimal
- `\IfStrEq` et `\IfStrEqCase` testent si deux chaînes sont égales, en ignorant ou non la casse


2. Commandes qui renvoient une chaîne

- `\StrBefore` et `\StrBehind` renvoient la sous-chaîne se trouvant avant ou après une sous-chaîne d’une autre chaîne
- `\StrBetween` renvoie une sous-chaîne se trouvant entre deux sous-chaînes d’une autre chaîne
- `\StrSubstitute` renvoie une chaîne dont les occurrences d’une sous-chaîne ont été remplacées par une chaîne
- `\StrDel` renvoie une chaîne dont les occurrences d’une sous-chaîne ont été supprimées
- `\StrLeft` et `\StrRight` renvoient une chaîne contenant les i premiers ou derniers caractères d’une chaîne
- `\StrChar` renvoie le i^{e} caractère d’une chaîne
- `\StrMid` renvoie la sous-chaîne allant du i^{e} caractère au j^{e}

3. Commandes qui renvoient un nombre

- `\StrLen` renvoie la longueur d’une chaîne
- `\StrCount` renvoie le nombre de fois qu’une sous-chaîne apparaît dans une autre
- `\StrPosition` renvoie la position d’une sous-chaîne dans une autre

Voyons un petit exemple avec le résultat qui est produit par ce code :


xstring

Code

```

\StrRight{LaTeX}{3} \\
\StrSubstitute{toto est tout}{to}{tu} \\
\StrLen{Anticonstitutionnellement} \\
\IfBeginWith{Maison}{Mais}{OUI}{NON}

```

```

TeX
tutu est tuut
25
OUI

```

12.3 Définir un flottant

Il existe deux types de flottants par défaut en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: les figures et les tables. Un des intérêts de ceux-ci est la possibilité d’ajouter une légende avec la commande `\caption`. Un autre est de pouvoir générer une liste

de tous ceux insérés dans le document. On définit un nouveau type de flottant avec la commande `\newfloat` définie dans le package `float`.


Exemple 1 Exemple d'utilisation de la formule de Pythagore.

Soit un triangle rectangle dont les longueurs de deux des côtés adjacents à l'angle droit valent respectivement 3 et 4 centimètres. Grâce au théorème de Pythagore, on peut écrire :

$$h^2 = 3^2 + 4^2 = 9 + 16 = 25$$

La longueur de l'hypothénuse vaut donc $h = \sqrt{25} = 5$ centimètres.

La commande `\floatstyle` permet de définir le style de tous les nouveaux flottants qui seront créés avec `\newfloat`. La commande `\newfloat` prend trois paramètres qui sont le nom du flottant, son positionnement et enfin une extension pour le fichier auxiliaire utilisé pour les lister. Le nom est défini avec `\floatname` et on insère la liste des flottants avec `\listof`.

 **Code**
float

```

\floatstyle{ruled}
\newfloat{example}{!ht}{lex}
\floatname{example}{Exemple}

\begin{example}
  Soit un triangle rectangle dont les longueurs % ...
  \caption{Exemple d'utilisation de la formule de Pythagore.}
\end{example}

\listof{example}{Liste des exemples}

```

Les différents styles sont `plain`, `plaintop`, `boxed` et `ruled`. Enfin, avec le package `float`, le nouveau descripteur de positionnement `H` permet de forcer le flottant à être placé à l'endroit où il a été déclaré. En fait, cela annule le comportement flottant.

On peut également modifier le style d'un flottant déjà existant avec la commande `\restylefloat`. Par exemple, pour faire en sorte que toutes les figures soient entourées d'une boîte, on peut écrire :

```

\floatstyle{boxed}
\restylefloat{figure}


```


12.4 Définir un compteur

On est parfois amené à devoir définir un nouveau compteur. On les utilise pour la numérotation ou l'énumération. Un nouveau compteur se définit avec la commande `\newcounter` qui prend en paramètre un nom.

On initialise la valeur d'un compteur en utilisant la commande `\setcounter` et on modifie sa valeur avec `\addtocounter`. Enfin, on affiche la valeur d'un compteur avec `\the` suivi du nom du compteur.

Il y avait 12 bières,
Marie en apporte quatre de plus, ce qui en fait 16,
mais Pierre est passé et il n'en reste plus que 9.

 **Code**

```
\newcounter{cntBeers}
\setcounter{cntBeers}{12}
Il y avait \thecntBeers{} bières,

\addtocounter{cntBeers}{4}
Marie en apporte quatre de plus, ce qui en fait \thecntBeers,

\addtocounter{cntBeers}{-7}
mais Pierre est passé et il n'en reste plus que \thecntBeers.
```

On peut afficher un compteur avec un autre style en utilisant l'une des commandes suivantes :

- `\arabic` : chiffres arabes
- `\roman` ou `\Roman` : chiffres romains minuscules ou majuscules
- `\alph` ou `\Alph` : lettres romaines minuscules ou majuscules
- `\fnsymbol` : symboles (pour un compteur ne dépassant pas 9)

12.5 Définir une longueur

On a parfois besoin de définir des longueurs, et pour cela, on utilise la commande `\newlength` qui prend un nom en paramètre. Attention cependant que ce dernier doit être précédé d'un backslash, contrairement au nom qu'on donnait à un nouveau compteur.

Une fois la longueur créée, on peut l'initialiser avec `\setlength` et on peut modifier sa valeur avec `\addtolength`. On ne peut pas directement afficher la valeur d'une longueur, mais seulement l'utiliser dans des commandes

comme `\hspace` par exemple. Si on veut l'afficher, il faut utiliser des packages dédiés comme celui décrit à la section 2.4.5 (page 31).

Papa
est là!



Code

```
\newlength{\sep}
\setlength{\sep}{1cm}
\hspace{\sep} Papa

\addtolength{\sep}{5mm}
\hspace{\sep} est là !
```

Il existe également les commandes `\settowidth`, `\settoheight` et `\settodepth` qui permettent de fixer une longueur égale à la largeur, la hauteur ou la profondeur d'un texte spécifié en second paramètre. On peut par exemple obtenir un 0 traversé par un I (⓪) avec le code suivant :



Code

```
\newlength{\oisp}\settowidth{\oisp}{0}
(0\hspace{-0.75\oisp}I\hspace{0.25\oisp})
```

CHAPITRE 13


Divers

Ce dernier chapitre fait le tour de quelques packages qui permettent de réaliser diverses choses comme insérer des mots-croisés, générer des nombres aléatoires, des filigranes, faire des calculs ...

13.1 Mots-croisés

Le package `cwpuzzle` permet d'insérer des mots-croisés, avec leurs solutions. Il est également possible de générer des sudoku ou des kakuro. Le package est très riche et offre de nombreuses autres possibilités.

