
Initiation à **LATEX**

avril–mai 2012

Thierry MASSON

- ✉ CPT (UMR 7332)
Case 907 - Campus de Luminy
F-13288 Marseille Cedex 9
- ✉ thierry.masson@cpt.univ-mrs.fr
✉ science.thilucmic.fr

Version du 17 avril 2012

Au programme...

1	Découverte de l'univers \LaTeX	3
2	Installation et maintenance	7
3	Découverte du langage de \LaTeX	13
4	Structure du document source	21
5	Des <i>packages</i> essentiels	27
6	Les mathématiques	37
7	Un peu de technique	55
8	Les listes	63
9	Les tableaux	65
10	Les couleurs	67
11	Les images et les graphiques	69
12	Les flottants	73
13	La bibliographie	75
14	La chimie	77
15	Le style du document	81
16	La classe beamer	85
	URLs des liens cités dans le texte	99

De la documentation

Livres : quelques livres très bien faits.

L^AT_EX Companion, F. Mittelbach et M. Goossens, Addison-Wesley 2004.

L^AT_EX, Apprentissage, guide et référence, B. Desgraupes, Vuibert 2003.

Guide to L^AT_EX, H. Kopka et P. Daly, Addison-Wesley 2003.

L^AT_EX pour l'impatient, W. Appel, C. Chavalier, E. Cornet, S. Desreux, H&K, 2009.

L^AT_EX, l'essentiel, D. Bitouzé, J.-C. Charpentier, Pearson, 2010.

Documents électroniques : quelques PDF à lire.

The Not So Short Introduction to L^AT_EX¹, un fascicule à conserver sous la main.

Comprehensive L^AT_EX symbol list², la référence pour trouver un symbole.

Sites internet : quelques liens pour rester à jour.

L^AT_EX Reference Manual³, la référence de toutes les commandes...

Comprehensive T_EX Archive Network⁴, des *packages* bien classés...

L^AT_EX Wikibook⁵, documentation collective sur L^AT_EX, disponible aussi en PDF.

Detexify² – L^AT_EX symbol classifier⁶, retrouver un symbole en le dessinant.

The PracTeX Journal⁷, un journal réalisé par des utilisateurs de L^AT_EX.

L^AT_EX Community⁸ et **T_EX blog⁹**, pour se tenir au courant de l'actualité.

Documentation en ligne sur L^AT_EX¹⁰, ma propre page web de liens.

Forums : pour poser des questions après le cours.

L^AT_EX Community : le forum¹¹, un forum très réactif et très informatif...

1 – Découverte de l'univers **LATEX**

Logiciels de composition de textes

Il existe plusieurs catégories de logiciels pour générer du texte.

Éditeur de texte : C'est un logiciel destiné à la création et à l'édition de fichiers textes sans mise en forme des polices, des couleurs, de justification...

Les fichiers textes produits sont légers et exploitables par d'autres éditeurs de texte.

→ Coloration syntaxique de langages informatiques, y compris **LATEX**.

→ Interface avec d'autres outils : dictionnaires, compilateurs...

Bloc-notes¹² (Windows), **Emacs**¹³ (Linux/Windows), **Pico**¹⁴ (Linux/Windows), **TextEdit**¹⁵ (Mac OS)...

Traitement de texte : C'est un logiciel qui produit une composition du texte (au sens typographique) à partir des données textuelles fournies par l'utilisateur.

Le rendu est composé en direct lors de la frappe.

→ Logiciel **WYSIWYG** (*What You See Is What You Get*).

Le format de sauvegarde retient les informations de composition : mise en forme, couleurs, polices... → Les fichiers sont plus gros.

→ Les fichiers sont souvent moins exploitables par d'autres logiciels.

Microsoft Word¹⁶ (Windows), **OpenOffice**¹⁷ (Linux/Windows/Mac OS)...

Logiciels de composition de textes (suite)

Logiciel de mise en page (PAO)¹⁸ : “Publication Assistée par Ordinateur”.

Composition de très grande qualité typographique à partir de textes et d'images.

→ Livres, magazines, brochures...

Notions de maquette, calques, chaînage de textes, habillage d'images...

Logiciels haut de gamme, donc souvent chers.

→ Apprentissage difficile, plutôt réservés à des professionnels.

InDesign¹⁹ (Windows/Mac OS), **QuarkXPress**²⁰ (Windows/Mac OS), **Scribus**²¹ (Linux/Windows/Mac OS)...

TEX : Ce n'est pas un “traitement de texte”, mais un “traitement de document”.

Document source → analyse/compilation → document final.

Peut rivaliser avec les logiciels de PAO.

Flexibilité énorme (langage de programmation).

Mise en œuvre très différente des autres logiciels.

LATEX se charge automatiquement de nombreux aspects et détails typographiques.

Même logique que les navigateurs **WEB** : fichier texte source en **HTML**²².

Quelques repères

T_EX est à la fois

- un langage de programmation,
- un **moteur de compilation** appelé **tex**²³.

Il est accompagné d'un ensemble de macros, nommé **PLAIN T_EX**, regroupées sous forme d'un **format**.

LAT_EX²⁴ est un autre ensemble de macros qui remplace **PLAIN T_EX**.

- **LAT_EX** est un format pour le moteur **tex**.
- La compilation du texte utilise toujours le moteur **tex**.
- **LAT_EX** est aussi un langage de *description sémantique* du texte.
- Séparation du fond sémantique et de la forme visuelle.
- **LAT_EX** introduit la notion de *packages* (“modules” en français).
- Ajout de fonctionnalités de façon modulaire.
- Extensions très simples et très nombreuses des fonctionnalités de **LAT_EX**.

Le moteur **tex** d'origine crée un fichier **DVI**²⁵.

Aujourd'hui, il est remplacé par le moteur **pdftex** qui crée un fichier **PDF**²⁶.

Exécuter **pdflatex**, c'est lancer **pdftex** avec le format **LAT_EX**.

Le logiciel **tex** est accompagné de nombreux autres utilitaires pour gérer l'indexation d'un document, la bibliographie, la création de graphiques...

Les avantages de **LAT_EX**

- Le fichier source est du texte.
 - Taille très petite : quelques Mo pour un livre de 600 pages.
 - Très grande portabilité (tous les systèmes d'exploitation).
 - Le texte peut être généré par un logiciel tiers (insertion dans un flux automatisé).
- Logiciels gratuits, ouverts et stabilisés → pérennité des documents.
- Typographie de très grande qualité due au moteur **tex** : césures, ligatures...
- Possibilité de programmation : macros personnelles, mise en page, aspect des éléments du texte, automatisations diverses...
- Séparation du fond et de la forme.
 - Changement de style aisément.
 - Gestion automatique de nombreux éléments du document (table des matières, références croisées, bibliographie...).
- Capacité à gérer des gros documents complexes.
- Gestion aisée de documents écrits dans des langues et dialectes différents.
 - Très utilisé en linguistique.
- Nombreuses extensions sous forme de *packages*.
- Écosystème riche : **makeindex**, **bibtex**, **metapost**...
- *Last but not least* : excellente composition des formules mathématiques !

.....

Les inconvénients de **LATEX**

- Installation et maintenance peu aisées, mais ça s'améliore (utilitaire **tlmgr**).
- Non **WYSIWYG** (*What You See Is What You Get*).
Le document de départ est loin visuellement du document final.
Les environnements de travail d'aujourd'hui pallient largement à ce problème.
- Extrême rigueur requise pour la composition (en fait, c'est un avantage).
- Limitations typographiques sur certains points précis (forme des paragraphes...).
Les logiciels de **PAO** font mieux.
- Pas assez de polices d'écriture. Double origine :
 - **TEX** fonctionne avec ses propres polices, dans un format très spécifique.
 - Peu de polices complètes existent pour les mathématiques.
- Le langage de programmation de **TEX** est différent des langages de programmation usuels : il est difficile à maîtriser et très déroutant (consulter les forums !).
- Limitations intrinsèques dues à une programmation ancienne (ordinateurs avec peu de ressources) sur le nombres de fontes en mathématiques, le nombre de registres...
- Blocages “technologiques” à cause de l'écosystème et du succès.

2 – Installation et maintenance

L'installation de **TEX**

Pour utiliser **LAT_EX** il faut installer deux types de ressources :

1. Des logiciels (“moteurs”) **tex**, **pdftex**, **bibtex**, **makeindex**... et des *dizaines de milliers* de fichiers regroupés à différents endroits sur la machine (classes, *packages*, polices, documentation...).

Il existe pour ça des distributions prêtes à l’emploi et à l’installation.

TeXLive²⁷ (🐧macOSWindows) est la référence.

2. Une interface conviviale qui contient :

- un éditeur de texte pour écrire le fichier source ;
- un moyen de lancer la compilation (terminal intégré, menu ou bouton) ;
- un logiciel (intégré ou non) de visualisation du **DVI**, du **PS** ou du **PDF** produit ;
- des fonctionnalités diverses qui permettent une meilleure productivité : aide à la frappe (auto-complétion), correction orthographique, aller-retour “source”/“résultat visuel”, interface avec d’autres moteurs...

Il existe pour ça des logiciels intégrés :

TeXworks²⁸ (🐧macOSWindows), **TeXShop**²⁹ (macOS), **TeXMaker**³⁰ (🐧macOSWindows), **emacs**³¹ (🐧), **Aquamacs**³² (macOS), **TeXnicCenter**³³ (Windows), **WinEdit**³⁴ (Windows)...

TeXworks est un très bon choix.

Liste de logiciels pour installer **LAT_EX**

🐧 Installation fondamentale : **TeXLive**²⁷.

Logiciels d’interfaces : **TeXworks**²⁸, **TeXMaker**³⁰, **emacs**³¹ (+ **AucTeX**³⁵).

Utilitaires divers : **JabRef**³⁶, **KBibTeX**³⁷ (gestion bibtex), **Aspell**³⁸ (correcteur orthographique).

Windows Installation fondamentale : **TeXLive**²⁷.

Logiciels d’interfaces : **TeXworks**²⁸, **TeXMaker**³⁰, **TeXnicCenter**³³, **WinEdit**³⁴, **emacs**³¹ (+ **AucTeX**³⁵)

Utilitaires divers : **JabRef**³⁶ (gestion bibtex).

macOS Installation fondamentale : **MacTeX**³⁹.

Logiciels d’interfaces : **TeXShop**²⁹, **TeXworks**²⁸, **TeXMaker**³⁰, **Aquamacs**³² (+ **AucTeX**³⁵).

Utilitaires divers : **BibDesk**⁴⁰, **JabRef**³⁶ (gestion bibtex), **TeX Live Utility**⁴¹ (gestion **TeXLive**), **CocoAspell**⁴² (correcteur orthographique).

.....
Les types de fichiers de l'univers **LATEX**

LATEX utilise et produit un grand nombre de types de fichiers.

Fichiers visibles côté utilisateur, essentiellement dans le dossier de travail :

Fichiers créés par l'utilisateur

- .tex Fichier source **TEX** ou **LATEX** qui est compilé par le moteur **tex** ou **pdftex**.
- .bib Base de donnée bibliographique, utilisée par **bibtex**.

Fichiers générés par **LATEX**

- .log Fichier donnant les détails de ce qui s'est passé lors de la compilation.
- .aux Informations sur le document, utilisé par une seconde compilation.
- .toc Informations pour la table des matières.
- .lof Informations pour la liste des figures.
- .lot Informations pour la liste des tableaux.
- .idx Informations pour créer un index.
- .dvi Fichier de sortie produit par le moteur **tex**.
- .pdf Fichier de sortie produit par le moteur **pdftex**.

Fichiers générés par d'autres moteurs

- .ind Fichier généré à partir de .idx par **makeindex**.
- .ilg Fichier log du moteur **makeindex**.
- .bbl Fichier généré à partir de .aux et .bib par **bibtex**.
- .blg Fichier log du moteur **bibtex**.

.....
Les types de fichiers de l'univers **LATEX** (suite)

Fichiers "cachés", sur lesquels l'utilisateur n'intervient pas (ou très peu) :

Fichiers utilisés par **LATEX**

- .cls Fichier de "classe" appelé par **\documentclass[-]{-}**.
- .clo Fichier d'option de "classe" appelé par **\documentclass[-]{-}**.
- .sty Fichier de "style" (*packages*) appelé par **\usepackage[-]{-}**.
- .cfg Fichier de configuration de certains *packages* (**graphics**, **hyperref**...).
- .def Fichier de définition associés à certains *packages* (**inputenc**, **babel**...).
- .fd Fichier de description des polices d'écriture.
- .vf Fichier de polices virtuelles.
- .tfm Fichier de métrique des fontes.
- .dtx Fichier de distribution des *packages* (produit les .sty, .def, .pdf...).
- .ins Fichier gérant l'installation d'un *package* par une compilation **LATEX**.

Fichiers utilisés par d'autres moteurs

- .map Fichier de "mappage" entre les noms des polices (**pdftex**, **dvips**...).
- .enc Fichier d'encodage des polices (**pdftex**, **dvips**...).
- .pfb Fichier des polices vectorielles (**pdftex**, **dvips**...).
- .bst Fichier de style bibliographique (**bibtex**).
- .ist Fichier de style d'index (**makeindex**).

..... Les types de fichiers de l'univers **LAT_EX** (suite)

Remarque 1 : Des classes et des *packages* peuvent générer leur propres fichiers auxiliaires avec d'autres extensions dans le dossier de travail → liste non exhaustive.

Par exemple **ntheorem** produit un fichier auxiliaire **.thm** pour y placer les métadonnées relatives aux environnements de type "théorème" qu'il gère.

Remarque 2 : Les fichiers de polices d'écriture sont très nombreux dans une installation **T_EX** et sont de divers types : **.afm**, **.enc**, **.fd**, **.lig**, **.map**, **.mf**, **.otf**, **.pfb**, **.pk**, **.tfm**, **.ttf**, **.vf**...

C'est l'un des aspects les plus compliqués d'une installation **T_EX**.

Remarque 3 : Chaque *package* est fourni avec des fichiers **.ins** et **.dtx** : la compilation du fichier **.ins** crée et installe les fichiers du *package* (en compilant **.dtx**).
→ documentation, **.sty**, et d'autres fichiers.

Remarque 4 : Le premier fichier utilisé par le moteur **tex** pour compiler en **LAT_EX** est **latex.fmt** qui définit **LAT_EX** en tant que format (**.fmt**).

C'est une version pré-compilée d'un fichier nommé **latex.ltx** (8000 lignes de code) dans lequel figurent toutes les macros constituant **LAT_EX**.

Ce dernier fichier est consultable pour comprendre comment sont définies certaines commandes ou certains environnements.

Remarque 5 : Les autres logiciels fournis avec le moteur **tex** ont aussi à leur disposition des fichiers particuliers, avec leurs propres extensions.

..... Les dossiers **texmf**

Description d'une distribution **TeXLive** récente (2011).

Les fichiers d'une installation **T_EX** sont regroupés et structurés dans des dossiers **texmf**.

Il y a plusieurs dossiers **texmf** :

– L'installation principale et fondamentale se trouve dans un ensemble de dossiers **texmf** auxquels personne n'est censé toucher...

Typiquement, ils sont dans **/usr/local/texlive/2011/** et portent des noms comme : **texmf**, **texmf-config**, **texmf-dist**, **texmf-var**.

– L'administrateur de la machine peut installer des fichiers complémentaires à l'installation fondamentale dans un dossier **texmf-local** :

/usr/local/texlive/texmf-local/

Tout apport dans ce dossier est utilisable par tous les utilisateurs de la machine.

– Chaque utilisateur dispose d'un **texmf** personnel, à créer si nécessaire :

\$HOME/texmf/ (Windows), **\$HOME/Library/texmf/** (Mac),

%USERPROFILE%\texmf (Windows).

L'utilisateur peut y déposer les fichiers qu'il veut, il est le seul à pouvoir les utiliser.

Il y a une distribution **TeXLive** proposée tous les ans (vers l'été en général).

Le dossier **texmf-local** n'est pas touché par ces mises à jours.

Plusieurs distributions **TeXLive** peuvent cohabiter (années différentes).

.....
Recherche de fichiers dans les dossiers **texmf**

Règle fondamentale

Par principe, le moteur **tex** cherche en priorité un fichier souhaité (**.tex**, **.sty**, **.aux**, **.def**, **.pdf**, **.jpg**...) dans l'ordre suivant :

1. dans le dossier courant de travail (celui du fichier source) ;
2. puis dans le dossier **texmf** de l'utilisateur ;
3. puis dans **texmf-local** ;
4. enfin dans les **texmf** fondamentaux.

Par conséquent, tout fichier dans le **texmf** de l'utilisateur est prioritaire.

On trouvera dans le fichier **.log** la liste des fichiers trouvés et leur chemin.

Pour gagner du temps lors de la recherche d'un fichier dans ces immenses dossiers, chaque dossier **texmf** comporte un fichier **ls-R**, qui contient toute la liste hiérarchique des fichiers qu'il renferme.

→ Lorsqu'on ajoute un fichier dans un tel dossier, il faut actualiser le fichier **ls-R**.

On peut utiliser la commande **mktexlsr** dans un terminal ou le moteur **t1mgr**.

Les fichiers déposés dans votre dossier **texmf** personnel sont pris en compte directement sans mettre à jour de fichier **ls-R**.

.....
Structure des dossiers **texmf**

Chaque dossier **texmf** est structuré en sous-dossiers. Parmi ceux-ci on trouve :

bibtex Ce dossier contient lui-même deux sous dossiers intéressants :

- bib** pour les fichiers de bibliographie **.bib**
- bst** pour les fichiers de style bibliographiques **.bst**.

doc Ce dossier contient la documentation sur l'installation, les logiciels et les *packages*.

fonts Ce dossier contient tout ce qui concerne les polices d'écriture, organisés dans un fouillis (ordonné !) de sous dossiers...

tex C'est le dossier le plus intéressant. Il contient lui-même d'autres sous dossiers, dont :

- generic** Pour les fichiers et dossiers consacrés à tous les formats.
- latex** Pour les fichiers et dossiers consacrés au format **LATEX**.

web2c Ce dossier contient certains fichiers de configuration de l'installation **TEX** et les formats **.fmt**.

Il existe de nombreux autres dossiers et sous dossiers moins utiles pour le débutant.

En particulier chaque moteur a son propre dossier : **asymptote**, **dvipdfmx**, **dvips**, **makeindex**, **metapost**, **xdvi**...