	M	O	L	E	
L	A		O	T	A
	L	I	T		

 **Code** **cwpuzzle**

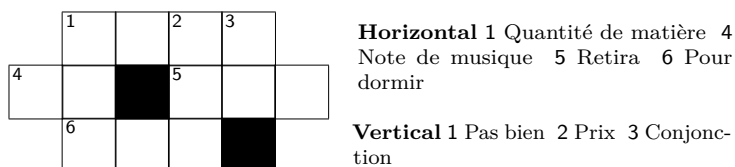
```
\PuzzleSolution
\begin{Puzzle}{4}{3}
|{}|M|O|L|E|{}|.
|L|A|*|O|T|A|.
|{}|L|I|T|*|{}|.
\end{Puzzle}
```

L'environnement `Puzzle` permet d'insérer des mots-croisés. Les cases sont définies à partir de la commande `|` qui prend un paramètre qui est soit vide (`{}`), soit une lettre, soit un astérisque (`*`).

La commande `|` prend deux options. La première est le numéro à mettre dans la case et la seconde est un paramètre de style. Les différentes valeurs possibles sont :

- `f` : cadre
- `.` : vide
- `*` : case noire
- `0` : cercle
- `o` : cadre et cercle
- `t`, `b`, `l` et `r` : trait renforcé en haut, bas, gauche et droite
- `T`, `B`, `L` et `R` : trait renforcé en haut, bas, gauche et droite et cadre

Les définitions doivent être insérées avec l'environnement `PuzzleClues`. La commande `\Clue` permet d'insérer une définition. Le second paramètre qui est le mot défini n'est actuellement pas utilisé.



Enfin, remarquez que `PuzzleSolution` permet d'afficher la solution et de masquer les définitions, tandis que `PuzzleUnsolved` permet de faire l'inverse.

Code

cwpuzzle

```

\begin{multicols}{2}
\PuzzleUnsolved
\begin{Puzzle}{4}{3}
|{}| |[1]M |0 |[2]L |[3]E |{}|.
|[4]L |A |* |[5]0 |T |A |.
|{}| |[6]L |I |T |* |{}|.
\end{Puzzle}
\begin{PuzzleClues}{\bf Horizontal}
\Clue{1}{MOLE}{Quantité de matière}
% ...
\end{PuzzleClues}


\begin{PuzzleClues}{\bf Vertical}
\Clue{1}{MAL}{Pas bien}
% ...
\end{PuzzleClues}
\end{multicols}

```

13.2 Code-barres


Pour réaliser des codes-barres pour des livres, correspondant à un numéro ISBN, le moyen le plus simple consiste à utiliser la commande `\EANisbn` du package `ean13isbn`. Il suffit de définir le format désiré (`SC0`, `SC1` ...) ainsi que le numéro ISBN en option de la commande.



	Code	ean13isbn
<code>\EANisbn [SC0, ISBN=978-80-7340-097-2]</code>		


Une autre possibilité plus riche consiste à utiliser le package `pst-barcode` qui se base sur PSTricks. Vous ne pourrez donc pas compiler votre document avec PDFL^AT_EX. La commande `\psbarcode` prend trois paramètres qui sont les données, les options et le type du code-barres.



	Code	pst-barcode
<pre>\begin{pspicture}(3.5,2.5) \psbarcode{9788073400972}{includetext}{ean13} \end{pspicture}</pre>		

De nombreux autres formats sont disponibles avec ce package. Pour en citer quelques-uns : EAN-13, UPC-A, onecode, ISBN, Postnet, PDF417, Data Matrix, QR Code.



 Code
pst-barcode

```


\begin{pspicture}(2,2)
  \psbarcode{http://www.latex-howto.be}{eclevel=M}{qrcode}
\end{pspicture}\quad
\begin{pspicture}(2,2)
  \psbarcode{http://www.latex-howto.be}{rows=36 columns=36
    parse}{datamatrix}
\end{pspicture>

```

13.3 Filigrane

On peut ajouter un filigrane sur toutes les pages d'un document grâce au package `watermark`. Le package définit les quatre commandes `\watermark`, `\leftwatermark`, `\rightwatermark` et `\thiswatermark` qui permettent de définir un filigrane qui sera sur toutes les pages, sur les pages paires, impaires ou seulement sur la page courante. Remarquez par exemple le logo L^AT_EX qui se trouve sur le fond de cette page.

Afin de positionner ce logo, il a fallu le placer dans une `minipage` pour pouvoir le précéder d'un espace vertical. Une autre solution consiste à utiliser TikZ/PGF avec l'option `overlay`.

 Code
watermark

```

\thiswatermark{
  \begin{minipage}{10cm}
  \vspace{14cm}
  \begin{center}
    \rotatebox{55}{\scalebox{8}{\color[gray]{0.9} \LaTeX}}
  \end{center}
  \end{minipage}
}
```

13.4 Nombre aléatoire


Grâce au package `lcg`, il est possible de générer des nombres pseudo-aléatoires. Une fois le package chargé avec ses options, on obtient le prochain nombre pseudo-aléatoire avec la commande `\rand`. Ce nombre est ensuite stocké dans le compteur `rand` et on peut afficher sa valeur avec les commandes décrites à la section 12.4. Parmi les options possibles, on retrouve `first` et `last` qui indiquent l'intervalle dans lequel les nombres sont choisis.

Si on veut par exemple lancer trois fois un dé, il faut importer :

```
\usepackage[first=1,last=6]{lcg}
```

et un tirage donnera par exemple :

6, 2 et 5




Code
lcg

```
\rand\arabic{rand}, \rand\arabic{rand} et \rand\arabic{rand}
```

13.5 Alphabet phonétique international

On peut écrire la prononciation des mots selon l'alphabet phonétique international (IPA) en utilisant le package `tipa`. La commande `\textipa` permet d'introduire la prononciation d'un mot.

LaTeX se prononce [`\textipa{"leItEk}`], [`\textipa{"leItEx}`] ou [`\textipa{"lA:tEk}`]



Code
tipa

```
\LaTeX{} se prononce [\textipa{"leItEk}], [\textipa{"leItEx}] ou
[\textipa{"lA:tEk}]
```

Le tableau 13.1 reprend les raccourcis qui permettent d'obtenir directement un caractère de l'alphabet phonétique international.

:	:	0	ɸ	@	ə	J	j	T	θ
;	;	1	i	A	ɑ	K	ɸ	U	ʊ
"	"	2	ʌ	B	β	L	ʌ	V	v
		3	ɜ	C	ɛ	M	ɱ	W	w
		4	ɥ	D	ð	N	ŋ	X	χ
		5	ɐ	E	ɛ	O	ɔ	Y	ɣ
		6	ɒ	F	ɸ	P	ʔ	Z	ʒ
		7	ɣ	G	ɣ	Q	ɸ		
		8	ø	H	ɦ	R	r		
		9	ɘ	I	ɪ	S	ʃ		

Table 13.1. Raccourcis d'écriture pour les caractères de l'IPA.

Des commandes spéciales permettent d'obtenir facilement des variantes de certains caractères. Tout d'abord, la commande `*` permet d'obtenir une lettre tournée de 180 degrés pour f, k, r, t ou w. Si on l'utilise avec j, n, h, l ou z, elle permet d'obtenir certains caractères particuliers.

J ɿ ɹ ʔ ʌ j ɲ ħ ʃ ʒ

**Code****tipa**

```
\textipa{\*f \*k \*r \*t \*w} \textipa{\*j \*n \*h \*l \*z}
```

Lorsque la commande `*` est suivie d'autre lettres, elle se contente de les imprimer sans aucune modification. Les trois autres commandes `\;`, `\:` et `\!` sont utilisées pour obtenir des petites majuscules, des symboles rétroflexes, occlusif ou clic.