Le dossier `texmf` de l'utilisateur

Le dossier `texmf` de l'utilisateur peut servir à :

- placer un *package* absent de l'installation par défaut ou qui n'est pas à jour ;
- placer des fichiers de style personnels ou récupérés sur l'Internet (`.sty`, `.bst`...), par exemple des feuilles de style d'éditeurs (livres, articles) ;
- organiser des fichiers communs à différents projets : des images (logos d'institut par exemple), des macros regroupées dans un fichier `.tex`...
- placer les bases de données bibliographiques qu'on gère (`.bib`).

Ce dossier `texmf` doit être structuré de la même façon que les autres dossiers `texmf`, mais sans nécessairement contenir autant de dossiers.

On ne crée donc un dossier/sous dossier que lorsque c'est nécessaire.

- Les fichiers des *packages* `.sty` iront dans `texmf/tex/latex/`.
- Les bases de données bibliographiques `.bib` iront dans `texmf/bibtex/bib/`.
- Les fichiers de style bibliographiques `.bst` iront dans `texmf/bibtex/bst/`.
- Les fichiers d'images `.eps`, `.pdf`, `.jpg`, `.png`... iront dans `texmf/tex/latex/`.

On peut créer des sous dossiers sans contrainte au delà de ces dossiers. Par exemple il est souhaitable de créer un dossier `images` dans `texmf/tex/latex/` pour regrouper les images et de regrouper dans des dossiers spécifiques les styles (revues, livres...).

L'installation d'un *package* dans son `texmf`

Les étapes à suivre pour installer un *package* dans son propre dossier `texmf` :

1. Récupérer les fichiers du *package*.

Localiser un *package* sur le site du [CTAN](#) :

Page de recherche du CTAN⁴³ (choisir *package descriptions*), ou classement par **fonction primaire des packages**⁴⁴.

Une fois trouvé, la page du *package* à une URL du type

<http://www.ctan.org/tex-archive/macros/latex/contrib/geometry/>

On peut obtenir le *package* sous forme d'un dossier archive (`.zip`).

2. Décompresser ce fichier et placer le dossier obtenu dans votre `texmf/tex/latex/` (ou autre selon la nature du *package*).

3. Dans ce dossier doit figurer un fichier de type `.ins` et/ou `.dtx`.

Lancer la compilation de `pdflatex` sur `.ins` (de préférence) ou sur `.dtx`.

Pour ça on peut passer par un terminal : `pdflatex geometry.ins`

ou on peut ouvrir ce fichier dans son éditeur `TeX` préféré...

Cette compilation produit les fichiers du *package* (`.sty`, `.fd`, `.cfg`, `.def`...).

Si le dossier ne contient pas un tel fichier, il n'y a rien à faire à cette étape.

Dans l'absolu, si on en a la possibilité, il est préférable de mettre à jour les *packages* avec l'utilitaire `tlmgr` ou mieux encore avec une interface graphique à cet utilitaire.

.....
À la recherche d'une documentation

Pour tout ce qui concerne les commandes et les environnements définis par **LATEX**, consulter **LATEX Reference Manual**³ (versions [HTML](#) ou [PDF](#) disponibles).

Les logiciels et les *packages* installés sur l'ordinateur sont en général documentés.

On peut accéder à cette documentation avec la commande **texdoc** :

texdoc geometry

ouvre le [PDF](#) de la documentation du *package* **geometry**.

→ fonctionne pour à peu près tous les *packages*.

texdoc article

ouvre le [PDF](#) de documentation des classes standard de **LATEX**.

texdoc pdftex

ouvre le [PDF](#) de documentation de **pdftex**.

texdoc texlive

ouvre le [PDF](#) de documentation de **TeXLive**.

texdoc lshort et **texdoc symbols**

ouvrent les [PDF](#) de **The Not So Short Introduction to LATEX**¹ et

Comprehensive LATEX symbol list² respectivement.

Il existe un logiciel d'interface pour **texdoc**.

3 – Découverte du langage de LATEX

La philosophie de TEX

Avec **TEX** et **LATEX**, on tape du texte (presque) normalement et on fait appel de temps en temps à des macros (des morceaux de programmes) pour obtenir des effets divers ou des fonctionnalités particulières.

Comme dans tout langage de programmation, il y a des caractères réservés.

En voici la liste, ainsi que la façon de les obtenir concrètement dans du texte :

<code>~</code>	<code>\~{}</code>	<code>\$</code>	<code>\\$</code>	<code>&</code>	<code>\&</code>	<code>%</code>	<code>\%</code>	<code>#</code>	<code>\#</code>
<code>^</code>	<code>\^{}</code>	<code>_</code>	<code>_</code>	<code>{</code>	<code>\{</code>	<code>}</code>	<code>\}</code>	<code>\</code>	<code>\textbackslash</code>

(voir aussi le package **textcomp**)

Avec des macros on peut : modifier les styles typographiques (taille, graisse, forme, couleur...), modifier la mise en page (marges, indentations...), ajouter des notes en bas de page, composer des titres, générer une table des matières, composer des mathématiques (symboles, disposition des formules...), insérer des images, créer des tableaux, composer des dessins...

Parmi ces macros, figurent les “commandes”, sous la forme `\cmd` (“\” suivi d’un nom).

TEX ne connaît que des commandes avec ou sans arguments.

LATEX a introduit les arguments optionnels et la notion d’environnement.

Le rôle des caractères réservés

Les caractères réservés se rencontrent dans les situations suivantes :

- `%` marque le début d’un commentaire.
- `\` marque le début d’une macro.

Exemple : `\newpage` est la commande pour forcer un saut page.

- `{` et `}` ouvrent et ferment un **groupe**, c’est à dire une entité unique pour **TEX**.

Cela sert à créer un contexte local ou à englober l’argument (long) d’une macro.

Exemples : `{\large texte en grand}`, `\textit{texte en italique}`.

→ les (re)définitions opérées à l’intérieur d’un groupe restent locales.

- `\~{}` crée un espace insécable.
- `$` commence et termine le *mode mathématique*.
- Dans le mode mathématique, `\^{}{}` sert à placer des exposants et `_{}{}` sert à placer des indices.
- `\&` intervient dans les séparations des colonnes de tableaux et de matrices, et comme positions d’alignement de formules de mathématique.
- `\#` se rencontre dans la définition de macros.

⚠ Les caractères `[` et `]` ne sont pas réservés, mais ils interviennent pour délimiter des arguments optionnels dans des macros **LATEX**, ce qui fait que leur rôle est ambigu.

→ Cette situation génère parfois des erreurs incompréhensibles.

.....
Commandes et environnements, généralités

Une commande est du type `\cmd`, où `cmd` est son nom.

Une commande peut accepter zéro, un ou plusieurs arguments.

→ Ce sont des “variables” au sens de la programmation.

Il y a deux types d’arguments :

- argument obligatoire, entouré par des accolades : `{-}` ;
- argument optionnel, entouré par des crochets droits : `[-]`.

L’ordre des arguments est important. Il est fixé par la syntaxe de la commande.

```
\pagebreak[3]
\section{Titre d'une section}
\Large ... \cite[p.~100]{Mass08}
```

Les environnements délimitent un morceau de texte : `\begin{env} ... \end{env}`.

`env` est le nom de l’environnement.

Il peut y avoir des arguments obligatoire et/ou optionnels après le `\begin{env}`.

```
\begin{center} ... \end{center}
\begin{tabular}{rlc} ... \end{tabular}
\begin{figure}[th] ... \end{figure}
```

.....
Commandes et environnements, exemples

Exemples de commandes :

```
\documentclass[-]{-}, \usepackage[-]{-}
\chapter[-]{-}, \section[-]{-}, \footnote{-}, \tableofcontents
\Large, \tiny, \textbf{-}, \textsf{-}, \textcolor{-}{-}
\par, \bigskip, \medskip, \smallskip, \vspace{-}
\newpage, \clearpage, \pagebreak[-]
\alpha, \otimes, \sum, \sin, \sqrt{-}, \frac{-}{-}
\includegraphics[-]{-}, \scalebox{-}{-}
\{, \$, \, , \backslash
```

Exemples d’environnements :

```
\begin{document} ... \end{document}
\begin{enumerate} ... \end{enumerate}
\begin{abstract} ... \end{abstract}
\begin{flushleft} ... \end{flushleft}
\begin{tabular}{-} ... \end{tabular}
\begin{minipage}{-} ... \end{minipage}
\begin{equation} ... \end{equation}
\begin{figure}[-] ... \end{figure}
\begin{thebibliography}[-] ... \end{thebibliography}
```

.....

Le B.A.BA de la frappe sous LATEX

- Pour TEX, un ou plusieurs espaces entre mots signifie la même chose.
On n'hérite pas “d'espaces” superflus entre les mots.
Il faut utiliser une commande pour augmenter l'espace entre deux mots : `\hspace{<dim>}`.
 - TEX absorbe les espaces qui suivent les commandes.
`\LaTeX est formidable` → LATEX est formidable,
`\LaTeX{} est formidable` → LATEX est formidable.
 - TEX traite les tabulations comme des espaces.
On peut utiliser des tabulations pour structurer le fichier source.
Il y a un environnement spécifique pour créer des tabulations.
 - TEX ignore les retours chariot isolés (un seul passage à la ligne).
Un retour chariot isolé est considéré comme un espace.
 - Une ligne vide (ou plus) signifie “passage dans un nouveau paragraphe”.
Arrêt du paragraphe courant, passage à la ligne avec éventuel espace vertical supplémentaire, indentation de la nouvelle ligne...
Utiliser sans modération pour aérer le code source.
Equivaut à placer `\par` à la fin de la ligne du paragraphe à terminer.
-

Le B.A.BA de la frappe sous LATEX (suite)

- TEX ignore ce qui est sur la même ligne au delà d'un caractère %.
Permet de commenter le fichier source, de désactiver des lignes de texte.
Le “retour chariot” à la fin d'une ligne contenant un % est aussi ignoré par TEX.
- \ interrompt la ligne courante et passe à la ligne.
À utiliser dans les environnements `center`, `flushleft`, `flushright`...
⚠ la nouvelle ligne ne commence pas un nouveau paragraphe.
`\[\langle dim \rangle]` ajoute en plus la longueur `<dim>` verticalement entre les lignes.
`\[*[\langle dim \rangle]` interdit en plus un saut de page.
- Un `_` (\ suivi d'un espace) créé un espace de largeur fixe.
Un `\~` collant deux mots crée un espace de largeur fixe et insécable.
L'espace habituel entre les mots a une largeur variable pour permettre la justification des lignes.
- Les guillemets (anglais) sont saisis avec la syntaxe ‘ ‘mot’ ’ → “mot”
‘ ‘ est un premier ‘ suivi d'un second ‘. Idem pour ’ ’.
- Il y a trois sortes de tirets :
– → - (trait d'union), – → – (tirets de listes), --- → — (début de dialogue)
- Les points de suspension sont obtenus pas `\dots` ou `\ldots` → ...

.....
Les commandes de changement de style du texte

Attribut	Exemple	Commande
family	Roman	\rmfamily \textrm{-}
family	Sans serif	\sffamily \textsf{-}
family	Type Writer	\ttfamily \texttt{-}
series	medium	\mdseries \textmd{-}
series	bold	\bfseries \textbf{-}
shape	up	\upshape \textup{-}
shape	<i>italic</i>	\itshape \textit{-}
shape	<i>slanted</i>	\s1shape \textsl{-}
shape	SMALL CAPS	\scshape \textsc{-}
medium, up		\normalfont \textnormal{-}

Des *packages* peuvent définir d'autres commandes que celles-ci en fonction de ce que contient la police courante.

Les commandes \normalfont et \textnormal{-} rechargent les valeurs par défaut.

→ Elles annulent toutes les autres commandes.

.....
Les commandes \emph{-} et \em

La commande \emph{-} met en valeur le texte qu'elle contient en basculant entre les formes droites et les formes italiques.

Elle admet un équivalent sous la forme {\em ...}.

\sffamily\mdseries\upshape Veritatis \emph{simplex} oratio est
Veritatis *simplex* oratio est

\sffamily\bfseries\upshape Veritatis \emph{simplex} oratio est
Veritatis **simplex** oratio est

\itshape Veritatis \emph{simplex} oratio est
Veritatis *simplex* oratio est

\bfseries\itshape Veritatis \emph{simplex} oratio est
Veritatis ***simplex*** oratio est

.....

Les commandes de changement de la taille du texte

Commande	Exemple
\tiny	tiny
\scriptsize	scriptsize
\footnotesize	footnotesize
\small	small
\normalsize	normalsize
\large	large
\Large	Large
\LARGE	LARGE
\huge	huge
\Huge	Huge

Les tailles déterminées par ces commandes sont relatives à la taille de référence choisie pour le document en entier (option de `\documentclass[-]{-}`).

.....

Exemples de changement de style et de taille

```
\sffamily\bfseries\s lshape Veritatis simplex oratio est
Veritatis simplex oratio est
```

```
\rmfamily\bfseries\itshape Veritatis simplex oratio est
Veritatis simplex oratio est
```

```
\sffamily\bfseries\itshape Veritatis simplex
\normalfont oratio est
Veritatis simplex oratio est
```

```
\sffamily\large Veritatis
{\rmfamily\itshape{\bfseries\footnotesize simplex} oratio} est
Veritatis simplex oratio est
```

Noter sur cet exemple la portée des commandes.

.....
Exemples de changement de style et de taille (suite)

```
\sffamily\large\bfseries\itshape Veritatis simplex oratio est
=
\itshape\large\bfseries\sffamily Veritatis simplex oratio est
Veritatis simplex oratio est = Veritatis simplex oratio est
```

L'ordre des commandes ne change pas le résultat.

⚠ Il faut bannir les commandes d'anciennes versions de LATEX :
`\rm, \sf, \tt, \bf, \sl, \it, \sc`.

```
{\it\bf test}
{\bf\it test}
{\itshape\bfseries test}
test test test
```

Le résultat est incorrect et dépend de l'ordre des commandes !
Seul le dernier produit ce qu'on souhaite.

.....
Gestion de la mise en page

Quelques commandes de mise en page :

- `\newpage` impose un saut de page.
`\pagebreak[num]` suggère un saut de page.
De `num` = 0 (on peut...) à `num` = 4 (on veut...).
`\nopagebreak[num]` dissuade un saut de page.
`\clearpage` traite les flottants en court et impose un saut de page.
`\cleardoublepage` procède de même en commençant une nouvelle page à droite (mode `twoside`).
- `\enlargethispage{<dim>}` agrandit la page courante de la dimension spécifiée.
Ne pas abuser de cette commande !
- `\indent` crée l'indentation en début de paragraphe.
`\noindent` supprime l'indentation en début de paragraphe.
La dimension `\parindent` correspond à cette indentation :
→ `\setlength{\parindent}{0pt}` supprime toute indentation.
- `\newline` interrompt la ligne courante et en démarre une nouvelle.
`\linebreak[num]` suggère une interruption de ligne.
`\nolinebreak[num]` dissuade une interruption de ligne.
- `\footnote{-}` insère une note de bas de page.

..... La justification

Par défaut, LATEX justifie le texte dans les paragraphes : en jouant sur les espacements entre mots et les césures de mots, le texte est aligné sur les bords gauches et droits.

Les environnements `flushleft`, `flushright` et `center` permettent, respectivement, de coller le texte sur le bord gauche, de coller le texte sur le bord droit et de centrer le texte.

Ces environnements acceptent des paragraphes et des ruptures de lignes avec `\backslash`.

```
\begin{center}
Actibus immensis\\
urbs fulget\\[2mm]
massiliensis
\end{center}
```

Actibus immensis
urbs fulget
massiliensis

Ces environnements ont des commandes (presque) équivalentes : `\raggedright`, `\raggedleft` et `\centering` (⚠ noms inversés !).

```
{\raggedleft Felix, qui potuit
rerum cognoscere causas\par}
```

Felix, qui potuit rerum
cognoscere causas

⚠ ces commandes ne fonctionnent que sur un paragraphe entier, d'où le `\par`.

..... Quelques commandes diverses

Quelques commandes de nature typographique :

- Les environnements `quote` et `quotation` servent à citer du texte : retraits du texte à droite et à gauche, `quotation` indente les paragraphes.
- L'environnement `verse` est destiné à la poésie.
- L'environnement `verbatim` sert à présenter du texte non interprété.
- Césures : LATEX dispose d'une liste de césures.

On peut localement l'aider en découplant un mot soi-même : `a1\-\gè\-\bre`

On peut le faire globalement : `\hyphenation{a1-gè-bre vec-to-riel}`

`\showhyphens{-}` affiche dans la console et le `.log` les suggestions de LATEX :

`\showhyphens{algèbre vectoriel} → a1-gèbre vec-to-riel`

- “Pavé noir” : `\rule[<dimh>]{<dimx>}{<dimy>}` créé un bloc plein de taille $\langle dimx \rangle \times \langle dimy \rangle$ relevé de la hauteur `<dimh>` :

`\rule{1cm}{3pt} → _____,`

`x\rule[1ex]{1cm}{1pt}x → x_____x`

Si l'une des dimensions est nulle, la place est réservée mais sans rien afficher :

`x\rule{1cm}{0pt}x → x x`

→ Très pratique pour créer des espaces rigides.

⚠ Ne pas abuser, c'est un dépannage !

4 – Structure du document source

Structure \LaTeX du document source

Tout fichier \LaTeX a la structure suivante :

$\text{\documentclass[...]{...}}$	Choix de la classe du document.
$\text{\usepackage[...]{...}}$	
...	
$\text{\usepackage[...]{...}}$	Appel des <i>packages</i> ...
AUTRES DÉCLARATIONS	... et définitions générales globales.
$\text{\begin{document}}$	
CORPS DU TEXTE	Le corps du texte est contenu dans l'environnement $\text{\begin{document}} \dots \text{\end{document}}$.
$\text{\end{document}}$	

Tout ce qui est au delà de $\text{\end{document}}$ est ignoré par \LaTeX .

On appelle **préambule** la zone comprise entre $\text{\documentclass[-]{-}}$ et $\text{\begin{document}}$.

$\text{\documentclass[-]{-}}$: les classes

La classe d'un document détermine :

- son apparence ;
- sa structure logique.

Les classes standard de \LaTeX :

article C'est la classe usuelle pour les articles scientifiques.

Cette classe permet d'écrire des petits documents.

report C'est une classe prévue pour des documents plus conséquents avec chapitres et résumé.

Peut convenir pour un rapport d'étudiant.

book Comme son nom l'indique, est spécialement prévue pour les livres (pas de résumé).

En fait, **report** peut aussi faire l'affaire pour des livres.

beamer C'est une classe qui permet de préparer des présentations par vidéo-projecteurs.

Le document est composé dans le format paysage, avec des lettres assez grosses.

memoir Cette classe a été conçue par son auteur pour répondre à l'éternelle question :

Comment modifier l'aspect de tel ou tel élément d'un document \LaTeX ?

C'est une solution "tout-en-un" (non modulaire) dont la documentation est énorme.

Les classes de KOMA-script C'est un ensemble de classes et de *packages* associés qui remplacent les classes de \LaTeX : **scrbook**, **scrreprt**, **scrartcl**, **scrletter**.

..... \documentclass[-]{-} : les options de classes

De nombreuses options de classes modifient l'aspect global du document :

Taille des caractères : `10pt, 11pt, 12pt`.

Taille de la page : `a4paper, a5paper, letterpaper...`

Mise en page : `twoside, landscape, onecolumn, twocolumn, openright, openany, titlepage`.

Divers : `draft, final, leqno, fleqn, openbib`.

À la place de certaines options, il est préférable d'utiliser des *packages* qui font mieux :

`geometry` permet de gérer la taille de la page et son orientation,

`multicol` gère les mises en page multi-colonnes,

`amsmath` propose des options `fleqn, leqno` et `reqno`,

`titling` personnalise la page de titre,

`natbib` redéfinit la gestion de la bibliographie,

`draftwatermark` s'occupe de signaler que le document est un “draft”...

Les options données avec la classe sont “passées” aux *packages*.

Les *packages* les utilisent ou non.

..... \usepackage[-]{-} : les packages

Les *packages* sont des modules qui ajoutent des fonctionnalités à `LATEX`.

→ il faut donc savoir si un *package* fait ce qu'on désire, et lequel.

Un *package* contient un ensemble de macros dans un fichier `.sty` (ou plusieurs fichiers).

Une fois appelé par la commande `\usepackage{-}`, on a accès à des commandes et des environnements nouveaux.

On peut appeler plusieurs *packages* en même temps :

`\usepackage{pack1, pack2, ...}`

→ regrouper les *packages* thématiquement proches.

Il y a des *packages* pour des usages très différents : encodage du texte source, gestion des polices, gestion de l'apparence des pages, inclusion de graphiques, fonctionnalités mathématiques avancées, gestion de langues, dessins internes au document, programmation...

Les *packages* peuvent comporter des options qu'on peut activer lors de l'appel.

La commande `\usepackage[english, french]{babel}` appelle le *package* `babel` avec les options `english` et `french`.

Les options passées directement aux *packages* sont prioritaires sur les options passées avec la classe du document.

.....
\usepackage[-]{-} : les packages (suite)

⚠ *Tous les packages sont égaux, mais certains sont plus égaux que d'autres.*

- Certains *packages* modifient des fonctionnalités fournies par d'autres *packages*
 ➔ il est préférable de les appeler en dernier (**babel**, **hyperref**...).
- Des *packages* différents peuvent avoir des fonctionnalités semblables et entrer en conflit l'un avec l'autre.
- *Un package peut en cacher un autre* : il est possible qu'un *package* appelle d'autres *packages*, donc surveiller les incompatibilités induites.

Des livres sur **LATEX** permettent de découvrir l'existence de *packages* intéressants et d'en comprendre le fonctionnement.

Le site du **CTAN**⁴⁴ permet de chercher dans la description des *packages*.

.....
Structure logique et sémantique d'un document

Chaque classe a sa propre structure logique.

La structure logique la plus utilisée est celle des découpages en parties, chapitres, sections, sous sections, paragraphes et sous paragraphes.

Les classes **report** et **book** définissent les commandes **\part**, **\chapter**, **\section**, **\subsection**, **\subsubsection**, **\paragraph**, **\subparagraph**.

La classe **article** ne connaît pas **\chapter**.

Ces commandes acceptent un argument optionnel pour donner une version courte.

\section[Titre court]{Version longue du titre qui n'en finit pas}

Le titre long est celui utilisé dans le corps du texte.

Le titre court est utilisé à la fois dans la table des matières et dans les entêtes des pages.

Après le texte principal, on peut placer des annexes avec la commande **\appendix**.

Automatiquement la numérotation de plus haut niveau (**\section** pour **article**, **\chapter** pour **report** et **book**) devient alphabétique.

La classe **book** permet de diviser le texte en trois grandes parties :

une préface qui démarre avec **\frontmatter**, le corps du texte qui démarre avec **\mainmatter** et une post-face qui démarre avec **\backmatter**.

Les chapitres ne sont pas numérotés dans la préface et la post-face.

La numérotation des pages se fait en romain dans la préface.

..... La page de garde

Dans certaines classes, **LATEX** compose une page de garde à partir de données spécifiques :

- `\title{–}` définit le titre,
- `\author{–}` renseigne les auteurs (séparés par la commande `\and`),
- `\date{–}` insère la date.

Des commandes `\thanks{–}` associées aux auteurs produisent des informations en notes de bas de page.

Après avoir renseigné ces commandes, `\maketitle` compose la page de garde.

Dans certaines classes, l'environnement `abstract` met en forme un résumé.

Il n'est pas nécessaire d'utiliser `\maketitle` pour faire un document avec **LATEX** :

- on peut se passer d'une page de garde,
- on peut composer soi-même la page de garde.

..... Éléments gérés par **LATEX**

LATEX gère un certain nombre d'éléments automatiquement.

Numérotations : **LATEX** s'occupe de numérotter de nombreux éléments : parties, chapitres, ..., paragraphes, pages, notes de bas de page, tableaux, figures, équations, théorèmes (ou équivalents), items de listes...

Table des matières : En se basant sur les commandes `\chapter{–}`, `\section{–}`, ... la commande `\tableofcontents` imprime une table des matières où elle est placée.

Selon la classe du document, cette table des matières démarre un nouveau chapitre (`report` et `book`) ou une section (`article`).

Il y a des commandes équivalentes pour la liste des tableaux (`\listoftables`) et la liste des figures (`\listoffigures`).

Références croisées : La commande `\label{lb}` définit une clé interne au document (`lb`) qui permet de faire référence ailleurs à la valeur d'une numérotation ou de la page où cette clé est définie.

La numérotation est celle du contexte ambiant : chapitre, section, note de bas de page, équation, tableau, figure, théorème, item...

La clé `lb` ne doit être définie qu'une seule fois dans tout le document.

`\ref{lb}` imprime la valeur de la numérotation ainsi préservée,

`\pageref{lb}` imprime la page.

..... Les multiples compilations

Pour réduire l'utilisation de la mémoire vive, **T_EX** (et donc **L_AT_EX**) ne conserve en mémoire aucune donnée sur le texte lors d'une compilation.

→ **L_AT_EX** utilise des fichiers auxiliaires dans lesquels il place des données à préserver : **.aux**, **.toc**, **.lof**, **.lot**..., et des *packages* en produisent d'autres.

- Première étape : en compilant, **L_AT_EX** remplit de données les fichiers auxiliaires.
 - **.aux** contient les données pour les références croisées,
 - **.toc** contient les entrées de la table des matières...
- Compilation suivante : **L_AT_EX** lit les fichiers auxiliaires et utilise leur contenu :
 - pour formater les références croisées,
 - pour composer la table des matières...

Produire un gros document exige souvent plusieurs compilations **L_AT_EX**.

Il faut parfois 3 compilations pour stabiliser la table des matières !

La console de compilation indique si le fichier **.aux** a été modifié et si une compilation supplémentaire est nécessaire.

Pour composer une bibliographie à l'aide d'une base de donnée **.bib** et/ou un index, il faut entrecouper ces compilations de l'usage des moteurs **bibtex** et/ou **makeindex**.

..... Découper un document en pièces

Lorsqu'on gère un projet important, il est utile de le découper en pièces plus petites.

De façon générale, le document source peut charger d'autres fichiers.

\input{fichier.tex} lit le fichier **fichier.tex** à l'endroit désigné, l'interprète, puis retourne à la suite du document courant.

- n'importe où dans le document (préambule ou corps du texte) ;
- on peut ainsi inclure une portion du texte final ou un ensemble de commandes ;
- la recherche se fait dans les dossiers **texmf** comme décrit auparavant : le fichier n'est pas obligé d'être dans le dossier courant.

\include{chap1} dans le corps du texte inclut le fichier **chap1.tex** (⚠ extension).

Cette commande démarre une nouvelle page avant et après.

- réservé à des chapitres entiers de documents importants (livres, polycopiés...) ;
- un fichier **.aux** est créé pour chaque fichier inclus.

\includeonly{chap1, chap3, chap4} dans le préambule permet de sélectionner certains fichiers seulement.

.....

Conseils élémentaires pour maîtriser **LATEX**

Apprendre **LATEX**, c'est apprendre le langage **TEX/LATEX** ainsi qu'un logiciel d'interface.

Règle 1 : Si votre clavier n'est pas américain, *trouver comment produire le symbole \.*

Règle 2 : Rechercher un symbole inconnu dans **Comprehensive LATEX symbol list**^{⊗2}.

Règle 3 : Rechercher la syntaxe d'une macro dans **LATEX Reference Manual**^{⊗3}.

L'interface : C'est le lieu de travail, donc l'essentiel.

- Choisir un environnement de travail riche, puissant et surtout bien maîtrisé.
- Créer un ensemble de fichiers d'exemples (vides de contenu sémantique) pour différents types de documents : articles, lettres, mémoires, rapports, CV...

Certains logiciels gèrent de tels modèles qu'il est facile d'appeler lors de la création d'un nouveau document.

Gestion du code source : Le code source, c'est le reflet de notre personnalité...

- Aérer le texte source, mettre des lignes de commentaires et des indentations.
Pour **LATEX**, une ou plusieurs lignes vides, c'est la même chose.
- Marquer clairement le début et la fin des groupes
(`{ ... }` ou `\begin{env} ... \end{env}`).
- S'il s'agit d'un gros projet (rapport, livre), découper le texte en petits morceaux, et définir un fichier "maître" qui appelle les autres avec des `\include{-}`.
 - Chaque fichier de travail (chapitre par exemple) est plus petit.
 - Commenter certains `\include{-}` pour ne pas compiler tout le document.

.....

Conseils élémentaires pour maîtriser **LATEX** (suite)

Gestion des erreurs : Les erreurs sont inévitables, il faut savoir les gérer.

- À l'ouverture d'un groupe, le fermer immédiatement avant de le remplir.
- Compiler souvent le texte pour éviter d'accumuler des erreurs et pour pouvoir mieux cerner les zones en cause.
 - ⚠ Une erreur à un endroit peut se révéler dans la compilation bien plus loin !
- Observer la console de compilation, et si ça ne suffit pas, ouvrir le fichier `.log`.
C'est une mine d'informations qui permet de suivre le déroulement de la compilation (plus d'infos dans le `.log` que dans la console) et de comprendre la source des erreurs.
- En cas de problème récalcitrant, déplacer le `\end{document}` vers le haut jusqu'à trouver la zone qui ne compile pas, ou mettre en commentaires de grands blocs de texte (certains logiciels le permettent facilement).

Personnalisation : Ne pas hésiter à personnaliser votre façon de travailler.

- Définir des macros sémantiques (qui donnent du sens au texte) dont il sera possible de modifier le comportement jusqu'à la dernière compilation.
`\section{-}` est une macro sémantique,
`\important{texte important}...`
- Ne pas hésiter à utiliser un *package* qui fera le travail demandé...
 - inutile de réinventer la roue à chaque besoin spécifique.

5 – Des packages essentiels

..... Une modèle de fichier source

Voici à quoi peut ressembler un fichier source **TEX** de base :

```
\documentclass[12pt]{article}
\usepackage{etex}
\usepackage[utf8]{inputenc}
\usepackage[TS1,T1]{fontenc}
\usepackage[a4paper, tmargin=3cm, bmargin=3cm,
            rmargin=2.2cm, lmargin=2.2cm]{geometry}
\usepackage{textcomp}
\usepackage{lmodern}
\usepackage[english,french]{babel}
\usepackage{hyperref}

\begin{document}
    Mon texte\ldots
\end{document}
```

Ces lignes sont commentées dans ce qui suit, mais pas dans l'ordre...

Ces *packages* peuvent être utilisés avec d'autres classes que la classe **article**.
→ **report**, **book**...

..... La gourmandise de certains *packages* : **etex**

Certains *packages* sont gourmands en espace mémoire (longueurs, compteurs...).

On aboutit parfois à des messages d'erreurs du type :

! No room for a new \dimen
! No room for a new \count

Il n'y a que 256 registres possibles par construction du moteur **tex** d'origine.

Le moteur **pdftex** peut allouer $2^{15} = 32\,768$ registres.

Mais ces registres ne sont pas disponibles par défaut !

Le *package* **etex** active ces registres supplémentaires.

Il doit être placé avant les autres *packages* afin que ces derniers profitent de ces registres.

..... L'encodage du fichier source : **inputenc**

En entrée (fichier source), le moteur **tex** accepte seulement un encodage sur 8 bits
→ $2^8 = 256$ caractères possibles seulement.

Deux difficultés importantes :

1. Il faut recourir à des macros pour des symboles absents de ces 256 possibilités :

`\alpha` → α , `\textcopyright` → \circledcirc , et les lettres accentuées peu usuelles...

LATEX permet actuellement l'accès à plus de 5000 symboles par des macros diverses (avec des *packages* et polices appropriés).

2. *Il n'y a pas d'encodage standard sur 8 bits dans le monde informatique.*

La position des 256 caractères (l'encodage) dépend des systèmes d'exploitations.

LATEX a besoin de connaître l'encodage utilisé lors de la sauvegarde du fichier source.

Des noms ont été donnés à ces encodages :

`ansinew`⁴⁵ (Windows), `applemac`⁴⁶ (Macintosh), `latin1`⁴⁷ (Unix/Linux)...

`utf8` pour l'**UNICODE**⁴⁸.

→ Régler l'éditeur de texte sur un encodage, et appeler **inputenc** avec la bonne option :

`\usepackage[utf8]{inputenc}` sélectionne l'encodage **UNICODE**.

Privilégier l'**UNICODE** qui est un standard.

..... L'encodage des fontes : **fontenc**

Les fichiers de fontes utilisés par **LATEX** ne contiennent que 256 glyphes (8 bits...).

Pour composer un texte, **LATEX** doit assembler de nombreux fichiers de fontes.

“L'encodage des fontes” standardise la façon dont on morcelle une police d'écriture en blocs “réduits” de 256 glyphes.

Ce morcellement dépend surtout de l'alphabet utilisé.

→ incidence sur les césures des mots...

Le package **fontenc** informe **LATEX** sur les encodages utilisés dans le document.

Encodages usuels des polices d'Europe de l'ouest :

OT1 C'est l'encodage d'origine, “O” pour “Old”...

Cet encodage ignore les glyphes des lettres accentuées → ne pas l'utiliser.

T1 Cet encodage contient l'essentiel des caractères des langues de l'Europe de l'ouest.

C'est l'encodage à utiliser de préférence.

TS1 Encodage “Text Symbol”. C'est un encodage compagnon de **T1** qui permet l'accès à des glyphes de type “symboles” utilisés dans des textes, comme \circledcirc , $\%_00$ ou \ddagger .

`\usepackage[TS1, T1]{fontenc}` active les encodages **T1** et **TS1**.

Autres encodages : cyrillique, grec, chinois/japonais/koréen (CJK), phonétique...

On peut basculer localement d'un encodage à un autre : possibilité d'écrire avec de nombreux types d'alphabets dans un document **LATEX**.

.....
Le package `textcomp`

textcomp donne accès à des symboles divers rangés dans l'encodage **TS1**.
Quelques exemples parmi les 127 accessibles :

<code>\textasteriskcentered</code>	*	<code>\textdollar</code>	\$
<code>\texteuro</code>	€	<code>\textyen</code>	¥
<code>\textasciicircum</code>	^	<code>\textasciitilde</code>	~
<code>\textbar</code>		<code>\textbackslash</code>	\
<code>\textbraceleft</code>	{	<code>\textbraceright</code>	}
<code>\textlbrackdbl</code>	〔	<code>\textrbrackdbl</code>	〕
<code>\textrightarrow</code>	→	<code>\textleftarrow</code>	←
<code>\textdownarrow</code>	↓	<code>\textuparrow</code>	↑
<code>\textlangle</code>	<	<code>\textrangle</code>	>
<code>\textcopyright</code>	©	<code>\textregistered</code>	®
<code>\texttrademark</code>	™	<code>\textservicemark</code>	SM
<code>\textdagger</code>	†	<code>\textdaggerdbl</code>	‡
<code>\textperthousand</code>	%	<code>\textpertenthousand</code>	%
<code>\textdegree</code>	°	<code>\textcelsius</code>	°C
<code>\textborn</code>	★	<code>\textmu</code>	μ
<code>\textohm</code>	Ω	<code>\textmho</code>	℧
<code>\textleaf</code>	leaf	<code>\textmusicalnote</code>	♪

.....
Accès à d'autres symboles : le package `pifont`

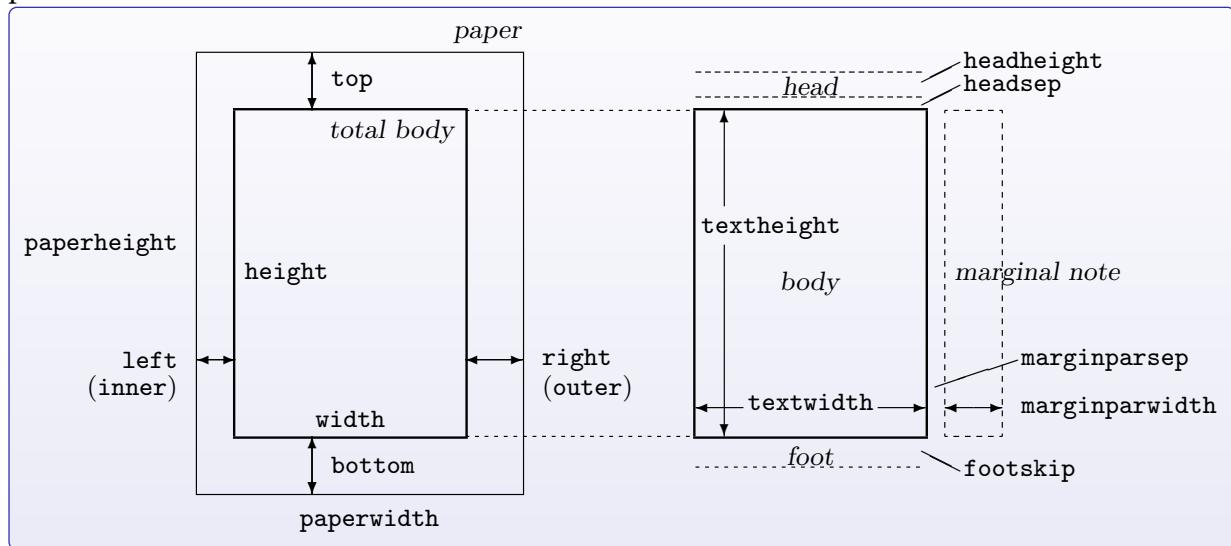
Le package **pifont** permet d'accéder aux caractères de la police **ZAPF DINGBATS** par la commande `\ding{nbr}` où *nbr* est le numéro dans le tableau ci-dessous :