B E A H L R
ɖ ʎ ɳ ʂ ʐ
ɓ ɗ ɡ ʃ ʈ ʉ

**Code****tipa**

```
\textipa{\;B \;E \;A \;H \;L \;R}  
\textipa{\:d \:l \:n \:r \:s \:z}  
\textipa{\!b \!d \!g \!j \!G \!o}
```

On peut également insérer les accents et marques diacritiques via des raccourcis présentés dans le tableau 13.2.

<code>\'a</code>	á		<code>\ t</code>	ṭ	<code>\ t</code>	t̚	
<code>\"a</code>	ä	<code>\"*{a}</code>	a̠	<code>\ a</code>	a̡	<code>\ a</code>	a̢
<code>\~a</code>	ã	<code>\~*{a}</code>	ã̠	<code>\ c{k}</code>	ḳ		
<code>\r{a}</code>	â	<code>\r*{a}</code>	â̠	<code>\ o</code>	ọ		
<code>\s{m}</code>	ṁ			<code>\ 'e</code>	ẹ		
<code>\.'e</code>	ě	<code>\'.e</code>	ě̠	<code>\ 'e</code>	ẹ		
<code>\'=a</code>	á̇			<code>\ <a</code>	ạ		
<code>\^ .a</code>	â̇			<code>\ >a</code>	ạ		
<code>\~ .a</code>	ã̇			<code>\ x{e}</code>	ẹ		
<code>\u=a</code>	ạ̃			<code>\ w{k}</code>	ḳ		
				<code>\ m{t}</code>	ṭ		

Table 13.2. Accents et marques diacritiques de l'IPA.

CHAPITRE 14

Classes de documents

Ce chapitre présente quelques classes de documents usuelles qui vous permettront de facilement rédiger des rapports, des lettres, des présentations, des folders, des CV ... À chaque fois, un exemple minimal vous est présenté avec le code complet correspondant. Afin d'en savoir plus sur les différentes classes présentées, nous vous renvoyons vers leurs documentations respectives.

Tous les exemples de ce chapitre sont disponibles au téléchargement sur :
<http://www.latex-howto.be>.

N'hésitez pas à les télécharger et à les utiliser comme modèles pour rédiger vos premiers documents.

14.1 Écrire une lettre



Code

```
\documentclass[11pt,a4paper]{letter}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{lmodern}

\address{Sébastien Combéfis \\ Rue et numéro \\ 1234 Ville}
\signature{Sébastien Combéfis}

\begin{document}
  \begin{letter}{Le destinataire \\ Sa rue \\ Et sa ville}
    \opening{Madame, Monsieur,}

    Je vous écris suite à notre entretien téléphonique de ce
    mercredi 6 octobre 2010 afin de \dots

    \closing{Je vous prie d'agréer, Madame, Monsieur, l'
      expression de mes sentiments distingués.}
    \cc{Service des achats}
    \encl{Le listing des clients potentiels.}
    \ps{PS : N'oubliez pas de me transférer les devis.}
  \end{letter}
\end{document}
```

Sébastien Combéfis
Rue et numéro
1234 Ville
8 octobre 2010

Le destinataire
Sa rue
Et sa ville

Madame, Monsieur,

Je vous écris suite à notre entretien téléphonique de ce mercredi 6 octobre 2010 afin de ...

Je vous prie d'agréer, Madame,
Monsieur, l'expression de mes
sentiments distingués.

Sébastien Combéfis

Copie à : Service des achats

P. J. : Le listing des clients potentiels.

PS : N'oubliez pas de me transférer les devis.

14.2 Présenter un CV



Code

```

\documentclass[11pt,a4paper]{moderncv}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{lmodern}

\moderncvtheme[blue]{casual}

\firstname{Sébastien}
\familyname{Combéfis}
\address{Rue et numéro}{1234 Ville}
\phone{+32123456789}
\email{info@latex-howto.be}

\begin{document}
  \maketitle

  \section{Études}

  \cventry{2004--2007}{Ingénieur civil informaticien}{
    Université catholique de Louvain}{Louvain-la-Neuve}{
    Grande Distinction}{}
  \cventry{2002--2004}{Candidature ingénieur civil}{Université
    catholique de Louvain}{Louvain-la-Neuve}{Distinction}{}
  \cventry{1996--2002}{Secondaire général renové Math-Sciences
    }{Lycée de Berlaymont}{Waterloo}{Grande Distinction}{}

  \section{Mémoire}

  \cvline{Titre}{Viral marketing and community detection
    algorithms}
  \cvline{Promoteur}{Pr. Pierre Dupont}
  \cvline{Description}{Étude des techniques de marketing ...}

  \section{Expérience}

  \cventry{2009}{Président}{UCLouvain ACM Student Chapter ASBL
    }{Louvain-la-Neuve}{}{Fondateur et président ...}
  \cventry{2007--2009}{Assistant}{Université catholique de
    Louvain}{Louvain-la-Neuve}{}{Assistant de cours ...}

  \section{Langues}

  \cvlanguage{Français}{Avancé}{Langue maternelle}
  \cvlanguage{Anglais}{Intermédiaire}{}

  \section{Informatique}

  \cvcomputer{Application}{Eclipse, OpenOffice}{OS}{Windows,
    Linux, Mac OS}
  \cvcomputer{Langage}{Java, C, C++, Python, PHP, MySQL, (X)
    HTML, CSS, \LaTeX}{}{}
\end{document}

```

Sébastien Combéfis

Études

2004–2007 **Ingénieur civil informaticien**, *Université catholique de Louvain*, Louvain-la-Neuve, Grande Distinction.

2002–2004 **Candidature ingénieur civil**, *Université catholique de Louvain*, Louvain-la-Neuve, Distinction.

1996–2002 **Secondaire général renové Math-Sciences**, *Lycée de Berlaymont*, Waterloo, Grande Distinction.

Mémoire

Titre Viral marketing and community detection algorithms
 Promoteur Pr. Pierre Dupont
 Description Étude des techniques de marketing viral sur différents types de réseaux, utilisation de structures communautaires dans ces réseaux pour accélérer les algorithmes et détection de communautés influentes.

Expérience

2009 **Président**, *UCLouvain ACM Student Chapter ASBL*, Louvain-la-Neuve.
 Fondateur et président de l'UCLouvain ACM Student Chapter ASBL, association qui a pour but de promouvoir l'informatique et ses applications auprès du grand public et notamment des étudiants, professeurs et professionnels ICT. Pour ce faire, elle organise divers événements tels que des conférences, des tutoriaux, des formations et des concours de programmation (<http://uclouvain.acm-sc.be>).

2007–2009 **Assistant**, *Université catholique de Louvain*, Louvain-la-Neuve.
 Assistant de cours au Département d'Ingénierie Informatique (INGI) de l'École Polytechnique de Louvain (EPL) de l'Université catholique de Louvain (UCLouvain).