	33 ☐	34 ☑	35 ☒	36 ☓	37 ☔	38 ☕	39 ☖	40 ☗
41 ☔	42 ☕	43 ☖	44 ☗	45 ☘	46 ☙	47 ☚	48 ☛	49 ☚
51 ☐	52 ☑	53 ☒	54 ☓	55 ☔	56 ☕	57 ☖	58 ☗	59 ☖
61 ☔	62 ☕	63 ☖	64 ☗	65 ☘	66 ☙	67 ☚	68 ☛	69 ☖
71 ☔	72 ☐	73 ☑	74 ☒	75 ☓	76 ☔	77 ☐	78 ☑	79 ☒
81 ☐	82 ☑	83 ☒	84 ☓	85 ☔	86 ☕	87 ☖	88 ☗	89 ☖
91 ☐	92 ☑	93 ☒	94 ☓	95 ☔	96 ☕	97 ☖	98 ☗	99 ☖
101 ☐	102 ☑	103 ☒	104 ☓	105 ☔	106 ☕	107 ☖	108 ☗	109 ☖
111 ☐	112 ☑	113 ☒	114 ☓	115 ☔	116 ☕	117 ☖	118 ☗	119 ☖
121 ☐	122 ☑	123 ☒	124 ☓	125 ☔	126 ☕			120 ☗
161 ☐	162 ☑	163 ☒	164 ☓	165 ☔	166 ☕	167 ☖	168 ☗	169 ☖
171 ☐	172 ☑	173 ☒	174 ☓	175 ☔	176 ☕	177 ☖	178 ☗	179 ☖
181 ☐	182 ☑	183 ☒	184 ☓	185 ☔	186 ☕	187 ☖	188 ☗	189 ☖
191 ☐	192 ☑	193 ☒	194 ☓	195 ☔	196 ☕	197 ☖	198 ☗	199 ☖
201 ☐	202 ☑	203 ☒	204 ☓	205 ☔	206 ☕	207 ☖	208 ☗	209 ☖
211 ☐	212 ☑	213 ☒	214 ☓	215 ☔	216 ☕	217 ☖	218 ☗	219 ☖
221 ☐	222 ☑	223 ☒	224 ☓	225 ☔	226 ☕	227 ☖	228 ☗	229 ☖
231 ☐	232 ☑	233 ☒	234 ☓	235 ☔	236 ☕	237 ☖	238 ☗	239 ☖
241 ☐	242 ☑	243 ☒	244 ☓	245 ☔	246 ☕	247 ☖	248 ☗	249 ☖
251 ☐	252 ☑	253 ☒	254 ☓					250 ☗

Voir aussi le package de symboles **marvosym**...

.....
La géométrie d'une page **LAT_EX**

Une page **LAT_EX** est divisée en zones précises, dont la géométrie est gouvernée par différents paramètres :



Il est possible de fixer soi-même ses paramètres, mais il faut alors respecter certaines contraintes, comme par exemple :

$$\begin{aligned} \text{paperwidth} &= \text{left} + \text{width} + \text{right} \\ \text{paperheight} &= \text{top} + \text{height} + \text{bottom} \end{aligned}$$

.....
Gestion de la géométrie : **geometry**

Le package **geometry** permet de gérer cette situation complexe.

Il donne à l'utilisateur la possibilité de fixer les valeurs de certains paramètres, et il se charge de calculer ceux qui manquent.

- En cas de sous détermination, il utilise des valeurs par défaut.
- En cas de sur-détermination, il y a des règles de priorité.
- Il est possible de sauvegarder une géométrie, de basculer sur une nouvelle géométrie (qui démarre sur une nouvelle page) et d'appeler une géométrie sauvée.
- Il est possible de choisir une zone de travail (**layout**) plus petite que la taille du papier.

La géométrie de la page peut être fixée par :

- Les options dans `\usepackage[-]{geometry}`
- La commande `\geometry{-}` placée dans le préambule.

Il faut bien sûr appeler le *package* avant d'utiliser cette commande.

Cette commande est prioritaire sur les options du *package*.

Les options spécifiées avec ce *package* sont prioritaires sur celles de même nature fournies avec la classe du document (taille du papier).

..... La syntaxe d'une géométrie

```
\usepackage[a4paper, tmargin=3cm, bmargin=3cm,
           rmargin=2.2cm, lmargin=2.2cm]{geometry}
```

Spécifie : la taille du papier (`a4paper`), les marges “top”, “bottom”, “right” et “left” (`tmargin`, `bmargin`, `rmargin` et `lmargin`).

C'est équivalent à :

```
\usepackage[a4paper, vmargin=3cm, hmargin=2.2cm]{geometry}
```

où `vmargin` et `hmargin` sont les marges “verticales” et “horizontales”.

Quelques options utiles :

Options générales `verbose`, `twoside`, `showframe`, `landscape`, `portrait`...

Taille du papier `a0paper`, ..., `a6paper`, `screen`, `paperwidth=<dim>`, `paperheight=<dim>`...

Taille du corps `textwidth=<dim>`, `textheight=<dim>`, `text={<dim>, <dim>}`...

Taille des marges `tmargin=<dim>`, ..., `hmargin=<dim>`, `vmargin=<dim>`, `inner=<dim>`, `outer=<dim>`...

Possible de définir des marges ou au contraire une taille du corps du texte.

Gestion des marges intérieures et extérieures pour les impressions en recto-verso.

..... S'adapter aux langues : `babel`

`LATeX` utilise par défaut la typographie de la langue anglaise (américaine).

Les mots-clés structurants (“Chapter”, “Contents”, “Table”...) sont aussi en anglais.

Le package `babel` permet de gérer les spécificités de plus de 40 langues :

- traduction des mots usuels produits par `LATeX` ainsi que de certaines macros ;
- utilisation des motifs de césure et des règles typographiques de la langue spécifiée ;
- lot de macros spécifiques à la langue.

```
\usepackage[english,french]{babel}
```

installe les langues “english” et “french” pour le document et sélectionne “french” comme langue par défaut (le dernier de la liste).

Il faut placer dans cette liste toutes les langues utilisées par la suite dans le document.

`\selectlanguage{english}` active les attributs de la langue anglaise.

```
\begin{otherlanguage}{german} ... \end{otherlanguage}
```

permet d'englober un morceau de texte en allemand.

`babel` définit de nombreuses commandes pour gérer plusieurs langues dans les macros personnelles ➔ lire sa documentation pour des détails.

⚠ Il faut placer l'appel à `babel` après les *packages* susceptibles de définir des mots-clés.

.....
babel et les spécificités de la langue française

Traduction des mots usuels : “Chapter” → “Chapitre”, “Contents” → “Table des matières”, “Table” → “Tableau”...

Date : `\today` donne la date en français :

anglais → April 17, 2012

français → 17 avril 2012

Typographie française : Indentation pour le premier paragraphe, guillements français : `\og mot \fg` → « mot », espace avant les doubles ponctuations : ; ! ?

Macros diverses : `\primo` → 1^o , `\secundo` → 2^o , `M\up{me}` → M^{me}, `1\ier` → 1^{er}, `3\ieme` → 3^e, `\Nos` → N^{os} ...

Listes : Les items des listes sont “–” au lieu de “•” et les espaces sont ajustés à la typographie française.

Césures : Les césures françaises sont différentes des césures anglaises :

En anglais, `\showhyphens{signal container}` produit
[] \T1/lmr/m/n/12 sig-nal con-tainer

En français, `\showhyphens{signal container}` produit
[] \T1/lmr/m/n/12 si-gnal contai-ner
→ `\tex` propose de couper ces mots à des endroits différents.

.....
Les mots-clés de LATEX

Les mots-clés suivants sont traduits par **babel** selon la langue installée.

Commande	Anglais	Français
<code>\prefacename</code>	Preface	Préface
<code>\abstractname</code>	Abstract	Résumé
<code>\partname</code>	Part	partie
<code>\chaptername</code>	Chapter	Chapitre
<code>\appendixname</code>	Appendix	Annexe
<code>\refname</code>	References	Références
<code>\bibname</code>	Bibliography	Bibliographie
<code>\contentsname</code>	Contents	Table des matières
<code>\listfigurename</code>	List of Figures	Table des figures
<code>\listtablename</code>	List of Tables	Liste des tableaux
<code>\indexname</code>	Index	Index
<code>\figurename</code>	Figure	FIGURE
<code>\tablename</code>	Table	TABLE
<code>\pagename</code>	Page	page
<code>\seename</code>	see	voir
<code>\alsoname</code>	see also	voir aussi

Il est possible de redéfinir soi-même ces commandes.

.....

Hyperliens, navigation et métadonnées avec PDF

Le format **PDF**²⁶ a des fonctionnalités avancées :

hyperliens internes qui permettent de passer d'une page à une autre dans le document ;

hyperliens externes qui acceptent les URL usuelles, par exemple vers des pages web (<http://...>) ou des adresses emails (<mailto:...>) ;

table des matières qui permet de naviguer dans le document (*bookmarks*) ;

métadonnées du document qui renseignent sur l'auteur, le titre, les mots-clés, le logiciel de création...

Le package **hyperref** permet de profiter quasi-automatiquement de ces fonctionnalités :

- hyperliens internes vers les `\label{-}` à partir de `\ref{-}` et `\pageref{-}` ;
- hyperliens internes à partir de la table des matières **LaTeX** (`\tableofcontents`) vers les pages concernées, de même pour les listes de figures et de tableaux ;
- hyperliens internes vers les entrées bibliographiques à partir de `\cite{-}` ;
- table des matières du document **PDF** générée sur les informations `\chapter{-}` ...
`\subsection{-}` du document.

.....

Configuration de **hyperref**, métadonnées du PDF

La commande `\hypersetup{-}` permet de personnaliser le comportement de **hyperref** et du document **PDF** produit.

```
\hypersetup{
  plainpages=false,
  hidelinks=false,
  colorlinks=true, linkcolor=black, anchorcolor=black,
  citecolor=black, menucolor=black, urlcolor=black,
  bookmarks=true, bookmarksopen=true, bookmarksnumbered=true,
  pdftitle={Exposé sur LaTeX},
  pdfauthor={Thierry Masson},
  pdfsubject={cours sur LaTeX},
  pdfcreator={TeX}, pdfproducer={pdfTeX},
  pdfkeywords={LaTeX, cours}
}
```

→ possibilité d'ouvrir automatiquement le document en pleines pages, couleurs de différents types de liens, comportement des *bookmarks*, métadonnées sur le document...

De nombreuses autres options disponibles...

.....
Quelques commandes utiles

hyperref fournit les commandes essentielles suivantes :

- `\href{http://www.lesite.fr}{lien}` crée un lien externe à partir du texte “lien” vers la page `http://www.lesite.fr`.
- `\hypertarget{label}{lieu}` crée un but de nom “label” à l’emplacement du texte “lieu”.
`\hyperlink{label}{lien}` crée un lien interne vers le but “label” à partir du texte “lien”.
- `\hyperref[label]{lien}` crée un lien interne vers le but créé par `\label{label}`.
- `\texorpdfstring{nom TeX}{nom PDF}`

Les *bookmarks* de PDF ne peuvent pas contenir des macros `\TeX`.

Cette commande donne une alternative “nom PDF” à “nom TeX”.

Elle doit être utilisée dans les commandes qui produisent les *bookmarks*, comme par exemple `\section{-}`.

L’apparence d’un lien peut être modifiée par des commandes `\LaTeX`, et par les options de `\hypersetup{-}` (pour la couleur, la présence d’un cadre...).

 Il est préférable d’appeler ce *package* en dernier car il modifie de nombreuses macros définies par d’autres *packages*.

.....
Les polices **LATIN MODERN**

Les polices par défaut de `\TeX`, **COMPUTER MODERN**, ou leurs versions plus récentes (**CM-SUPER**), sont insuffisantes d’un point de vue typographique, par manque de nombreux glyphes (lettres accentuées pour l’essentiel).

Les polices **LATIN MODERN** sont composées de 72 fontes de caractères, auxquelles on ajoute 20 fontes pour les mathématiques.

En tout, il y a 58 604 caractères pour un total d’environ 69 000 glyphes...

`\usepackage{lmodern}` charge ces polices dans le document.

An exploration of the Latin Modern fonts⁴⁹ décrit la richesse de ces polices.

Autres packages de polices

Les polices standard de **POSTSCRIPT** sont accessibles à **LATEX**.

Les polices du projet **TEX GYRE**⁵⁰ en sont des clones.

Le *package* (entre parenthèses) correspondant les charge.

TERMES (**tgtermes**) : Il se nommait Javert, et il était de la police.

TIMES (**mathptmx**) : Il se nommait Javert, et il était de la police.

PAGELLA (**tgpage11a**) : Il se nommait Javert, et il était de la police.

PALATINO (**mathpazo**) : Il se nommait Javert, et il était de la police.

BONUM (**tgbonum**) : Il se nommait Javert, et il était de la police.

BOOKMAN-LIGHT (**bookman**) : Il se nommait Javert, et il était de la police.

SCHOLA (**tgschola**) : Il se nommait Javert, et il était de la police.

NEWCENTURYSCHLBK (**newcent**) : Il se nommait Javert, et il était de la police.

ADVENTOR (**tgadventor**) : Il se nommait Javert, et il était de la police.

AVANT GARDE (**avant**) : Il se nommait Javert, et il était de la police.

HEROS (**tgheros**) : Il se nommait Javert, et il était de la police.

HELVETICA (**helvet**) : Il se nommait Javert, et il était de la police.

CURSOR (**tgcursor**) : Il se nommait Javert, et il était de la police.

COURIER (**courier**) : Il se nommait Javert, et il était de la police.

CHORUS (**tgchorus**) : Il se nommait Javert, et il était de la police.

ZAPFCHANCERY (**chancery**) : Il se nommait Javert, et il était de la police.

Autres packages de polices (suite)

Exemples de polices avec l'éventuel *package* qui l'active :

ANTYKWA TORUNSKA (**anttor**) : Il se nommait Javert, et il était de la police.

AREV SANS (**arev**) : Il se nommait Javert, et il était de la police.

ARTEMISIA (**gfsartemisia**) : Il se nommait Javert, et il était de la police.

BERA (**bera**, **beraserif**) : Il se nommait Javert, et il était de la police.

BIOLINUM (**libertine**) : Il se nommait Javert, et il était de la police.

CANTARELL (**cantarell**) : Il se nommait Javert, et il était de la police.

CHARTERBT (**charter**) : Il se nommait Javert, et il était de la police.

CONCRETE (**ccffonts**) : Il se nommait Javert, et il était de la police.

IWONA (**iwona**) : Il se nommait Javert, et il était de la police.

KP-FONTS (**kpfonts**) : Il se nommait Javert, et il était de la police.

LATIN MODERN (**lmodern**) : Il se nommait Javert, et il était de la police.

LATO (**lato**) : Il se nommait Javert, et il était de la police.

LIBERTINE (**libertine**) : Il se nommait Javert, et il était de la police.

LIBRIS ADF STD (**libris**) : Il se nommait Javert, et il était de la police.

UTOPIA (**fourier**) : Il se nommait Javert, et il était de la police.

VENTURIS ADF (**venturis**) : Il se nommait Javert, et il était de la police.

The **LATEX Font Catalogue**⁵¹ expose d'autres polices.

6 – Les mathématiques

Généralités sur les mathématiques en L^AT_EX

- L^AT_EX intègre un “mode mathématique” pour composer les formules de mathématique.
- Deux modes sont possibles :
 - formule dans le texte : `$... $`
Nous avons `$a=1$ et $b=2$.` Nous avons $a = 1$ et $b = 2$.
 - formule centrée hors du texte, avec ou sans numérotation : divers environnements, `equation`, `align`, `multiline`...

```
\begin{equation*}
\phi(x) = 
\sum_{n=0}^{\infty} c_n e^{2i\pi n x} \quad \phi(x) = \sum_{n=0}^{\infty} c_n e^{2i\pi n x}
\end{equation*}
```

Le rendu des deux modes n'est pas exactement le même.

Dans le texte on a $\int_a^b f(x)dx = \lim_{n \rightarrow \infty} a_n$, et hors du texte on a :

$$\int_a^b f(x)dx = \lim_{n \rightarrow \infty} a_n$$

La première formule est composée en *textstyle*, la seconde en *displaystyle*.

Gain d'espace vertical dans le mode texte.

Généralités sur les mathématiques en L^AT_EX (suite)

- En mode mathématique, la typographie change (L^AT_EX utilise une police spécifique).
Ça permet de distinguer a (`a`) de “a”
 - Ne pas oublier le mode mathématique, même pour une seule lettre !
Ça produit aussi l'affreux *Diff* plutôt que le plus acceptable *Diff*.
 - Solution plus loin...
- Les *packages* `amssymb` (symboles et polices supplémentaires), `amsmath` et `mathtools` (commandes supplémentaires) étendent les fonctionnalités de L^AT_EX.
Toujours appeler `amssymb` et `amsmath` lorsqu'on fait des mathématiques.
- Les accents sont interdits en mode mathématique.
Des commandes spécifiques sont fournies.
- Exposants et indices :
`a^2` → a^2 , `a_2` → a_2 , `a^2_n` → a_n^2 , `e^{i 2 \pi}` → $e^{i2\pi}$
- Lettres grecques :
`\alpha \Gamma \pi` → $\alpha\Gamma\pi$

Généralités sur les mathématiques en L^AT_EX (suite)

- Symboles divers :
 $\backslashoplus \backslashcdot \backslashotimes \rightarrow \oplus \cdot \otimes$
 $\backslashinfty \backslashhbar \backslashnabla \backslashimath \rightarrow \infty \hbar \nabla i$
 $\backslashvdots \backslashldots \backslashddots \rightarrow \vdots \ldots \ddots$
- Fonctions usuelles :
 $\backslashcos(x) + i \backslashsin(x) = \backslashexp(ix) \rightarrow \cos(x) + i \sin(x) = \exp(ix)$
- Éléments de formules :
 $\backslashsqrt{16} = 4 \rightarrow \sqrt{16} = 4$
 $\backslashsqrt[3]{8} = \backslashfrac{4}{2} \rightarrow \sqrt[3]{8} = \frac{4}{2}$
 $\backslashint_a^b f(x) dx \rightarrow \int_a^b f(x) dx$
 $\backslashsum_{n \in \mathbb{Z}} c_n \rightarrow \sum_{n \in \mathbb{Z}} c_n$
- La commande `\not` place une négation devant certains symboles :
 $\$ \backslashnot < \backslashnot \backslashequiv \backslashnot \backslashle \backslashnot \backslashsubseteqq \$ \rightarrow \not< \not\equiv \not\leq \not\subseteqq$
- Barres et flèches :
 $\backslashoverline{x + iy} \rightarrow \overline{x + iy}$
 $\backslashoverrightarrow{AB} \rightarrow \overrightarrow{AB}$

Généralités sur les mathématiques en L^AT_EX (suite)

- L'environnement `array` pour les tableaux mathématiques :

```
\begin{array}{cc}
a & b \\
c & d
\end{array}
```

- Le mode mathématique définit un certain nombre d'espaces :

$\backslashthinspace, \backslash, \Rightarrow \Leftarrow$	$\backslashnegthinspace, \backslash! \Rightarrow \Leftarrow$
$\backslashmedspace, \backslash: \Rightarrow \Leftarrow$	$\backslashnegmedspace \Rightarrow \Leftarrow$
$\backslashthickspace, \backslash; \Rightarrow \Leftarrow$	$\backslashnegthickspace \Rightarrow \Leftarrow$
$\backslashenskip \Rightarrow \Leftarrow$	
$\backslashquad \Rightarrow \Leftarrow$	
$\backslashqquad \Rightarrow \Leftarrow$	

- On peut forcer l'apparence *textstyle* ou *displaystyle* avec les commandes

`\textstyle` et `\displaystyle`:

$\$ \backslashsum_{k \geq 0} \backslashfrac{1}{k^2} \$ \rightarrow \sum_{k \geq 0} \frac{1}{k^2}$

$\$ \backslashdisplaystyle \backslashsum_{k \geq 0} \backslashfrac{1}{k^2} \$ \rightarrow \sum_{k \geq 0} \frac{1}{k^2}$

Utile par exemple dans les cellules d'un tableau `array`.