Langues

Français	Avancé	<i>Langue maternelle</i>
Anglais	Intermédiaire	

Informatique

Application	Eclipse, OpenOffice	OS Windows, Linux, Mac OS
Langage	Java, C, C++, Python, PHP, MySQL, (X)HTML, CSS, \LaTeX	

Rue et numéro – 1234 Ville
 ☎ +32123456789 • ✉ info@latex-howto.be

14.3 Mettre en page un journal



Code

```

\documentclass[11pt,a4paper]{papertex}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}

\renewcommand{\logo}{
  \mylogo{\includegraphics[width=2cm]{uko-logo.png}}
}
\author{Sébastien Combéfis}
\title{World News}
\edition{UKO Publication}

\begin{document}
  \begin{frontpage}
    \firstimage{Antennae-Hubble.jpg}{Source: NASA}

    \firstnews{Découverte d'une nouvelle galaxie par des
chercheurs de la NASA}{Des chercheurs de la NASA ont
découverts, tôt ce matin, ...}{8:42}

    \secondnews{Disques-durs quantiques}{Les disques durs
quantiques seront bientôt disponibles à la vente pour
les particuliers.}{Lors d'un salon international
consacré aux technologies nouvelles ...}{Technologie
}{9:12}

    \thirdnews{Le pop-corn est bon pour le moral}{Une nouvelle
étude vient de montrer que le pop-corn contribue à
la bonne humeur.}{Si vous vous sentez déprimés de
temps en temps, rien de tel qu'un bol de pop-corn
caramélisé ...}{../images/P1000166.jpg}{Diététique
}{12:08}

    \begin{indexblock}{Table des matières}
      \indexitem{Biologie -- Une dangereuse bactérie a été
découverte}{bacterie}

      \indexitem{Informatique -- Barbie geek fait son
apparition}{barbie-geek}
    \end{indexblock}

    \begin{authorblock}
      \includegraphics[width=2cm]{uko-logo.png}

      UKO Publication
    \end{authorblock}

    \begin{weatherblock}{Météo Belgique -- Louvain-la-Neuve}
      \weatheritem{symbols_1.png}{Aujourd'hui}{12}{7}{}
      \weatheritem{symbols_11.png}{Demain}{11}{6}{}
      \weatheritem{symbols_9.png}{Lundi}{8}{5}{}
    \end{weatherblock}
  \end{frontpage}

```



UKO Publication

Friday 3rd December, 2010 || 23:12h



Source : NASA

Découverte d'une nouvelle galaxie par des chercheurs de la NASA

8:42 h | Des chercheurs de la NASA ont découvert, tôt ce matin, une nouvelle galaxie dans une zone jusqu'alors inexplorée de l'Univers. Cette découverte est une grande première, étant donné la technologie qui a été utilisée n'avait jusqu'à présent encore jamais servi pour de telles recherches.

TECHNOLOGIE

Disques durs quantiques

Les disques durs quantiques seront bientôt disponibles à la vente pour les particuliers.

9:12 h | Lors d'un salon international consacré aux technologies de téraoctets de données sur des surfaces inférieures au centimètre carré. Ces derniers devraient être disponibles pour le grand public d'ici la fin de l'année.

Table des matières

Biologie – Une dangereuse bactérie a été découverte p. 2

Informatique – Barbie geek fait son apparition p. 2

DIÉTÉTIQUE

Le pop-corn est bon pour le moral

Une nouvelle étude vient de montrer que le pop-corn contribue à la bonne humeur.

12:08 h | Si vous vous sentez déprimés de temps en temps, rien de tel qu'un bol de pop-corn caramélisé. En effet, une récente étude menée par des diététiciens allemands a montré que la consommation de pop-corn caramélisé contribuait à remonter le moral. Les conclusions de cette sérieuse étude seront publiées le mois prochain dans la revue Food&Health.



Météo Belgique – Louvain-la-Neuve

 <p>Aujourd'hui 12 7 °C</p>	 <p>Demain 11 6 °C</p>	 <p>Lundi 8 5 °C</p>
---	--	--

paper TeX



UKO Publication

 Version PDF gratuite — <http://www.latex-howto.be/>



Code

```

\newsection{Science}

\begin{news}{3}{Une nouvelle souche résistante intrigante}{
  Des chercheurs en microbiologie d'un laboratoire
  japonais ont fait une découverte surprenante qui
  pourrait révolutionner la biologie moderne.}{Biologie}{
  bacterie}
\authorandplace{SC}{Japon}
\image{Bacteria.jpg}{Photo des bactéries prise au
  microscope électronique.}
\timestamp{10:15} Alors qu'ils étaient en train d'étudier
  des levures dans le cadre d'une étude ...

\columntitle{lines}{L'OMS se réunit la semaine prochaine.}

  Cela inquiète bien évidemment ces chercheurs ainsi que
  toute la communauté scientifique ...
\end{news}

\begin{news}{3}{Barbie geek s'installe dans les supermarchés
}{La nouvelle barbie se veut geek ! Depuis hier on peut
voir trôner une barbie informaticienne dans les rayons
des principaux supermarchés du pays.}{Informatique}{
barbie-geek}
\authorandplace{SC}{Belgique}
\timestamp{12:21} Depuis ce matin, on a pu constater l'
  apparition d'une nouvelle barbie dans les principaux
  supermarchés belges ...

\expandedtitle{doublebox}{Cette nouvelle barbie saurait-
  elle séduire des filles et les diriger vers des
  études en informatique ou renforcera-t-elle les à-
  prioris geeks ?}

  Les professionnels de l'informatique se posent néanmoins
  des questions quant à la mauvaise image qu'une telle
  barbie pourrait donner sur le métier d'informaticien
  ...
\end{news}
\end{document}

```


BIOLOGIE

Une nouvelle souche résistante intrigante

Des chercheurs en microbiologie d'un laboratoire japonais ont fait une découverte surprenante qui pourrait révolutionner la biologie moderne.

SC, Japon



Photo des bactéries prise au microscope électronique.

10:15 h | Alors qu'ils étaient en train d'étudier des levures dans le cadre d'une étude sur la production d'insuline artificielle, des

chercheurs japonais ont découvert des bactéries assez particulières. Ces dernières ont la capacité de se reproduire avec d'autres bactéries, en créant ainsi des nouvelles qui ont une capacité de résistance assez forte aux antibiotiques les plus puissants.

Le mécanisme utilisé par ces bactéries est assez particulier et ses détails échappent encore aux chercheurs. Il semblerait qu'une protéine primitive soit impliquée dans le mécanisme de mitose, mais cela doit encore être démontré grâce à des expériences qui sont

actuellement mises en œuvre par le laboratoire.

L'OMS se réunit la semaine prochaine.

Cette affaire inquiète bien évidemment ces chercheurs ainsi que toute la communauté scientifique. Les résultats de ces recherches seront présentés la semaine prochaine lors d'un colloque international qui sera suivi d'une réunion de l'OMS afin de prendre des décisions quant à cette situation. ■

INFORMATIQUE

Barbie geek s'installe dans les supermarchés

La nouvelle barbie se veut geek ! Depuis hier, on peut voir trôner une barbie informaticienne dans les rayons des principaux supermarchés du pays.