.....
Tables de symboles mathématiques

Les symboles de l'AMS requièrent le *package* **amsmath** (qui charge **amssymb**).

Symboles de ponctuation

, , ; ; \colon : \ldotp \cdotp \cdot

Accents en mode mathématique

\hat{a}	\acute{a}	\bar{a}	\dot{a}
\check{a}	\grave{a}	\vec{a}	\ddot{a}
\breve{a}	\tilde{a}	\mathring{a}	

Accents extensibles

\widetilde{abcde}	\widehat{abcde}
\overline{abcde}	\underline{abcde}
\overbrace{abcde}	\underbrace{abcde}
\overleftarrow{abcde}	\overrightarrow{abcde}

Accents extensibles AMS

\underleftarrow{abcde}	\overleftarrow{abcde}
\underrightarrow{abcde}	\underleftarrow{abcde}

Lettres grecques

\alpha α	\theta θ	\o \o	\tau τ
\beta β	\vartheta ϑ	\pi π	\upsilon υ
\gamma γ	\iota ι	\varpi ϖ	\phi ϕ
\delta δ	\kappa κ	\rho ρ	\varphi φ
\epsilon ϵ	\lambda λ	\varrho ϱ	\chi χ
\varepsilon ε	\mu μ	\sigma σ	\psi ψ
\zeta ζ	\nu ν	\varsigma ς	\omega ω
\eta η	\xi ξ		
\Gamma Γ	\Lambda Λ	\Sigma Σ	\Psi Ψ
\Delta Δ	\Xi Ξ	\Upsilon Υ	\Omega Ω
\Theta Θ	\Pi Π	\Phi Φ	

Lettres grecques AMS

\digamma \digamma \varkappa \varkappa

Lettres hébraïques AMS

\beth \beth \daleth \daleth \gimel \gimel

Opérateurs

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

Opérateurs AMS

```
\injlim \projlim \varliminf \varlimsup \varinjlim \varprojlim
```

Symboles divers

\ldots	\cdots	\vdots	\ddots
\aleph	\prime	\forall	\infty
\hbar	\emptyset	\exists	\imath
\nabla	\neg	\surd	\jmath
\flat	\triangle	\ell	\top
\natural	\clubsuit	\wp	\bot
\sharp	\diamondsuit	\Re	\parallel
\backslash	\heartsuit	\Im	\angle
\partial	\spadesuit	\dots	\mid

Symboles divers AMS

\hbar \hbar	\hslash \hslash	\vartriangle \vartriangle
\triangledown \triangledown	\square \square	\lozenge \lozenge
\circledS \circledS	\angle \angle	\measuredangle \measuredangle
\nexists \nexists	\mho \mho	\Finv \Finv
\Game \Game	\Bbbk \Bbbk	\backprime \backprime
\varnothing \varnothing	\blacktriangle \blacktriangle	\blacktriangledown \blacktriangledown
\blacksquare \blacksquare	\blacklozenge \blacklozenge	\bigstar \bigstar
\sphericalangle \sphericalangle	\complement \complement	\eth \eth
\diagup \diagup	\diagdown \diagdown	

Symboles de tailles variables

\sum \sum	\bigcap \bigcap	\bigodot \bigodot
\prod \prod	\bigcup \bigcup	\bigotimes \bigotimes
\coprod \coprod	\bigsqcup \bigsqcup	\bigoplus \bigoplus
\int \int	\bigvee \bigvee	\biguplus \biguplus
\oint \oint	\bigwedge \bigwedge	

Opérations binaires

\pm	\cap	\diamond	\oplus
\mp	\cup	\bigtriangleup	\ominus
\times	\uplus	\bigtriangledown	\otimes
\div	\sqcap	\triangleleft	\oslash
$*$	\sqcup	\triangleright	\odot
\star	\vee	\bigcirc	\circ
\wedge	\dagger	\bullet	\setminus
\ddagger	\cdot	\wr	\amalg
$+$	$-$		

Opérations binaires AMS

\dotplus	\smallsetminus	\Cap
\Cup	\barwedge	\veebar
\doublebarwedge	\boxminus	\boxtimes
\boxdot	\boxplus	\divideontimes
\ltimes	\rtimes	\leftthreetimes
\rightthreetimes	\curlywedge	\curlyvee
\circleddash	\circledast	\circledcirc
\centerdot	\intercal	

Relations binaires

\leq	\geq	\equiv	\models
\prec	\succ	\sim	\perp
\preceq	\succeq	\simeq	\mid
\ll	\gg	\asymp	\parallel
\subset	\supset	\approx	\bowtie
\subseteq	\supseteq	\cong	\neq
\smile	\sqsubseteq	\sqsupseteq	\doteq
\frown	\in	\ni	\propto
$=$	\vdash	\dashv	$<$
$>$	$:$		

Relations binaires AMS

\leqq	\leqslant	\eqslantless
\lessapprox	\lessapprox	\approxeq
\lessdot	\lessdot	\lessgtr
\lesseqgtr	\lesseqgtr	\doteqdot
\risingdotseq	\fallingdotseq	\backsim
\backsimeq	\subsetneq	\Subset
\sqsubset	\preccurlyeq	\curlyeqprec
\precsim	\precapprox	\vartriangleleft
\trianglelefteq	\vDash	\Vdash
\smallsmile	\smallfrown	\bumpeq
\Bumpeq	\geqq	\geqslant
\eqslantgtr	\gtrsim	\gtrapprox
\gtrdot	\ggg	\gtrless
\gtreqless	\gtreqless	\eqcirc
\circeq	\triangleq	\thicksim
\thickapprox	\supseteqq	\Supset
\sqsupset	\succcurlyeq	\curlyeqsucc
\succsim	\succapprox	\vartriangleright
\trianglerighteq	\Vdash	\shortmid
\shortparallel	\between	\pitchfork
\varpropto	\blacktriangleleft	\therefore
\backepsilon	\blacktriangleright	\because

Relations binaires négatives AMS

\nless	\nleq	\neq
\nleqq	\nleqslant	\neqq
\nvertneqq	\nnsim	\napprox
\nprec	\npreceq	\nprecsim
\precnapprox	\nsim	\nshortmid
\nmid	\nvDash	\nvDash
\ntriangleleft	\ntrianglelefteq	\nsubseteq
\subsetneqq	\nsubsetneq	\subsetneqq
ς	\ngtr	\ngeq
\ngeqslant	\ngeq	\gneq
\gneqq	\gvertneqq	\gnsim
\napprox	\nsucc	\nsucc
\nsucc	\succnsim	\succnapprox
\ncong	\nshortparallel	\nparallel
\nvDash	\nVDash	\ntriangleright
\ntrianglerighteq	\nsupseteq	\nsupseteqq
\supsetneqq	ς	\supsetneqq
ς	\nsubsetneqq	\nsubsetneqq

Flèches

\leftarrow	\leftarrow	\uparrow
\Leftarrow	\Leftarrow	\Updownarrow
\rightarrow	\rightarrow	\downarrow
\Rightarrow	\Rightarrow	\Downarrow
\leftrightarrow	\leftrightarrow	\Updownarrow
\Leftrightarrow	\Leftrightarrow	\Updownarrow
\mapsto	\mapsto	\nearrow
\hookleftarrow	\hookleftarrow	\searrow
\leftharpoonup	\leftharpoonup	\swarrow
\leftharpoondown	\leftharpoondown	\nwarrow
\rightleftharpoons		

Flèches AMS

\dashrightarrow	\dashleftarrow	\leftleftarrows
\leftrightsquigarrow	\leftarrow	\twoheadleftarrow
\leftarrowtail	\looparrowleft	\leftrightharpoons
\curvearrowleft	\circlearrowleft	\Lsh
\upuparrows	\upharpoonleft	\downharpoonleft
\multimap	\leftrightsquigarrow	\rightrightarrows
\rightleftarrows	\rightleftarrows	\rightleftarrows
\twoheadrightarrow	\rightarrowtail	\looparrowright
\leftrightharpoons	\curvearrowright	\circlearrowright
\Rsh	\downdownarrows	\upharpoonright
\downharpoonright	\rightsquigarrow	

Flèches négatives AMS

\nleftarrow	\nrightarrow	\nLeftarrow
\nrightarrow	\nleftrightsquigarrow	\nLeftrightarrow

Délimiteurs de tailles variables

$($	$)$	$/$	$/$	\uparrow
$[$	$]$	\backslash	\downarrow	
$\{$	$\}$	$ $	$ $	\uparrow
\langle	\rangle	$\backslash $	$ $	\downarrow
\lfloor	\rfloor	$\backslash $	$ $	\updownarrow
\lceil	\rceil			\Updownarrow

Délimiteurs de tailles variables AMS

\lvert	\rvert	\lVert	\rVert
----------	----------	----------	----------

Le chapitre 3 de **Comprehensive L^AT_EX symbol list**⁶² donne une liste très complète de tous les symboles mathématiques utilisables avec L^AT_EX.

Commandes générales de typographie

Dans le mode mathématique, on dispose des commandes suivantes pour changer de style :

Commande	Description	Exemple
<code>\mathnormal{-}</code>	défaut	<i>abcXYZ</i>
<code>\mathrm{-}</code>	romane	<i>abcXYZ</i>
<code>\mathbf{-}</code>	romane épaisse	abcXYZ
<code>\mathsf{-}</code>	<i>sans serif</i>	<i>abcXYZ</i>
<code>\mathit{-}</code>	italique	<i>abcXYZ</i>
<code>\mathtt{-}</code>	<i>typewriter</i>	<i>abcXYZ</i>
<code>\mathcal{-}</code>	calligraphique	<i>XYZ</i>
<code>\mathscr{-}</code>	calligraphique	<i>XYZ</i>
<code>\mathbb{-}</code>	<i>black board</i>	\mathbb{XYZ}
<code>\mathfrak{-}</code>	gothique	\mathfrak{abcXYZ}

`\mathbb{-}` et `\mathfrak{-}` sont définies par le package **amsfonts**.

`\mathscr{-}` est définie avec l'option **mathscr** du package **euscript**.

Les arguments de ces commandes sont en mode mathématique :

```
$\mathrm{E=mc^2}$, $\mathbf{E=mc^2}$, $\mathfrak{E=mc^2}$  
E =  $mc^2$ , E =  $mc^2$ , E =  $mc^2$ 
```

Du texte à l'intérieur des mathématiques : `\text`

Il faut souvent placer des morceaux de texte à l'intérieur du mode mathématique.

⚠ Il ne faut pas utiliser `\mathrm{-}` !

Le package **amsmath** fournit la commande `\text{-}` à cet usage.

- `\text{-}` reprend la fonte actuellement utilisée dans le texte environnant.
- `\text{-}` s'adapte à la taille, par exemple s'il est placé en indice ou exposant.
- `\text{-}` accepte du texte avec des accents.

```
\sffamily\bfseries Diff $=\text{Diff}$  
Diff = Diff
```

```
$x^2 \geq 0 \text{ pour tout } x \in \mathbb{R}$  
 $x^2 \geq 0$  pour tout  $x \in \mathbb{R}$ 
```

```
$x^2 \geq 0 \text{ pour tout } x \in \mathbb{R}$  
 $x^2 \geq 0$  pour tout  $x \in \mathbb{R}$ 
```

```
$M_{\text{Higgs}}$ est la masse du Higgs  
MHiggs est la masse du Higgs
```

La graisse épaisse en mode mathématique

⚠ `\mathbf{-}` ne sert pas à mettre en graisse épaisse des formules mathématiques.

- Elle change la forme des caractères : b (`b`) est dans la forme italique alors que **b** (`(\mathbf{b})`) est dans la forme droite.
- Elle ne donne pas accès aux symboles mathématiques en graisse épaisse :
 $\$ \mathbf{\alpha} + b \$$ donne $\alpha + b$.

On peut utiliser la commande `\boldsymbol{-}` du package **amsmath** :

$$\$ \boldsymbol{b+A\infty\Gamma} \$$$

$$\neq b+A\infty\Gamma$$

Si le symbole n'a pas de version épaisse définie (\mathbb{N} ci-dessus), on peut utiliser `\pmb{-}` (*poor man's bold*) définie par **amsmath** :

$$\$ \boldsymbol{b+A\infty\Gamma} \$$$

$$\neq \pmb{b+A\infty\Gamma}$$

Le résultat avec `\pmb{-}` est moins esthétique qu'avec `\boldsymbol{-}`.

Ces commandes requièrent de modifier le code mathématique.

→ À n'utiliser que pour des symboles particuliers.

La graisse épaisse en mode mathématique (suite)

La commande `\mathversion{bold}` active la graisse épaisse en mathématique sans avoir à modifier le code mathématique.

Il faut l'utiliser hors du mode mathématique.

Son contraire est `\mathversion{normal}`.

$$\$ \mathversion{bold} \mathfrak{X} = \alpha + b = \Gamma \div D \$,$$

$$\$ \mathversion{normal} \mathfrak{X} = \alpha + b = \Gamma \div D \$$$

$$\mathfrak{X} = \alpha + b = \Gamma \div D$$

La graisse épaisse activée par `\mathversion{bold}` est la même que celle utilisée par `\boldsymbol{-}`

`\bfseries\mathversion{bold}` passe le texte et les mathématiques en gras :

$$\mathbf{La relation } e^{i\pi} + 1 = 0 \mathbf{ est au cœur des mathématiques.}$$

$$\mathbf{La relation } e^{i\pi} + 1 = 0 \mathbf{ est au cœur des mathématiques.}$$

.....
Symboles extensibles verticalement

LATEX propose une série de symboles extensibles verticalement.

- La taille finale peut être gérée manuellement avec les commandes `\big`, ..., `\Bigg` :

$$] \rightarrow] \ \backslash \big \] \rightarrow] \ \backslash \Big \] \rightarrow] \ \backslash \bigg \] \rightarrow] \ \backslash \Bigg \] \rightarrow]$$

- La taille peut être obtenue automatiquement, avec les commandes `\left ... \middle ... \right`

`\left` est suivi d'un symbole extensible placé en ouverture,
`\right` est suivi d'un symbole extensible placé en fermeture,
`\middle` (optionnel) est suivi d'un symbole extensible placé au milieu.
La taille verticale est calculé par le contenu entre l'ouverture et la fermeture.
Le délimiteur “.” (un point) peut être utilisé après `\left` ou `\right` pour ouvrir ou fermer sans placer de symbole.

$$\$ \left\langle \sum_n \right. \left. \prod_p \right\rangle . \$$$

$$\left\langle \sum_n \right. \left. \prod_p \right\rangle .$$

Des *packages* peuvent définir d'autres symboles extensibles verticalement.

.....
Symboles extensibles horizontalement

LATEX propose une série de symboles “surlignant” ou “soulignant” extensibles horizontalement.

Certains de ces symboles ont une extension “infinie”, d'autres non.

`\overbrace{-}`, `\underbrace{-}` acceptent des exposants et des indices :

$$\$ \underbrace{a+b}_{\text{plus}} _{\text{plus}} \text{ } \underbrace{c-d}^{\text{moins}} _{\text{moins}} \$$$

$$a + b = c - d$$

Le *package* `mathtools` définit les symboles extensibles suivants :

$$\overbracket{-} \{ - \} \overbrace{abcde} \underbracket{-} \{ - \} \underbrace{abcde}$$

D'autres *packages* proposent d'autres symboles extensibles.

.....
Flèches extensibles

amsmath définit des flèches extensibles :

```
$A \xleftarrow[\text{en dessous}]{\text{vers la gauche}} B
\xrightarrow[\text{en dessous}]{\text{vers la droite}} C$
```

$$A \xleftarrow[\text{en dessous}]{\text{vers la gauche}} B \xrightarrow[\text{en dessous}]{\text{vers la droite}} C$$

mathtools ajoute les flèches suivantes :

$\xleftrightarrow[-]{-}$	$\xleftarrow[\text{en haut}]{\text{en bas}}$	$\xhookleftarrow[-]{-}$	$\xleftarrow[\text{en haut}]{\text{en bas}}$
$\xLeftarrow[-]{-}$	$\xleftarrow[\text{en haut}]{\text{en bas}}$	$\xhookrightarrow[-]{-}$	$\xrightarrow[\text{en haut}]{\text{en bas}}$
$\xRightarrow[-]{-}$	$\xrightarrow[\text{en haut}]{\text{en bas}}$	$\xmapsto[-]{-}$	$\xrightarrow[\text{en haut}]{\text{en bas}}$
$\xLeftrightarrow[-]{-}$	$\xrightleftharpoons[\text{en haut}]{\text{en bas}}$		

.....
Les matrices

Pour les matrices, **matrix** (de **amsmath**) est plus optimal que **array**.

Réserver **array** aux tableaux dont les entrées sont en mode mathématique.

```
\begin{matrix}
a & b \\
c & d
\end{matrix}
```

$$\begin{matrix} a & b \\ c & d \end{matrix}$$

pmatrix, **bmatrix**, **Bmatrix**, **vmatrix** et **Vmatrix** produisent respectivement :

$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$	$\left\{ \begin{matrix} a & b \\ c & d \end{matrix} \right\}$	$\left \begin{matrix} a & b \\ c & d \end{matrix} \right $	$\left\ \begin{matrix} a & b \\ c & d \end{matrix} \right\ $
--	--	---	---	---

smallmatrix permet de créer des petites matrices pour le texte :

```
$A=\left(\begin{smallmatrix} a&b\\c&d \end{smallmatrix}\right)$
```

$$A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix} \right)$$

Les matrices (suite)

Pour insérer des pointillés, on dispose des commandes

`\vdots` → \vdots , `\cdots` → \cdots , `\ddots` → \ddots

Ces pointillés n'occupent qu'une seule cellule de la matrice.

\hdotsfor[-]{-} permet de placer des pointillés sur plusieurs colonnes :

```
\begin{vmatrix}
  a & b & c & d & e & \\
  f & \hdotsfor{3} & g & \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \\
  h & i & j & k & l & \\
\end{vmatrix}
```

L'argument optionnel sert à ajuster l'espacement horizontal entre les points (facteur multiplicatif).

L'environnement cases

cases permet de présenter des “cas” :

```
\begin{equation}
A = \begin{cases}
B & \text{si } a < 0 \\
C & \text{si } a \geq 0
\end{cases}
\end{equation}
```

mathtools définit les environnements `dcases`, `dcases*`, `rcases`, `rcases*`, `drcases`, `drcases*`, `cases*`.

.
Commandes diverses

– Fractions.

Utiliser la commande `\frac{-}{-}`.

⚠ Ne pas utiliser la commande `\over` obsolète.

Des variantes de cette commande sont disponibles :

`\dfrac{-}{-}` force le *displaystyle* et `\tfrac{-}{-}` force le *textstyle*.

$$A = \frac{1}{2} = \dfrac{1}{2} = \tfrac{1}{2}$$

– Coefficients binomiaux.

$$A = \binom{n}{p} = \dbinom{n}{p} = \tbinom{n}{p}$$

– Intégrales multiples.

$$\begin{aligned} & \text{\displaystyle } \int_X f(x) dx = \int_Y g(y) dy \\ & = \int_Z h(z) dz = \dots \int_T r(t) dt \\ & \int_X f(x) dx = \iint_Y g(y) dy = \iiint_Z h(z) dz = \int \dots \int_T r(t) dt \end{aligned}$$

.
Commandes diverses (suite)

– Placer un symbole au dessus ou au dessous d'un autre.

`\overset{-}{-}` et `\underset{-}{-}` permettent de superposer des symboles :

$$\overset{*}{X} \xrightarrow{f} \underset{*}{Y}$$

– Superpositions.

`\substack{-}` superpose des expressions, pour des sommes par exemple.

Le résultat est de la taille des indices et exposants :

$$\begin{aligned} & \text{\displaystyle } \sum_{\substack{0 < i < m \\ 0 < j < n}} P_{i,j} \\ & = \substack{a \\ b} \\ & \sum_{\substack{0 < i < m \\ 0 < j < n}} P_{i,j} = \substack{a \\ b} \end{aligned}$$

– Commandes pour les “modulos”.

$$\text{\mod } m \equiv n \pmod p \quad \text{\bmod } m \equiv n \pmod p \quad \text{\pmod } m \equiv n \pmod p$$

.....
Définir un nouvel opérateur

Les commandes de type opérateurs ont un comportement spécifique (espace, fontes) :

```
$r \sin \theta$  $r \sin \theta$       $r \sin \theta$  $r \sin \theta$   

$ \text{sin} \theta$  $\text{sin} \theta$       $r \mathbf{sin} \theta$  $r \sin \theta$ 
```

Utiliser `\DeclareMathOperator{-}{-}` (préambule) pour définir un tel opérateur.

```
\DeclareMathOperator{\antisin}{antisin}  

On pose  $\text{antisin}(x) = \cos(x)$ .  

On pose  $\text{antisin}(x) = \cos(x)$ .
```

`\DeclareMathOperator*{-}{-}` déclare un opérateur de type “lim” :

```
\DeclareMathOperator*{\antimax}{antimax}  

\$ \text{antimax}_a \in A = \min_a \in A \quad \text{qqquad}  

\$ \text{\displaystyle antimax}_a \in A = \min_a \in A \$  

\text{antimax}_{a \in A} = \min_{a \in A} \quad \text{antimax} = \min_{a \in A}
```

.....
Les environnement d'équations “hors texte”

Le package **amsmath** définit la règle suivante pour les équations hors texte :

- Un environnement sans étoile : l'équation est numérotée.
- Un environnement avec étoile : l'équation n'est pas numérotée.

Il est possible de placer une commande `\label{-}` dans un environnement d'équation sans étoile pour faire référence ailleurs dans le texte au numéro de l'équation.

`\ref{-}` rappelle le numéro, `\eqref{-}` place en plus des parenthèses autour.

`equation` sert à composer une simple équation, sur une ligne.

```
\begin{equation}  

A+B=C \label{eq:a} \quad A + B = C \quad (2)  

\end{equation}  

Label : \eqref{eq:a}, \ref{eq:a}.      Label : (2), 2.
```

```
\begin{equation*}  

A+B=C \quad A + B = C  

\end{equation*}
```

T_EX (pur, avant **L_AT_EX**) définit **\$\$...\$\$** pour les formules hors texte.

L_AT_EX définit les commandes `\[... \]` et l'environnement **displaymath**

⚠ Bannir ces commandes : mauvaise gestion des espaces.

.....
L'environnement **gather**

gather permet de rassembler plusieurs équations, sans alignements. Chacune est numérotée, sauf si `\nonumber` (ou `\notag`) est utilisé.

```
\begin{gather}
A+B=B \label{gat:a}\\
C=D+E \nonumber \\
E=F \label{gat:c}\\
G+H=I+J \notag
\end{gather}
\eqref{gat:a} et \eqref{gat:c}. (3) et (4).
```

$$\begin{aligned} A + B &= B & (3) \\ C &= D + E \\ E &= F & (4) \\ G + H &= I + J \end{aligned}$$

```
\begin{gather*}
A+B=B \\
C+D=E+F+G
\end{gather*}
```

$$\begin{aligned} A + B &= B \\ C + D &= E + F + G \end{aligned}$$

.....
L'environnement **align**

align rassemble plusieurs équations, avec alignements verticaux sur les caractères `&`.

```
\begin{align}
A+B &= B+A \label{al:a}\\
C &= D+E \nonumber \\
E &= F \label{al:c}
\end{align}
\eqref{al:a} et \eqref{al:c}. (5) et (6).
```

$$\begin{aligned} A + B &= B + A & (5) \\ C &= D + E \\ E &= F & (6) \end{aligned}$$

L'alignement peut s'effectuer sur plusieurs équations en ligne :

```
\begin{align*}
A+B &= C & B &= C-A & A &= C-B \\
D &= E+F & D-E &= F & D-F &= E
\end{align*}
```

$$\begin{aligned} A + B &= C & B &= C - A & A &= C - B \\ D &= E + F & D - E &= F & D - F &= E \end{aligned}$$

Un coup sur deux, le rôle des `&` est différent : alignement ou séparation des colonnes.

⚠ Bannir l'environnement `eqnarray` de `LATeX` au profit de `align`.
Problèmes d'espacements horizontaux, mauvais placement des numéros d'équations...

.....
L'environnement `multline`

`multline` permet de présenter une équation sur plusieurs lignes :
la première ligne est collée à gauche, les suivantes sont centrées,
la dernière est collée à droite.

Un seul numéro est attribué à l'équation.

```
\begin{multline}
A+B-C+D-E+F = \\ A + B - C + D - E + F = \\
-G+H+I-J+K-L \\ - G + H + I - J + K - L \\
+M-N+O-P+Q-R+S \\ + M - N + O - P + Q - R + S \\
-T+U-V+W-X+Y-Z \\ - T + U - V + W - X + Y - Z \quad (7)
\end{multline}
```

`\shoveright{-}` et `\shoveleft{-}` forcent l'alignement droit ou gauche.

```
\begin{multline*}
A+B-C+D-E+F = \\ A + B - C + D - E + F = \\
\shoveright{-G+H+I-J} \\ - G + H + I - J \\
\shoveleft{+K-L+M-N} \\ + K - L + M - N \\
+O-P+Q-R+S-T \\ + O - P + Q - R + S - T
\end{multline*}
```

Voir **amsmath** et **mathools** pour d'autres environnements.

.....
Décorations de formules

La commande `\boxed{-}` de **amsmath** permet d'encadrer une formule ou un morceau.

$$\boxed{E=mc^2}$$

$$E = \boxed{mc^2}$$

Le package **empheq** permet plus d'options pour décorer des formules.

```
\begin{empheq}[box=\fbox]{align}
E &= h \nu \notag \\
E &= mc^2
\end{empheq} \quad (8)
```

```
\begin{empheq}%
[left=A \Rightarrow \empheqbrace,
innerbox=\fbox]{align*}
B &= C \\
D &= E
\end{empheq}
```

$$A \Rightarrow \begin{cases} B = C \\ D = E \end{cases}$$

empheq reconnaît tous les environnements définis par **amsmath**.

.....

Le package **amsthm** : définition d'environnements

LAT_EX permet de définir des environnements pour présenter et numéroter automatiquement des théorèmes, des propositions, des définitions...

Le package **amsthm** améliore les commandes de **LAT_EX**.

`\newtheorem{lem}{Lemme}`

définit un nouvel environnement **lem** dont le titre est “Lemme”.

`\newtheorem{thm}{Théoreme}`

`\newtheorem{prop}[thm]{Proposition}`

définit **thm**, puis **prop**, dont le compteur de numérotation est le même que celui de **thm**.

`\newtheorem*[pyth]{Théorème de Pythagore}`

définit **pyth** qui ne sera pas numéroté. Utile pour des théorèmes spécifiques.

`\newtheorem{defn}{Définition}[section]`

définit **defn** dont le compteur de numérotation est remis à 0 à chaque début de section.

L'environnement **proof** sert aux démonstrations.

La commande `\qedsymbol` place un symbole de fin de démonstration (personnalisable).

.....

Le package **amsthm** : le style des environnements

On peut choisir un style avant de définir un “théorème” : **plain**, **definition**, **remark**.

`\theoremstyle{plain} % style par défaut`

`\newtheorem{theorem}{Theorem}[section]`

`\newtheorem{proposition}[theorem]{Proposition}`

`\theoremstyle{definition}`

`\newtheorem{example}[theorem]{Example}`

`\newtheorem{definition}[theorem]{Definition}`

`\theoremstyle{remark}`

`\newtheorem{remark}[theorem]{Remark}`

définit 5 environnements avec 3 styles différents.

`\newtheoremstyle{note}{2pt}{3pt}{\bfseries}{%`

`{2cm}{\sffamily}{:}{2em}{}`

définit un nouveau style “note” (voir la documentation pour la syntaxe).

.....

Le package `siunitx`

`siunitx` définit des commandes pour présenter les nombres et leurs unités physiques selon les standards (SI).

```
\num{123} \num{12345} \num{0,1234} \num{.1234} \num{1.2d-7}
123      12 345      0,1234      0,1234      1,2 × 10-7
```

```
\ang{45} \ang{32.7} \ang{1;2;3} \ang{;;5}
45°      32,7°      1°2'3"      5"
```

```
\si{kg.m/s^2} \si{\kilo\gram\metre\per\second\square}
kg m/s2      kg m s-2
```

Toutes les unités du SI sont connues, et les préfixes, de `\yocto` (10^{-24}) à `\yotta` (10^{24})... Il est possible de définir d'autres unités.

Un type de colonnes est proposé pour l'alignement de nombres dans les tableaux.

Le *package* prend en compte les spécificités de différentes langues :

`\sisetup{locale=FR}` passe en français.

→ Possible d'automatiser le basculement avec celui de `babel`.

.....

Le choix des polices du mode mathématique

Le fichier **Comprehensive L^AT_EX symbol list**⁵² montre comment accéder à de nombreux symboles mathématiques non définis par défaut.

Quelques *packages* activent d'autres polices de symboles globalement :

- **eulervm** : il charge des polices mathématiques basées sur les polices **EULER** et **COMPUTER MODERN**. Il laisse le choix de la police de texte.
- **MnSymbol** : il charge des polices mathématiques en harmonie avec la police **MINIONPRO**. Mais il peut être utilisé avec d'autres polices de texte.
- **mathdesign**, **fourier**, **mathpazo**, **mathptmx**, **kpfonts**, **arev**... : ces *packages* chargent des polices de mathématiques en même temps que les polices de texte.

Consulter aussi le site **A Survey of Free Math Fonts for T_EX and L^AT_EX**⁵².

7 – Un peu de technique

Définir une commande sans argument

La commande `\newcommand{-}{-}` permet de définir des commandes.

- Le premier argument est le nom de la commande à définir.
- Le second argument est le code définissant cette commande.

Le code peut être (presque) n'importe quoi.

→ On peut y stocker du texte tout bête ou produire de grandes choses !

`\newcommand{\qed}{Quod erat demonstrandum}`

Cette commande produit “Quod erat demonstrandum” lorsqu'elle est invoquée.

`\newcommand{\aspirine}{acide acétylsalicylique}`

L'aspirine est de l'`\aspirine`. → L'aspirine est de l'acide acétylsalicylique.

Si une commande est déjà définie, on peut la redéfinir en utilisant `\renewcommand` à la place de `\newcommand`, avec la même syntaxe.

Définir une commande avec arguments

Pour définir une commande avec arguments, on utilise la syntaxe :

`\newcommand{-}{-}{-}` où `[-]` contient le nombre d'arguments.

Une commande ne peut pas accepter plus de 9 arguments.

Les arguments sont utilisés dans le code sous forme de `#1` à `#9`.

On peut utiliser plusieurs fois le même argument.

`\newcommand{\bonjour}[1]{Bonjour #1, comment allez-vous ?}`
`\bonjour{Thierry}` → Bonjour Thierry, comment allez-vous ?

`\newcommand{\EstAmiDe}[2]{#1 est un ami de #2.}`
`\EstAmiDe{Laurel}{Hardy}` → Laurel est un ami de Hardy.

`\newcommand{\TexteImportant}[1]{\textbf{\large #1}}`
`I1 est \TexteImportant{important} de travailler.`
→ Il est important de travailler.

.....
Définir un environnement

La commande `\newenvironment{-}{-}{-}` permet de définir un nouvel environnement :

- Le premier argument est le nom de l'environnement *monenv*.
- Le second argument est le code exécuté par `\begin{monenv}`.
- Le troisième argument est le code exécuté par `\end{monenv}`

```
% définition
\newenvironment{centerbold}{%
  {\begin{center}\bfseries}%
  {\end{center}}}

% utilisation
\begin{centerbold}Ceci est centré\end{centerbold}
Ceci est centré
et en gras
```

`\newenvironment{-}{-}{-}` définit un environnement avec arguments.

.....
Les dimensions

TEX peut manipuler des dimensions dans différentes unités :

mm	millimètre	cm	centimètre	in	inch	pt	point ⁵³
bp	<i>big point</i>	pc	pica ⁵⁴	dd	point Didot ⁵⁵	cc	unité Cicéro ⁵⁶
sp	<i>scaled point</i>	ex	hauteur de “x”	em	largeur de “M”	mu	<i>math unit</i>

Quelques facteurs de conversions :

lin = 2.54cm = 72.27pt = 72bp ; 1pc = 12pt ; 1157dd = 1238pt ; 1cc = 12dd

Les unités **em** et **ex** dépendent de la fonte de caractères courante :

	1em	1ex	1pt	1cm	lin
\tiny	—	—	—	—	—
\normalsize	—	—	—	—	—
\Large	—	—	—	—	—

En plus des dimensions fixes, **TEX** manipule des dimensions “élastiques”.

Leur syntaxe est de la forme **2cm plus 2mm minus 3mm**.

Ces “marges de manœuvre” sont utilisées pour ajuster les espaces horizontaux et verticaux.

Les dimensions sont des commandes qu'on ne peut pas invoquer seules :

`\parindent`, `\parsep`, `\baselineskip`, `\ linewidth`, `\paperwidth`, `\textheight`,
`\tabcolsep`, `\topmargin`, `\itemsep`, `\labelsep`...

..... Jouer avec les longueurs

LATEX fournit des commandes pour gérer les dimensions.

`\newlength{\len}` définit la nouvelle commande de dimension `\len`.

Des opérations simples sont possibles :

`\setlength{\len}{<dim>}` assigne la valeur `<dim>` à `\len`,

`\addtolength{\len}{<dim>}` ajoute `<dim>` à `\len`.

Il est possible d'afficher une dimension avec la commande `\the` :

```
\newlength{\len} \setlength{\len}{3cm} \the\len,
\addtolength{\len}{1mm plus 2pt minus 3pc}\the\len
85.35826pt, 88.20352pt plus 2.0pt minus 36.0pt
```

En interne, **TeX** travaille en points, et donc les dimensions sont affichées dans cette unité.

Il existe une dimension élastique particulière : `\fill` peut aller de 0pt à ∞ .

`\the\fill` \rightarrow 0.0pt plus 1.0fill

Cette “dimension” occupe l'espace disponible, comme un ressort en extension.

On peut augmenter la “dureté” du ressort.

La commande `\stretch{nbr}` vaut `nbr` fois `\fill` où `nbr` est un nombre entier :

`\the\stretch{4}` \rightarrow 0.0ptplus4fill

..... Faire de l'espace horizontalement

TeX ignore les espaces entre objets. Il existe des commandes pour créer de vrais espaces.

`\hspace{<dim>}` produit un espace horizontal de dimension `<dim>` :

`a b\hspace{1cm}c` \rightarrow a b c (1cm = 

En début de ligne, l'espace est ignoré.

On peut le forcer avec la version étoilée : `\hspace*{<dim>}`.

Des espacements horizontaux de taille prédéterminée sont fournis :

`\quad` est un espace de 1em : a b

`\enspace` est un espace de la moitié de `\quad` : a b

`\quad` est un espace du double de `\quad` : a b

`\hfill` est un espace de dimension `\fill` :

a b

(l'espace entre a et b est le plus grand possible, `\hfill = \hspace{\fill}`)

`\stretch{2}` est deux fois plus “puissant” que `\stretch{1}` ($=\fill$) :

`a\hspace{\stretch{1}}b\hspace{\stretch{1}}c`

a b c

`a\hspace{\stretch{1}}b\hspace{\stretch{2}}c`

a b c

.....