SC, Belgique

12:21 h | Depuis ce matin, on a pu constater l'apparition d'une nouvelle barbie dans les principaux supermarchés belges. Elle a des

lunettes et est équipée d'un PC portable ; il s'agit de la barbie geek. À part cela, elle a un look plutôt sexy et porte des vêtements légers au top de la mode. Cette

dernière était déjà disponible aux USA le mois passé, où elle a rencontré un franc succès. Elle est introduite aujourd'hui en Europe.

Cette nouvelle barbie saurait-elle séduire des filles et les diriger vers des études en informatique ou renforcera-t-elle les à-prioris geeks ?

Les professionnels de l'informatique se posent néanmoins des questions quant à la mauvaise image qu'une telle bar-

bie pourrait donner sur le métier d'informaticien, en particulier auprès des filles. La question sera abordée lors de la prochaine réu-

nion de la fédération des professionnels de l'informatique qui aura lieu dans dix jours à Bruxelles. ■

14.4 Construire une présentation avec des slides



Code

```

\documentclass{beamer}

\usetheme{Rochester}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{lmodern}

\usepackage{moreverb}

% Informations sur le document
\title{Construire une présentation avec \LaTeX}
\subtitle{Le package Beamer}
\author{Sébastien Combéfis\inst{1}}
\institute{\inst{1}UK0}
\date{27 novembre 2010}

\begin{document}
  \frame[plain]{
    \titlepage
  }

  \section{Définir un transparent}
  \begin{frame}
    \frametitle{Plan}
    \tableofcontents[currentsection]
  \end{frame}

  \begin{frame}[containsverbatim]
    \frametitle{Définir un transparent}

    \begin{definition}[Beamer]
      \alert{Beamer} est package qui permet de réaliser des
      slides en \LaTeX. Un slide est représenté par l'
      environnement frame.
    \end{definition}

    \begin{block}{Code (document minimal)}
      \begin{verbatimtab}[3]
\begin{frame}
  \frametitle{Titre}

  Contenu
\end{frame}
      \end{verbatimtab}
    \end{block}
  \end{frame}

```

The image displays four slides from a Beamer presentation, arranged in a 2x2 grid. Each slide has a navigation bar at the bottom right.

Slide 1 (Top Left): Titled "Construire une présentation avec \LaTeX ". Subtitle: "Le package Beamer". Author: "Sébastien Combéfis¹". Date: "27 novembre 2010".

Slide 2 (Top Right): Titled "Plan". It lists three items: "1. Définir un transparent", "2. Modifier le style", and "3. Conclusion".

Slide 3 (Bottom Left): Titled "Définir un transparent". It contains a definition: "Définition (Beamer) **Beamer** est un package qui permet de réaliser des slides en \LaTeX . Un slide est représenté par l'environnement `frame`." Below this is a code block for a minimal document structure:

```
Code (document minimal)
\begin{frame}
  \frametitle{Titre}
  Contenu
\end{frame}
```

Slide 4 (Bottom Right): Titled "Modifier le style". Subtitle: "Globalement, pour tout le document". It lists "Plusieurs possibilités :" followed by a bulleted list: "■ Style pré-défini", "■ Ré-définitions de commandes", and "■ Définition d'un nouveau style".

**Code**

```

\section{Modifier le style}
\begin{frame}
  \frametitle{Modifier le style}
  \only<1>{\framesubtitle{Globalement pour tout le document}}
  \only<2>{\framesubtitle{De manière locale}}

  \only<1>{
    Plusieurs possibilités :

    \begin{itemize}
      \item Style prédéfini
      \item Redéfinitions de commandes
      \item Définition d'un nouveau style
    \end{itemize}
  }
  \only<2>{
    Il suffit d'utiliser les commandes habituelles de \
    LaTeX.
  }
\end{frame}

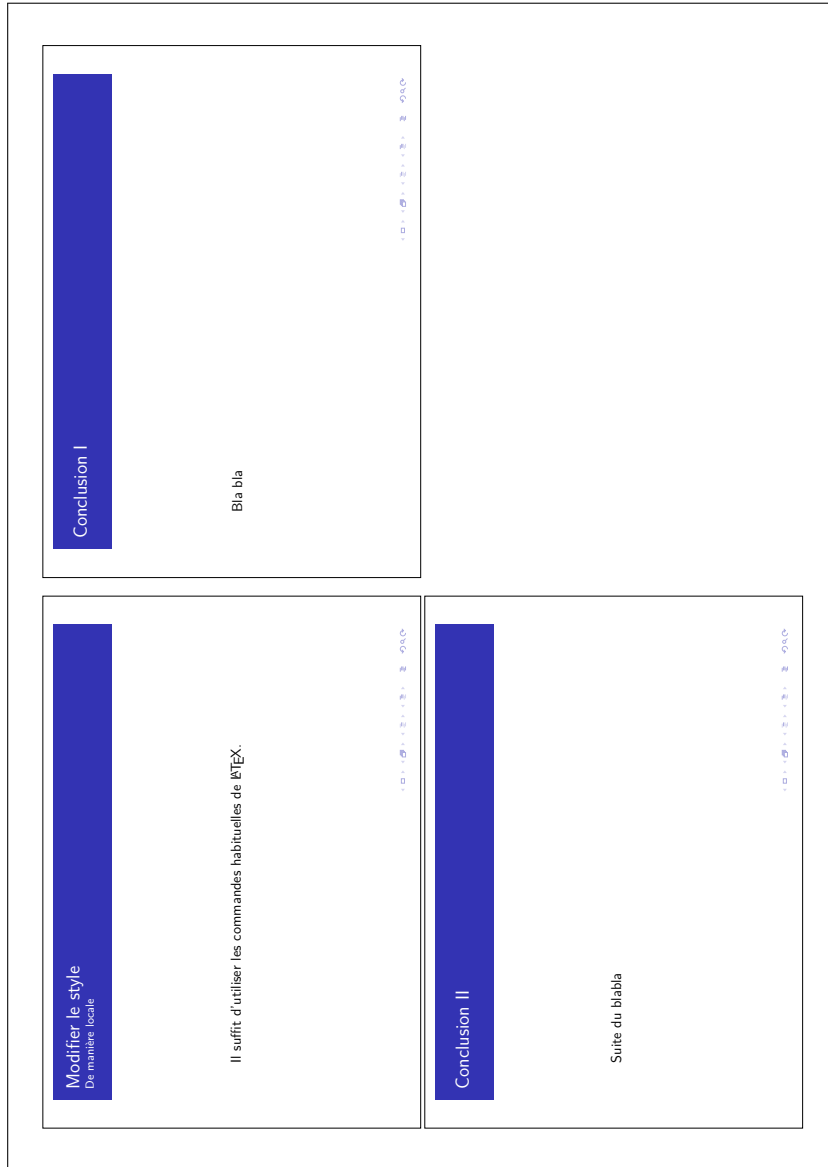
\section{Conclusion}
\begin{frame}[allowframebreaks]
  \frametitle{Conclusion}

  Bla bla

  \framebreak

  Suite du blabla
\end{frame}
\end{document}

```



ANNEXE A

Utiliser L^AT_EX sur mon ordinateur

Nous n'allons pas ici détailler toutes les procédures d'installation de L^AT_EX pour les différents systèmes d'exploitation existants. Des informations détaillées sont données sur <http://www.latex-howto.be>.

Afin de pouvoir rédiger des documents avec L^AT_EX, il faut deux éléments : une distribution qui contient tous les outils permettant de compiler un fichier `.tex` et de produire un document final et un environnement d'édition permettant de rédiger le code L^AT_EX avec l'aide d'outils intégrés.

Voici les combinaisons les plus courantes de distribution et d'éditeur, accompagné des numéros des versions les plus récentes aujourd'hui (décembre 2010) :

- Windows : MikTeX 2.9 et TeXnicCenter 2.0 Alpha 3
- Linux : TeXLive 2010 et Kile 2.0.3
- Mac OS X : MacTeX 2010 (TeXLive 2010 et TeXShop 2.37)

Il existe également des éditeurs fonctionnant sur plusieurs plateformes différentes. Parmi ceux-ci, citons Texmaker et TeXworks.

ANNEXE B

Quelques liens utiles

\LaTeX HowTo



Site web officiel du livre qui comporte, outre une version électronique de ce livre, de nombreuses autres informations, des liens utiles et ressources téléchargeables . . .

<http://www.latex-howto.be/>

UKO



Ce site propose un tutoriel complet sur \LaTeX , en français. Chaque chapitre est accompagné d'un projet pratique vous permettant de plonger dans \LaTeX tout en étant guidé et en vous permettant de découvrir les outils à votre disposition.

<http://www.ukonline.be/programmation/latex/>

CTAN



Ce site rassemble tous les packages \LaTeX qui ont été développés. Vous y trouverez, outre le code source des packages, leur documentation complète.

<http://www.ctan.org/>



TeXample

Ce site propose une collection d'exemples de dessins \LaTeX réalisés avec le package *TikZ/PGF*. Vous y trouverez plus de 200 exemples sur lesquels vous pourrez vous baser pour vos propres créations.

<http://www.texample.net/>



TeXblog

Ce blog est la référence en ce qui concerne la réalisation de documents avec \TeX et \LaTeX . Parcourir ce blog vous permettra d'apprendre énormément de choses.

<http://www.texblog.net/>



PracTeX

Ce site propose un journal en ligne consacré à \LaTeX . Les différents articles publiés abordent divers aspects de \LaTeX et sont de très bonne qualité. Abonnez-vous sans hésiter au flux RSS.

<http://www.tug.org/pracjournal/>

Les captures d'écran ont été réalisées le 4 décembre 2010 sur un Mac sous Mac OS X 10.6.5 avec le logiciel Paparazzi! 0.4.3 (<http://derailer.org/paparazzi/>).

Liste des packages

Cette section reprend la liste des packages présentés dans ce livre. Pour chaque package, vous trouverez également la version qui est décrite et a été utilisée dans ce livre.

<code>algorithm2e</code> (14/12/2009)	140
<code>amsthm</code> (08/2004)	115
<code>arydshln</code> (31/08/2004)	77
<code>bclogo</code> (10/09/2009)	100
<code>beamer</code> (12/07/2010)	184
<code>booktabs</code> (14/04/2005)	87
<code>ccaption</code> (02/09/2009)	122
<code>changebar</code> (19/09/2005)	132
<code>cleveref</code> (24/06/2010)	119
<code>color</code> (14/11/2005)	28
<code>colortbl</code> (13/02/2001)	79
<code>coolstr</code> (10/09/2009)	164
<code>cwpuzzle</code> (13/09/2009)	169
<code>datatool</code> (15/09/2009)	90
<code>dcolumn</code> (28/05/2001)	74
<code>delarray</code> (14/03/1994)	112
<code>desclist</code> (23/11/2004)	51
<code>ean13isbn</code> (20/06/2008)	171
<code>easylist</code> (28/02/2010)	50
<code>endnotes</code> (15/01/2003)	130

enumerate (05/03/1999)	43
enumitem (18/05/2009)	43
fancybox (15/05/2010)	98
fancyhdr (02/03/2004)	125
float (08/11/2001)	166
fncychap (08/11/1997)	13
footmisc (18/09/2009)	128
framed (04/10/2007)	101
fullpage	37
fvrb-ex (16/05/2010)	136
geometry (13/07/2010)	38
hyperref (01/2010)	34
ifthen (26/05/2001)	162
lastpage (24/09/2010)	121
layout (25/09/2000)	35
lcg (10/09/2008)	172
listings (22/02/2007)	136
longtable (01/02/2004)	83
makecell (03/08/2009)	77
manyfoot (11/09/2005)	129
marginnote (05/01/2010)	132
minitoc (16/07/2008)	16
moderncv (24/05/2010)	178
moreverb (03/06/2008)	134
multido (14/05/2010)	163
multicol (25/09/2009)	31
multirow (27/02/2010)	70
nccrules (13/05/2005)	22
numprint (17/02/2008)	31
papertex (30/06/2010)	180
paralist (09/05/2005)	45
pdfpages (12/01/2010)	39
photo (15/07/2004)	63
pstricks (25/07/2003)	158
pst-barcode (27/08/2009)	171
qtree (12/12/2008)	153

<code>rotating</code> (26/01/2010)	60
<code>savetrees</code> (03/06/2010)	37
<code>sectsty</code> (03/06/2010)	12
<code>shapepar</code> (09/2006)	30
<code>sectsty</code> (25/02/2002)	12
<code>shortvrb</code> (25/09/2009)	134
<code>sidecap</code> (06/06/2003)	57
<code>siunitx</code> (15/07/2010)	32
<code>slashbox</code> (27/05/1993)	71
<code>spreadtab</code> (06/06/2010)	89
<code>subfigure</code> (15/03/2005)	61
<code>tablists</code> (04/08/2009)	51
<code>tabularx</code> (07/01/1999)	83
<code>tipa</code> (25/03/2004)	173
<code>titlepic</code> (05/08/2008)	6
<code>titlesec</code> (12/08/2007)	10
<code>todonotes</code> (25/12/2009)	131
<code>ulem</code> (21/01/2010)	26
<code>url</code> (02/01/2010)	33
<code>varioref</code> (13/09/2009)	119
<code>verbatim</code> (12/03/2001)	135
<code>vmargin</code> (15/07/2004)	37
<code>watermark</code> (09/12/2004)	172
<code>wrapfig</code> (31/01/2003)	61
<code>xcolor</code> (21/01/2007)	80
<code>xstring</code> (28/03/2010)	164

Index

A

`\addcontentsline`, 15
`\addto`, 120
`\addtocounter`, 165
`\addtolength`, 165
`\ADLactivate`, 76
`ADLinactivate`, 76
`algorithm`, 138
`algorithm2e`, 138
`\allsectionsfont`, 12
`\Alph`, 165
`\alph`, 40, 165
`amsthm`, 113
`\AND`, 160
`\and`, 6
`\appendix`, 14
`\arabic`, 165
`array`, 70, 110
`\arrayrulewidth`, 74
`\arraystretch`, 79
`arydshln`, 75
`\author`, 6

B

`\backmatter`, 14
`\backslashashbox`, 69
`\baselineskip`, 22
`bclogo`, 98
`beamer`, 182