Faire de l'espace verticalement

`\vspace{\dim}` ajoute un espace vertical entre des paragraphes.

En début de page, cet espace vertical est supprimé :

il faut utiliser la version étoilée `\vspace*{\dim}`

⚠ il ne faut pas utiliser cette commande dans un paragraphe, mais entre des paragraphes.

LATeX définit 3 commandes d'espacement vertical :

`\smallskip` est un espace vertical qui vaut environ un quart de `\baselineskip`.

`\medskip` est un espace vertical qui vaut environ la moitié de `\baselineskip`.

`\bigskip` est un espace vertical qui vaut environ `\baselineskip`.

La longueur `\baselineskip` n'est pas absolue, elle dépend de certains choix effectués dans le document, en particulier de la hauteur des caractères.

.....

Les compteurs

Le langage de **TeX** contient la notion de compteurs : pages, formules, items de listes...

Les compteurs ne sont pas des commandes.

Les compteurs par défaut définis par **LATeX** sont les suivants :

<code>part</code>	<code>paragraph</code>	<code>figure</code>	<code>enumi</code>
<code>chapter</code>	<code>subparagraph</code>	<code>table</code>	<code>enumii</code>
<code>section</code>	<code>page</code>	<code>footnote</code>	<code>enumiii</code>
<code>subsection</code>	<code>equation</code>	<code>mpfootnote</code>	<code>enumiv</code>
<code>subsubsection</code>	<code>tocdepth</code>	<code>secnumdepth</code>	

Les noms sont assez évidents pour la plupart.

`tocdepth` indique la profondeur de la table des matières.

`secnumdepth` indique la profondeur de numérotation des `\part`, ..., `\subparagraph`.

Les compteurs `enumi`... interviennent dans les listes.

D'autres *packages* peuvent fournir des compteurs...

..... Jouer avec les compteurs

LATEX fournit des commandes pour gérer les compteurs.

On suppose que *ctr* est un compteur.

`\setcounter{ctr}{val}` assigne la valeur *val* à *ctr*,
`\addtocounter{ctr}{val}` incrémente *ctr* de la valeur *val*,
`\stepcounter{ctr}` incrémente *ctr* d'une unité.

Le contenu d'un compteur est affichable sous différentes formes :

- `\arabic{-}` affiche des chiffres arabes.
- `\roman{-}` affiche des chiffres romains minuscules.
- `\Roman{-}` affiche des chiffres romains majuscules.
- `\alph{-}` affiche des lettres minuscules.
- `\Alpha{-}` affiche des lettres majuscules.
- `\fnsymbol{-}` utilise la suite de 9 symboles * † ‡ § ¶ || ** †† ‡‡

..... Redéfinir l'affichage d'un compteur

À chaque compteur *ctr* est associée une commande `\thectr` qui “affiche” le compteur :
`\thepage` → 59

Personnaliser l'affichage d'un compteur consiste à redéfinir sa commande `\thectr`.

Cette commande peut afficher plus que la valeur du compteur concerné.

- On peut faire appel à une commande `\thectr` d'un autre compteur.
- On peut formater à volonté l'affichage.

`\renewcommand{\theequation}{\thesection-\Alpha{equation}}`
`\theequation` → 2-D lorsque `section` = 2 et `equation` = 4.
→ On utilise la commande d'affichage du compteur des sections.

`\renewcommand{\thesubsection}{%`
`\{\bfseries\thesection\Alpha{subsection}\}}`
affiche les numéros des sous sections sous la forme 2c.

Le package **amsmath** définit la commande `\numberwithin{-}{-}` qui assujettit le compteur donné dans le premier argument au compteur donné dans le second :

`\numberwithin{subsection}{section}` remet le compteur des sous sections à 0 à chaque début de section et l'affichage est du type 2.3.

Les compteurs doivent exister avant d'invoquer cette commande.

.....
La mise en boite

Principe fondamental de **LAT_EX** : chaque lettre est une boite, chaque mot est un ensemble de boites (les lettres), les paragraphes sont des grosses boites...



Chaque boite est définie par sa largeur (**width**), sa hauteur au dessus de la ligne de base (**height**) et sa profondeur en dessous de la ligne de base (**depth**).

Ainsi, lorsque **LAT_EX** a terminé de composer un tableau, ce dernier devient une boite dont il ne retient que les attributs de dimensions afin de la placer dans la page.

Dans de nombreuses commandes, il est possible d'utiliser les dimensions naturelles des objets **LAT_EX** sur lesquelles elles s'appliquent :

- **\width** désigne la largeur,
- **\height** désigne la hauteur,
- **\depth** désigne la profondeur,
- **\totalheight** désigne la hauteur totale **\height+\depth**.

La commande **** n'imprime pas son contenu tout en créant la boite associée :

ab → a b

Cette commande permet de dépanner dans certaines mises en page difficiles.

.....
La commande \makebox

La commande **\makebox[*largeur*][*position*]{–}** crée une boite de largeur finale *largeur* dont le contenu est positionné selon *position=c,l,r,s*.

Dans les exemples suivants, un cadre est ajouté pour visualiser la boite finale.

a\makebox{phénix}b	a phénix b (block insécable)
a\makebox[5em][r]{phénix}b	a phénix b
a\makebox[1.5em][l]{phénix}bcdef	aphé nix bcdef
a\makebox[5em][c]{phénix}b	a phénix b
a\makebox[8em][s]{A B C D}b	a A B C D b
a\makebox[2\width][l]{phénix}b	aphé nix b
a\makebox[10\height][c]{phénix}b	a phénix b

\framebox est comme **\makebox** avec un cadre en plus.

\fboxrule et **\fboxsep** désignent l'épaisseur du trait et la séparation intérieure.

\framebox{phénix} → phénix (défaut : **\fboxrule=0.4pt, \fboxsep=3.0pt**)

\setlength{\fboxrule}{1.5pt}\setlength{\fboxsep}{1pt}	phénix
--	---

⚠ Les commandes précédentes ne peuvent accepter qu'une boite déjà formatée (quelques mots, un tableau, une image...) mais pas un paragraphe à mettre en forme.

.....
La commande `\parbox`, l'environnement `minipage`

La commande `\parbox[position][hauteur][pos. interne]{largeur}{}{}` crée une boîte dont le contenu est un paragraphe entier qu'il forme sur des lignes de largeur *largeur*.

La hauteur finale est fixée par *hauteur*, le contenu est aligné verticalement selon *pos. interne=t,c,b,s*, le paramètre *position=t,c,b* fixe l'alignement vertical de la boîte finale par rapport au texte ambiant.

`A\parbox{5em}{...}B`

A B

`A\parbox[t][\height][t]{5em}{...}B`

A B

`A\parbox[b][1.5\height][t]{5em}{...}B`

A B

Il existe un environnement équivalent nommé `minipage` :

`\begin{minipage}[c][8ex][b]{5em}`
a a a a a a a a a a
\end{minipage}

A B

.....
La commande `\raisebox`

La commande `\raisebox{distance}[hauteur][profondeur}{}{}` relève son contenu d'une distance *distance* (+ vers le haut, - vers la bas), et produit une boîte finale de hauteur *hauteur* et de profondeur *profondeur*.

Cette commande ne peut contenir qu'une boîte déjà formatée.

`a\raisebox{1ex}{phénix}b`

a b

`a\raisebox{1ex}[1ex][0pt]{phénix}b`

a b

`a\raisebox{-1ex}{phénix}b`

a b

`a\raisebox{-1ex}[2ex][12pt]{phénix}b`

a b

`a\raisebox{-1ex}[\height][\depth]{phénix}b`

a b

`a\raisebox{-1ex}[0pt][0pt]{phénix}b`

a b

En combinant `\makebox` et `\raisebox`, on peut produire une boîte de dimension nulle.

`a\raisebox{1.5ex}[0pt][0pt]{\makebox[0pt][c]{phénix}}b`

phénix

`a\raisebox{1.5ex}[0pt][0pt]{\makebox[0pt][1]{phénix}}b`

ab
phénix

8 – Les listes

..... Les listes de LATEX

LATEX définit trois types de listes : listes à puces, listes numérotées et listes à mots-clés.

```
\begin{itemize}
\item In vino veritas      – In vino veritas
\item Natura abhorret a vacuo – Natura abhorret a vacuo
\end{itemize}
```

```
\begin{enumerate}
\item Qui rogat, non errat
\item Qui scribit, bis legit
\end{enumerate}
```

→ possible de placer des `\label{-}` pour faire référence au numéro de l'item.

```
\begin{description}
\item[Cicéron :] Cuiusvis hominis est errare
\item[Horace :] Qui cupid aut metuit
  liber non erit unquam
\end{description}
```

Cicéron :	Cuiusvis hominis est errare
Horace :	Qui cupid aut metuit liber non erit unquam

..... Emboîtements de listes

On peut emboîter jusqu'à 4 niveaux de listes :

```
\begin{enumerate}
\item UN
  \begin{itemize}
    \item un de UN
      \begin{enumerate}
        \item premier
        \item second
      \end{enumerate}
    \item deux de UN
  \end{itemize}
\item DEUX
\end{enumerate}
```

Chaque (sous)-niveau de liste ajuste ses paramètres : espaces verticaux, retraits...

.....
Les listes de `pifont`

Le package **pifont** définit des listes à partir de symboles de la police **ZAPF DINGBATS** :

```
\begin{dinglist}{43} ... \end{dinglist}
```

- ☞ le premier item de ma liste
- ☞ le second item de ma liste

```
\begin{dinglist}{253} ... \end{dinglist}
```

- ☞ le premier item de ma liste
- ☞ le second item de ma liste

```
\begin{dingauto}{192} ... \end{dingauto}
```

- ① le premier item de ma liste
- ② le second item de ma liste

```
\begin{dingauto}{202} ... \end{dingauto}
```

- ❶ le premier item de ma liste
- ❷ le second item de ma liste

Voir la documentation de **pifont** pour plus d'informations.

.....
Le package `enumitem`

Le package **enumitem** permet de modifier les paramètres des 3 listes usuelles **itemize**, **enumerate** et **description**, et il offre de définir de nouvelles listes.

```
\begin{enumerate}[label=\emph{\alph*}), parsep=2ex]
  \item ...
  \item ...
\end{enumerate}
```

Il est possible de changer globalement l'aspect d'une liste :

```
\setlist[itemize,1]{label=$\clubsuit$}
\setlist[itemize,2]{label=$\diamondsuit$}
```

Il est possible de définir de nouvelles listes :

```
\newlist{maliste}{enumerate}{1}
\setlist[maliste]{label=\alph*), font=\sffamily\bfseries}
```

Les dimensions définissant la géométrie d'une liste sont modifiables.

On peut personnaliser la police avec **font=...** et le label avec **label=....**

Voir la documentation de **enumitem** pour plus d'informations.

9 – Les tableaux

La syntaxe des tableaux

L'environnement `tabular` permet de composer des tableaux.
On indique le type des colonnes dans l'argument obligatoire.
Les cellules sont séparées par un `&`. Chaque ligne est terminée par un `\\"`.

Syntaxe de déclaration des colonnes

<code> </code>	insère une ligne verticale
<code>l</code>	alignement à gauche
<code>r</code>	alignement à droite
<code>c</code>	alignement au centre
<code>p{<dim>}</code>	paragraphe de largeur donnée
<code>@{code}</code>	supprime l'espace entre colonnes et insère <code>code</code>
<code>*{num}{opts}</code>	répète <code>num</code> fois la déclaration <code>opts</code>

`\multicolumn{num}{col}{contenu}` regroupe `num` cellules consécutives en une seule cellule de spécification `col` et y place `contenu`.

Les paramètres suivants déterminent l'aspect du tableau :

<code>\tabcolsep</code>	moitié de la largeur de l'espacement entre les colonnes	(6pt)
<code>\arrayrulewidth</code>	épaisseur des traits du tableau	(0.4pt)
<code>\doublerulesep</code>	séparation entre les doubles traits ()	(2pt)
<code>\arraystretch</code>	fraction avec laquelle l'espace entre lignes est multiplié	(1.0)

Les tableaux : exemples

```
A\begin{tabular}[b]{||rc|1|}\hline
ABC & DEF & GHI \\ \hline
J & K & L \\ \hline\hline
\multicolumn{2}{||c|}{MNO} & PQRS \\ \hline
\end{tabular}B
```

ABC	DEF	GHI
J	K	L
A	MNO	PQRS

```
\renewcommand{\arraystretch}{1.5}
a \begin{tabular}[c]{r@{=}l}
d & 2 \\
c & 3
\end{tabular} b
```

a $d=2$ b
c=3

L'argument optionnel de `tabular` gère l'alignement vertical du tableau : `t`, `c`, `b`

Le package **array**

Le package **array** définit d'autres commandes de déclarations de colonnes :

<code>m{<dim>}</code>	colonne de largeur donnée, verticalement centrée
<code>b{<dim>}</code>	colonne de largeur donnée, verticalement en bas
<code>>{code}</code>	insère <code>code</code> avant les données d'une cellule
<code><{code}</code>	insère <code>code</code> après les données d'une cellule
<code>!{code}</code>	insère <code>code</code> entre deux colonnes

La commande `\newcolumntype{-}{-}` permet de définir des nouveaux types de colonnes en utilisant la syntaxe des déclarations de colonnes.

La dimension `\extrarowheight` s'ajoute à la hauteur des cellules.

```
\setlength{\extrarowheight}{4pt}
\begin{tabular}{|>{\large}c|>{\bfseries}|}
\hline A & B \\
\hline 100 & 50 \\
\end{tabular}
```

A	B
100	50

```
\newcolumntype{R}{>{$}r<{$}}
\begin{tabular}{|R|=|>{\bfseries}|}
\hline a_1 & A \\
\hline b_1 & B \\
\end{tabular}
```

$a_1 = A$
$b_1 = B$

Le package **array** : autres exemples

```
\begin{tabular}{p{2em}p{2em}}
\hline a a a a a a & b \\
\end{tabular}
```

a a a	b
a a a	

```
\begin{tabular}{m{2em}m{2em}}
\hline a a a a a a & b \\
\end{tabular}
```

a a a	b
a a a	

```
\begin{tabular}{b{2em}b{2em}}
\hline a a a a a a & b \\
\end{tabular}
```

a a a	
a a a	b

```
\newcolumntype{z}{>{\raggedleft\arraybackslash}p{2em}}
\begin{tabular}{zz}
\hline a a a a a a & b b b b b b \\
\end{tabular}
```

a a a a	b b b b
a a a	b b b

Les commandes `\raggedright` et `\centering` peuvent aussi être utilisées.

⚠ Ces commandes redéfinissent `\`, qu'il faut rétablir aussitôt avec `\arraybackslash`.

10 – Les couleurs

Le package **xcolor**

Le package **xcolor** permet de gérer les couleurs dans **LATEX**.

Une couleur est définie à partir d'un **modèle** et d'une **spécification** :

description	modèle	spécification
<i>red-green-blue</i>	rgb	3 nombres compris entre 0 et 1
<i>Red-Green-Blue</i>	RGB	3 nombres entiers compris entre 0 et 255
Échelle de gris	gray	1 nombre compris entre 0 et 1
Niveau de gris	Gray	1 nombre entier entre 0 et 15
Nom explicite	named	nom d'une couleur prédéfinie
<i>cyan-magenta-yellow-black</i>	cmyk	4 nombres compris entre 0 et 1
<i>cyan-magenta-yellow</i>	cmy	3 nombres compris entre 0 et 1
<i>hue-saturation-brightness</i>	hsb	3 nombres compris entre 0 et 1
Longueur d'onde en nm	wave	1 nombre compris entre 363 et 814
<i>RRGGBB</i>	HTML	1 nombre hexadécimal entre 000000 et FFFFFF
<i>Hue-Saturation-Brightness</i>	HSB	3 nombres entiers entre 0 et 240

Définition d'une couleur

La commande

\definecolor{nom}{modèle}{spécification}

définit une couleur nommée *nom* dans le modèle et les spécifications choisis.

```
\definecolor{macouleur}{rgb}{rgb}{0.5, 0.2, 0.8} →   
\definecolor{macouleur}{RGB}{100,150,200} →   
\definecolor{macouleur}{gray}{0.8} →   
\definecolor{macouleur}{Gray}{10} →   
\definecolor{macouleur}{named}{magenta} →   
\definecolor{macouleur}{cmyk}{0.5,0.2,0.8,0.3} →   
\definecolor{macouleur}{cmy}{0.5,0.2,0.8} →   
\definecolor{macouleur}{hsb}{0.5, 0.2, 0.8} →   
\definecolor{macouleur}{wave}{650.57} →   
\definecolor{macouleur}{HTML}{43ADF9} →   
\definecolor{macouleur}{HSB}{10,100,200} → 
```

On peut redéfinir une couleur déjà définie à l'aide de cette commande.

Utilisation des couleurs

Plusieurs commandes sont définies pour utiliser les couleurs.

- `\color{couleur}` installe la couleur dont le nom est dans l'argument.
Noir `\color{magenta} couleur` et noir → Noir couleur et noir
- `\textcolor{couleur}{texte}` imprime le texte dans la couleur sélectionnée.
Noir `\textcolor{magenta}{couleur}` et noir → Noir couleur et noir
- `\colorbox{couleur}{texte}`, `\fcolorbox{couleur1}{couleur2}{texte}` produisent des boîtes de fond coloré.

`\colorbox{magenta}{test}` → 

`\fcolorbox{blue}{magenta}{test}` → 

Les longueurs `\fboxrule` et `\fboxsep` gèrent l'épaisseur du trait et de la séparation avec le texte.

- `\pagecolor{couleur}` colore le fond de la page.
- Ces commandes acceptent une variante qui spécifie la couleur par un modèle :

`\color[modèle]{spécification}`

`\textcolor[modèle]{spécification}{texte}`

`\colorbox[modèle]{spécification}{texte}`

`\fcolorbox[modèle]{spécification1}{spécification2}{texte}`

`\pagecolor[modèle]{spécification}`

Les couleurs prédéfinies et les mélanges

Le package `xcolor` définit les couleurs suivantes :

black		white		red		green		blue	
cyan		magenta		yellow		orange		pink	
brown		olive		purple		teal		violet	
gray		darkgray		lightgray					

Les options `dvipsnames`, `svgnames` et `x11names` de `xcolor` chargent de nombreuses autres couleurs prédéfinies.

Consulter la documentation de `xcolor`⁵⁷.