`\bf`, 24
`bf`, 25
`\bfseries`, 24
`bfseries`, 25
`\bibitem`, 143
`\bibliography`, 145
`\bibliographystyle`, 145
`\BibTeX`, 144
`\Big`, 109
`\big`, 109
`\Bigg`, 109
`\bigg`, 109
`\bigskip`, 21
`\bmod`, 108
`booktabs`, 85
`\bottomrule`, 85
`\boxput`, 97

C

`\caption`, 54, 65, 116, 119, 163
`caption`, 55
`\captiondelim`, 120
`\captionnamefont`, 120
`\captionof`, 119
`captionof`, 55
`\captionsfrench`, 120
`\captiontitlefont`, 120
`\cbcolor`, 129
`\cbdelete`, 129

`\cbend`, 129
`\cbstart`, 129
`ccaption`, 120
`\cdashline`, 75
`\cellcolor`, 77
`center`, 20
`\centering`, 19, 54, 65
`\cfoot`, 122
`changebar`, 129
`\chapter`, 10
`\chapterfont`, 12
`\chaptermark`, 121
`\chaptername`, 12
`\chaptertitlename`, 12
`\chead`, 122
`\circlepar`, 30
`\cite`, 144
`cleveref`, 117
`\cline`, 64
`Clue`, 168
`\cmidrule`, 85
`\cntprint`, 31
`\color`, 27
`color`, 27
`\colorbox`, 97
`colortbl`, 77
`\columnbreak`, 31
`\columnsep`, 31
`\columnseprule`, 31
`coolstr`, 162
`\cornersize`, 96
`\cref`, 117
`\crefrange`, 117

cpuzzle, 167

D

`\dashlinedash`, 76
`\dashlinegap`, 76
`\dashrule`, 23
`\dashrulefill`, 22
 datatool, 88
`\date`, 6
 dcolumn, 72
`\DeclareGraphicsExtensions`,
 60
`\DeclareMathOperator`, 108
`\DeclareNewFootnote`, 126
`\definecolor`, 28
 delarray, 110
`\depth`, 91
`desclist`, 49
`\diamondpar`, 30
 document, 5
`\documentclass`, 5
`\dominilof`, 17
`\dominilot`, 17
`\dominitoc`, 17
`\dotfill`, 22
`\doublebox`, 96
`\doublerulesep`, 74
`\DTLdisplaydb`, 88
`\DTLloaddb`, 88

E

ean13isbn, 169
`\EANisbn`, 169
`easylist`, 48
`\emph`, 25
`\endfirsthead`, 82
`\endfoot`, 82
`\endhead`, 82
`\endlastfoot`, 82
`\endnote`, 127
`\endnotemark`, 127
 endnotes, 127
`\endnotetext`, 127
`enumerate`, 40, 41
`\enumitem`, 41
`eqnarray`, 111

`\equal`, 161
`\extracolsep`, 80

F

`\fancybox`, 96
 fancybox, 96
`\fancyfoot`, 123
 fancyhdr, 122
`\fancyhead`, 123
`\fancyhf`, 123
`\fancypagestyle`, 123
`\fbox`, 90
`\fboxrule`, 90
`\fboxsep`, 91
`\fcolorbox`, 97
 figure, 54
`\figurename`, 120
`\fill`, 21
`\firstline`, 83
 fix-cm, 27
 float, 164
`\floatname`, 164
`\floatstyle`, 164
`flushleft`, 20
`flushright`, 20
 fncychap, 13
`\fnsymbol`, 125, 165
`\fontsize`, 27
 footmisc, 125
`\footnote`, 124
`\footnotemark`, 125
`\footnoterule`, 126
`\footnotesize`, 27
`\footnotetext`, 125
`\footrulewidth`, 122
`\footskip`, 34
`\framebox`, 90
 framed, 99
`\frontmatter`, 14
 fullpage, 36
 fvrb-ex, 134

G

geometry, 36, 37
`\graphicspath`, 60
 graphicx, 51, 93, 94, 97

H

`\hdashline`, 75
`\headheight`, 34
`\headrulewidth`, 122
`\headsep`, 34
`\heartpar`, 30
`\height`, 91
`\hexagonpar`, 30
`\hfill`, 21
`\hline`, 64
`\hrulefill`, 22
`\hspace`, 20
`\Huge`, 27
`\huge`, 27

I

`\IfBeginWith`, 162
`\IfDecimal`, 162
`\IfEndWith`, 162
`\IfInteger`, 162
`\IfStrEq`, 162
`\IfStrEqCase`, 162
`\IfSubStr`, 162
 ifthen, 160
`\ifthenelse`, 160
`\includegraphics`, 51
`inparaenum`, 43
`\isdecimal`, 162
`\isint`, 162
`\isnumeric`, 162
`\isodd`, 160
`\isundefined`, 161
`\it`, 24
 it, 25
`\item`, 39
`\itemindent`, 47
`itemize`, 39
`\itemsep`, 42, 47
`\itshape`, 24
`itshape`, 25

L

`\label`, 115
`\labelenumi`, 48
`\labelenumii`, 48
`\labelenumiii`, 48
`\labelenumiv`, 48

`\labelformat`, 118
`\labelitemi`, 39, 48
`\labelitemii`, 48
`\labelitemiii`, 48
`\labelitemiv`, 48
`\labelsep`, 47
`\labelwidth`, 47
`\LARGE`, 27
`\Large`, 27
`\large`, 27
`\lastline`, 83
`\layout`, 34
`lcg`, 170
`\left`, 109
`leftbar`, 100
`\leftmargin`, 47
`\leftmark`, 121
`\lengthtest`, 161
`\lenprint`, 31
`letter`, 174
`\lfoot`, 122
`\lhead`, 122
`\limits`, 106
`list`, 45
`listing`, 132
`listingcont`, 133
`\listinginput`, 133
`listings`, 134
`\listof`, 164
`\listoffigures`, 15
`\listofphotos`, 61
`\listoftables`, 15
`\listoftodos`, 128
`\listparindent`, 47
`\ListProperties`, 48
`longtable`, 81
`\lstineline`, 135
`\lstinputlisting`, 135
`lstlisting`, 134
`\lstlistoflistings`, 137
`\lstset`, 135

M

`\marginpar`, 128
`\mainmatter`, 14
`\makebox`, 89
`makecell`, 75
`\MakeShortVerb`, 132

`\maketitle`, 6
`manyfoot`, 126
`\marginnote`, 129
`marginnote`, 129
`\marginpar`, 58
`\marginparsep`, 34
`\marginparwidth`, 34, 128
`\markboth`, 121
`\markoverwith`, 26
`\markright`, 121
`\mathbb`, 105
`\mathcal`, 105
`\mathfrak`, 105
`\mbox`, 89
`\mboxfill`, 127
`\mdseries`, 24
`\medskip`, 21
`\midrule`, 85
`\minilof`, 17
`\minilot`, 17
`minipage`, 57
`\minitoc`, 17
`minitoc`, 16
`\mod`, 108
`moderncv`, 176
`moreverb`, 132, 133
`multicol`, 30
`multicols`, 30
`\multicolumn`, 67
`\multido`, 161
`multido`, 161
`\multidocount`, 162
`\multidostop`, 162
`\multirow`, 68
`multirow`, 68

N

`nccrules`, 22, 126
`\newcolumntype`, 72
`\newcommand`, 159
`\newcounter`, 165
`\newenvironment`, 159
`\newfloat`, 164
`\newfootnotedashrule`, 127
`\newfootnoterule`, 126
`\newlength`, 165
`\newtheorem`, 112
`\nocite`, 145

`\nolimits`, 106
`\nonumber`, 112
`\normalem`, 26
`\normalsize`, 27
`\NOT`, 160
`\numprint`, 31
`numprint`, 31
`\nutpar`, 30

O

`\oddsidemargin`, 34
`\operatorname`, 108
`\OR`, 160
`outline`, 48
`\Ovalbox`, 96
`\ovalbox`, 96
`\overbrace`, 106

P

`\pagenumbering`, 24
`\pageref`, 115
`\pagestyle`, 23
`\paperheight`, 34
`papertex`, 178
`\paperwidth`, 34
`\paragraph`, 10
`paralist`, 43
`\parbox`, 92
`\parsep`, 47
`\parskip`, 47
`\part`, 10
`\partopsep`, 47
`Photo`, 61
`photo`, 61
`\pmod`, 108
`\postcaption`, 120
`\precaption`, 120
`proof`, 113
`psbarcode`, 169
`pst-barcode`, 169
`pstricks`, 156
`\putphoto`, 61
`Puzzle`, 167
`PuzzleClues`, 168
`\PuzzleSolution`, 168
`\PuzzleUnsolved`, 168

Q

`\qedhere`, 113
`\qframesubtree`, 152
`\qlabelhook`, 152
`\qleafhook`, 152
`\qproof`, 152
`qtree`, 151
`\qtreefinalhook`, 152
`\qtreeinithook`, 152
`quotation`, 29
`quote`, 29

R

`\raggedleft`, 19
`\raggedright`, 19
`\raisebox`, 56, 93
`\rand`, 170
`\Ref`, 118
`\ref`, 115
`\reflectbox`, 94
`\renewcommand`, 160
`\renewenvironment`, 160
`\resizebox`, 94
`\restylefloat`, 164
`\rfoot`, 122
`\rhead`, 122
`\right`, 109
`\rightmargin`, 47
`\rightmark`, 121
`\rm`, 24
`\rmfamily`, 24
`\Roman`, 165
`\roman`, 165
`\rotatebox`, 69, 94
`rotating`, 58, 84
`\rowcolor`, 77
`\rowcolors`, 78
`\rule`, 80

S

`savetrees`, 36
`\sc`, 24
`sc`, 25
`\scalebox`, 93
`SCfigure`, 55

`\scriptsize`, 27
`\scshape`, 24
`scshape`, 25
`\section`, 10
`\sectionfont`, 12
`\sectionmark`, 121
`\sectionrule`, 12
`sectsty`, 12
`\selectfont`, 27
`\SelectFootnoteRule`, 126
`\setcounter`, 165
`\setdescription`, 43
`\setenumerate`, 43
`\setitemize`, 43
`\setlength`, 165
`\setlist`, 43
`\setmargins`, 36
`\setpapersize`, 36
`\settodepth`, 166
`\settoheight`, 166
`\settowidth`, 166
`\sf`, 24
`sf`, 25
`\sffamily`, 24
`sffamily`, 25
`shaded`, 100
`\shadowsize`, 96
`shapepar`, 30
`\shorttableofcontents`, 16
`\shorttoc`, 16
`shorttoc`, 16
`shortvrb`, 132
`SideBySideExample`, 134
`sidecap`, 55
`sidewaysfigure`, 58
`sidewaystable`, 84
`siunitx`, 32
`\sl`, 24
`sl`, 25
`\slashbox`, 69
`slashbox`, 69
`\slshape`, 24
`slshape`, 25
`\small`, 27
`\smallskip`, 21
`snugshade`, 100
`spreadtab`, 87
`\squarepar`, 30
`\stackrel`, 108
`\starpar`, 30
`\StrBefore`, 163
`\StrBehind`, 163
`\StrBetween`, 163
`\StrChar`, 163
`\StrCount`, 163
`\StrDel`, 163
`\StrLeft`, 163
`\StrLen`, 163
`\StrMid`, 163
`\StrPosition`, 163
`\StrRight`, 163
`\StrSubstitute`, 163
`\subfigure`, 59
`subfigure`, 59, 85, 119
`\subparagraph`, 10
`\subref`, 119
`\subsection`, 10
`\substr`, 162
`\subsubsection`, 10
`\subtable`, 85
`\sum`, 106

T

`tabenum`, 49
`\table`, 65
`\tablename`, 120
`\tableofcontents`, 15
`tablists`, 49
`tabular`, 63
`tabular*`, 80
`\tabularnewline`, 73
`tabularx`, 81
`\textbf`, 25
`\textcolor`, 27
`\textheight`, 34
`\textipa`, 171
`\textit`, 25
`\textnormal`, 25
`\textsc`, 25
`\textsf`, 25
`\textsl`, 25
`\texttt`, 25
`\textwidth`, 34
`\the`, 165
`thebibliography`, 143
`\thechapter`, 121
`\theendnotes`, 127
`\theenumi`, 40
`\thefootnote`, 125

- `\thepage`, 121
 - `\thesection`, 121
 - `\thetitle`, 11
 - `\thicklines`, 96
 - `\thinlines`, 96
 - `\thispagestyle`, 23
 - `\tiny`, 27
 - `tipa`, 171
 - `\title`, 6
 - `\titleformat`, 10
 - `\titlelabel`, 10
 - `titlepage`, 9
 - `\titlepic`, 6
 - `titlepic`, 6
 - `titlesec`, 10
 - `\titlespacing`, 10
 - `\today`, 6
 - `\todo`, 128
 - `todonotes`, 128
 - `\topmargin`, 34
 - `\toprule`, 85
 - `\topsep`, 47
 - `\totalheight`, 91
 - `\Tree`, 151
 - `\tt`, 24
 - `tt`, 25
 - `\ttfamily`, 24
 - `ttfamily`, 25
 - `turn`, 84
- ## U
- `ulem`, 25
 - `\uline`, 25
 - `\underbrace`, 106
 - `\underline`, 25
 - `\upshape`, 24
 - `\url`, 33
 - `url`, 33
 - `\urldef`, 33
 - `\urlstyle`, 33
- ## V
- `varioref`, 117
 - `\verb`, 131
 - `verbatim`, 131, 133
 - `\verbatiminput`, 133
 - `verbatimtab`, 132
 - `\verbatimtabinput`, 133
 - `\verbatimtabsize`, 132
 - `verse`, 29
 - `\vfill`, 21
 - `vmargin`, 36
 - `\vpageref`, 117
 - `\vpagerefrange`, 117
 - `\vref`, 117
 - `\vrefrange`, 117
 - `\vrule`, 74
 - `\vspace`, 20
- ## W
- `watermark`, 170
 - `\width`, 91
 - `wrapfig`, 59
 - `wrapfigure`, 59
- ## X
- `\Xcline`, 75
 - `xcolor`, 78
 - `\Xhline`, 75
 - `xstring`, 162