La commande `\colorlet{couleur}{mélange}` définit une couleur à partir de mélanges :

`blue!85!black` → , `blue!65!black` → , `blue!45!black` → 

`-red` →  (couleur complémentaire)

`red!50!green!60!yellow` → , `red!50!green!20!yellow` → 

`red!50!green!20` →  (dernière couleur : white)

Les commandes `\color`, `\textcolor`, `\colorbox`, `\fcolorbox` et `\pagecolor` acceptent directement des couleurs sous forme de mélanges :

`\textcolor{blue!85!black}{test}` → 

11 – Les images et les graphiques

L'inclusion d'images : le package **graphicx**

pdflatex permet d'insérer des images aux formats **.pdf**, **.jpg** ou **.png**

Le package **epstopdf** permet de convertir à la volée des fichiers **.eps** en **.pdf**.

\includegraphics[**clés**]{**nom de l'image**}, du package **graphicx**, insère une image.

L'argument obligatoire est le nom d'un fichier d'image (avec son chemin), l'argument optionnel définit une liste d'options, de la forme **clé=valeur** :

\includegraphics[**width=2cm**]{image.png}

\includegraphics[**angle=45,origin=c**]{dossier/image.pdf}

Les clés principales sont les suivantes :

Clé	Description	Exemple
width	largeur de l'image	width=1cm
height	hauteur de l'image	height=50pt
scale	mise à l'échelle de l'image	scale=2
angle	angle de rotation (degrés)	angle=45
origin	origine de la rotation (lrcbtB)	origin=br
viewport	définit la zone à afficher (bp par défaut)	viewport=0 0 72 72
trim	définit les marges à enlever	trim=10 15 5 7
clip	définit la zone à afficher	clip=true

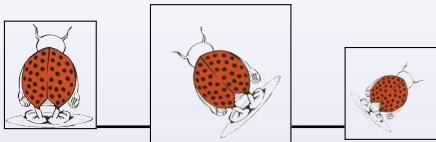
D'autres clés sont décrites dans la documentation du package **graphicx**.

La commande \includegraphics[**clés**]{**nom de l'image**} : exemples

\includegraphics[width=1.7cm]{pomme.png}



\includegraphics[width=1.2cm]{dos.png}
\includegraphics[width=1.2cm,origin=c,angle=45]{dos.png}
\includegraphics[origin=c,angle=-45,width=1.2cm]{dos.png}



(cadres et lignes ajoutés)

→ L'ordre des options est important.

→ Une figure sans rotation n'a pas de profondeur, une figure tournée peut en avoir une.

.
Rotations d'objets

`\rotatebox[-]{angle}{texte}` (**graphicx**) tourne *texte* de l'angle *angle*.

angle est donné en degrés, dans le sens trigonométrique.

texte peut être n'importe quel objet **LAT_EX** : texte, tableau, image...

L'argument optionnel définit des options sous la forme d'une liste de **c_lé=valeu**r :

Clé	Description	Exemple
origin	origine de la rotation (l _r c _t bB)	<code>origin=br</code>
x	abscisse du centre de rotation	<code>x=50pt</code>
y	ordonnée du centre de rotation	<code>y=5mm</code>

l = *left*, **r** = *right*, **c** = *center*, **t** = *top*, **b** = *bottom*, **B** = *Baseline*.

ABCD
`\rotatebox[origin=c]{45}{ABCD}`
`\rotatebox[origin=tr]{90}{ABCD}`
`\rotatebox[x=0pt,y=15pt]{60}{ABCD}`

.
Mises à l'échelle par un facteur

`\scalebox{H}[V]{texte}` (**graphicx**) met à l'échelle *texte* :

facteur *H* dans la direction horizontale,

facteur (optionnel) *V* dans la direction verticale.

Si le facteur *V* n'est pas spécifié, la mise à l'échelle se fait avec *H* dans les deux directions.

texte peut être n'importe quel objet **LAT_EX**.

ABCD ABCD
`\scalebox{2}{0.5}{ABCD}`
`\scalebox{2}{ABCD}`
`\scalebox{1}{-2}{ABCD}`
`\scalebox{-2}{ABCD}`

`\reflectbox{texte}` équivaut à `\scalebox{-1}{1}{texte}`.

ABCD ABCD
`\reflectbox{ABCD}`
`ABCD`

.....
Mises à l'échelle vers des dimensions données

`\resizebox{\dim-h}{\dim-v}{texte}` (**graphicx**) met à l'échelle *texte* pour atteindre une boîte à la dimension horizontale `\dim-h` et à la dimension verticale `\dim-v`.

Si `\dim-h` ou `\dim-v` vaut `!`, l'autre dimension impose la mise à l'échelle en préservant les proportions de *texte*.

texte peut être n'importe quel objet **LAT_EX**.

On peut utiliser `\height`, `\width`, `\totalheight`, `\depth` dans ces arguments pour faire référence aux dimensions naturelles de *texte*.

	ABCD	ABCD
<code>\resizebox{1cm}{0.5cm}{ABCD}</code>	ABCD	ABCD
<code>\resizebox{2.5cm}{!}{ABCD}</code>	ABCD	ABCD
<code>\resizebox{!}{15pt}{ABCD}</code>	ABCD	ABCD
<code>\resizebox{\width}{15pt}{ABCD}</code>	ABCD	ABCD
<code>\resizebox{\width}{3\height}{ABCD}</code>	ABCD	ABCD

12 – Les flottants

Le concept des flottants

Un *flottant* est un objet dont le placement dans le document est déterminé par **LAT_EX**.

→ images, tableaux, courbes, dessins, encarts... avec légendes.

L'objet n'est pas placé dans le flux de texte courant contrairement aux autres objets usuels.

Les flottants sont conservés en mémoire jusqu'à ce que **LAT_EX** rencontre un endroit adéquat pour les placer (dans l'ordre d'arrivée).

→ Parfois cet endroit ne se rencontre qu'à la fin du document.

→ Un flottant peut bloquer le placement des flottants suivants.

Par défaut, **LAT_EX** définit deux environnement flottants : **figure** et **table**.

figure s'utilise pour les graphiques, et **table** s'utilise pour les tableaux.

⚠ Pour insérer un graphique ou un tableau, on n'a pas besoin de ces environnements !

Ces deux environnements ont pour fonctions :

1. de placer leur contenu à un endroit du document où il y a suffisamment de place ;
2. de placer une légende numérotée renseignée par la commande **\caption[-]{-}**, de type **Figure** ou **Tableau** ;
3. de conserver les données (légende, page, numéro) dans un fichier auxiliaire pour construire une table des figures ou une table des tableaux.

L'environnement **figure**

Exemple d'utilisation de l'environnement **figure** :

```
\begin{figure}[ht]
  \centering
  \includegraphics[width=0.5\textwidth]{image.png}
  \caption[Légende courte]{Une légende pour décrire la figure}
  \label{fig-image}
\end{figure}
```

Placer **\label{-}** après **\caption{-}** pour récupérer le numéro.

L'option de **figure** impose le comportement pour placer le flottant dans la page :

t (haut de page), **b** (bas de page), **p** (page de flottants), **h** (ici si possible).

Ces options peuvent être combinées dans l'ordre de préférence : **[htp]**.

L'option **p** place le flottant sur une page réservée aux flottants.

Le package **float** ajoute l'option **H** qui place le flottant exactement à l'endroit défini.

⚠ N'utiliser **figure** que si on a besoin d'une légende et/ou d'un numéro...

Ceci s'applique aussi à l'environnement **table**.

\listoffigures et **\listoftables** composent une liste des figures et des tableaux.

13 – La bibliographie

L'environnement `thebibliography`

L'environnement `thebibliography` sert à créer une liste de références bibliographiques. Il débute une section non numérotée (`article`) ou un chapitre non numéroté (`book`).

```
\begin{thebibliography}{AAAA}
\bibitem{Arve98a} W.~Arveson.
  \newblock {\em An Invitation to  $C^{\ast}$ -Algebras},
  volume~39 of {\em Graduate Texts in Mathematics}.
  \newblock Springer-Verlag, 1998.

\bibitem{Bert96a} R.~Bertlmann.
  \newblock {\em Anomalies in Quantum Field Theory}.
  \newblock Oxford Science Publications, 1996.

\bibitem[Wei67]{Wein67a} Stephen Weinberg.
  \newblock A model of Leptons.
  \newblock {\em Phys. Rev. Lett.}, 19(21):1264–1266, Nov 1967.
...
\end{thebibliography}
```

L'argument obligatoire de l'environnement `thebibliography` sert à fixer la largeur maximale des libellés apparents (seul le nombre de lettres compte!).

Syntaxes de `\bibitem[-]{-}` et `\cite[-]{-}`

Chaque item de la liste est nommé `\bibitem[-]{-}`.

- L'argument optionnel est la clé apparente dans le document.
Sans cette option, chaque `\bibitem` produit un numéro.
On peut mélanger des clés manuelles et des clés automatiques.
- L'argument obligatoire de `\bibitem` est la clé interne de citation.
Argument des commandes de citation dans le texte `\cite{-}`.

Les commandes `\newblock` (option) permettent de séparer les entités sémantiques.
Selon la classe du document, elle peut ajouter un espace ou passer à la ligne...

La commande `\cite[-]{-}` permet de citer une entrée bibliographique.

- L'argument obligatoire est la clé interne de citation.
- L'argument optionnel permet d'ajouter du texte.

Exemples :

- `\cite{Arve98a}` → [1]
- `\cite{Wein67a,Bert96a}` → [Wei67, 2]
- `\cite[p.~100]{Bert96a}` → [Ber96, p. 100]
- `\cite[Fig.~2.40]{Bert96}` → [Ber96, Fig. 2.40]
- `\cite[Thm~4]{Arve98a}` → [1, Thm 4]

14 – La chimie

Le package **chemmacros**

chemmacros est un ensemble de commandes destinées aux chimistes :

- noms de symboles chimiques divers et composés chimiques ;
- compositions de noms de molécules selon le *Globally Harmonized System* ;
- écriture de réactions chimiques ;
- raccourcis usuels concernant les problèmes d'acido-basité et d'oxydo-réduction ;
- variables thermodynamiques ;
- unités usuelles en chimie (package **siunitx**) ;
- pictogrammes.

Commandes diverses :

```
\Hp1, \Ht0, \prt, \Nu, \pH, Ca\pch[2], S\mch[2],  
\ch{AgCl2-}, \ch{^227_{90}Th+}, \ch{2H2O},  
\ch{2 H2O}, \ch{SO4^2-}, \ch{Ca^2+ SO4^2-}  
H+, H3O+, p+, Nu-, pH, Ca2+, S2-, AgCl2-, 22790Th+, 2H2O, 2 H2O, SO42-, Ca2+SO42-
```

Le package **chemmacros** (suite)

Noms de molécules :

```
\iupac{\Z-2-Butene}  
(Z)-2-Butene
```

```
\iupac{Tetra\cyclo[2.2.2.1^1,4]\-un\|decane-2\‐dodecy]%-  
\‐5\‐(hepta\|decyl\|iso\|dodecy]\|thio\|ester)}  
Tetracyclo[2.2.2.11,4]-undecane-2-dodecyl-5-(heptadecylisododecylthioester)
```

Réactions chimiques :

```
\begin{reaction}  
4 Fe + 3 O2 -> 2 Fe2O3  
4 Fe + 3 O2 → 2 Fe2O3 {1}  
\end{reaction}
```

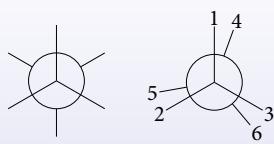
```
\ch{A + B ->[a] C}  
A + B a→ C
```

```
\ch{C\solid{} + 2 H2O\liquid{} -> CO2\gas{} + 2 H2\gas{}}  
C(s) + 2 H2O(l) → CO2(g) + 2 H2(g)
```

Le package `chemmacros` (suite)

Stéréochimie :

```
\newman{} \quad \newman(160){1,2,3,4,5,6}
```



Orbitales atomiques :

```
\orbital{s} \quad \orbital{p} \quad \orbital{sp}
```



Pictogrammes :

```
\ghspic{flame} \ghspic{skull}
```

Le package `chemfig`

`chemfig` est destiné à dessiner des molécules chimiques développées.

Il est basé sur le *package* de dessin `TikZ`.

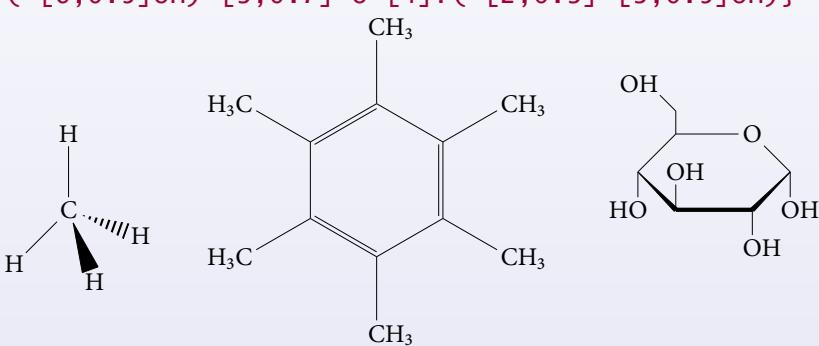
```
\chemfig{C(-[5]H)(-[2]H)(<[:-70]H)(<[:-20]H)}
\definesubmol[Me[H_3C]]{CH_3}
\chemfig{*6((-!\Me)=(-!\Me)-(-!\Me)=(-!\Me)-(-!\Me)=(-!\Me)-)}
```

```
\setcrambond{2pt}{}{}
```

```
\chemfig{HO-[2,0.5,2]?<[7,0.7](-[2,0.5]OH)-%
```

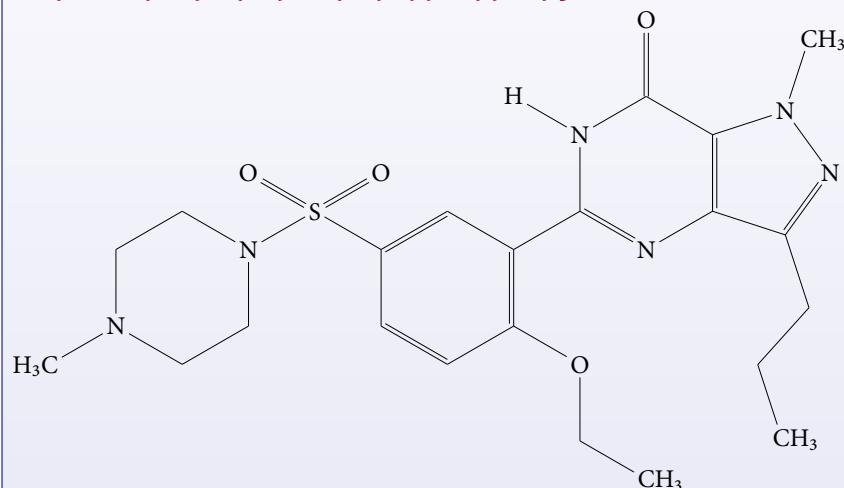
```
[,,,line width=2pt](-[6,0.5]OH)>[1,0.7]%
```

```
(-[6,0.5]OH)-[3,0.7] 0-[4]?(-[2,0.3]-[3,0.5]OH)}
```



Le package `chemfig` (suite)

```
% formule du viagra
\chemfig{N*6((-H_3C)--N(-S(=[:::+120]O)(=[:::+0]O)%
-[:::-60]*6(--(-O-[:::-60]-[:::+60]CH_3)%
=(*-6(=N-*5(-(-[:::-60]-[:::+60]CH_3)=N%
-N(-CH_3)--=(=O)-N(-H)-))-=-))-)}
```

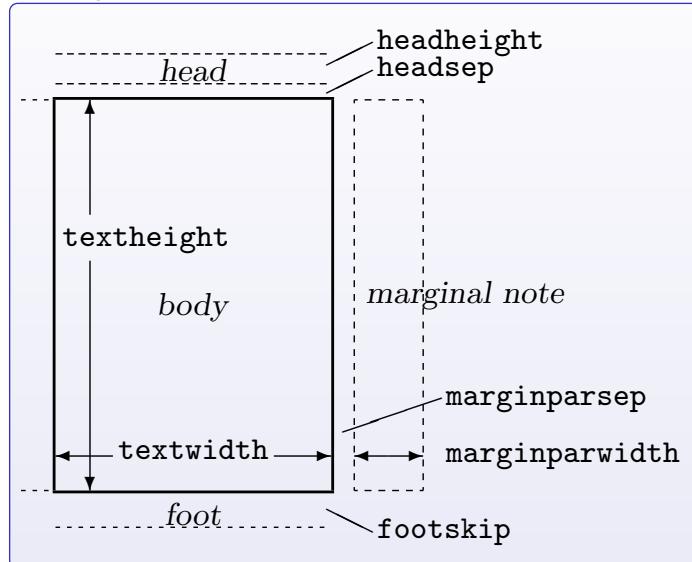


Lire la documentation pour plus de détails.

15 – Le style du document

Les éléments d'une page et d'un document

Une page composée par **LAT_EX** est constituée de différents éléments :



- l'entête, où peuvent figurer les titres de chapitres et de sections ;
- le pied de page, où figurent souvent le numéro de la page ;
- des notes de bas de page (**\footnote{-}**) ;
- des notes dans la marge (**\marginpar{-}**) ;
- et le corps du texte qui contient à son tour des paragraphes, divisions par chapitres et sections, *etc.*

Le style des pages

La commande **\pagestyle{-}** installe un style qui va gérer les entêtes et pieds de page.

On peut changer le style plusieurs fois dans le document.

La commande **\thispagestyle{-}** n'affecte que la page en cours.

Par défaut, 4 styles sont prédéfinis :

empty L'entête et le pied de page sont vides.

plain L'entête est vide, le pied de page contient le numéro de la page centré.

headings L'entête contient les chapitres et sections (ou sous divisions) ainsi que le numéro de la page, le pied de page est vide.

myheadings L'entête contient le numéro de la page et des informations fournies par l'utilisateur, le pied de page est vide.

Dans la classe **book**, un chapitre commence dans le style **plain**, les autres pages sont dans le style **headings** par défaut.

.....
Le package `fancyhdr`

Le package **fancyhdr** définit un nouveau style de page appelé **fancy**.

Ce style contient 6 parties : gauche, centre, droite pour l'entête et le pied de page.

On peut personnaliser le style à l'aide des 2 commandes :

`\fancyhead[-]{-}` et `\fancyfoot[-]{-}`.

Argument optionnel : **L**, **C**, **R** pour l'une des trois zones,
0, **E** pour les pages impaires ou paires.

```
\pagestyle{fancy}           % on installe le style
\fancyhead{} \fancyfoot{} % on vide le style
\fancyhead[L0]{\sshape\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\fancyfoot[LO,RE]{\thepage}\fancyfoot[RO,LE]{\today}
\renewcommand{\headrule}{\dotfill}
\renewcommand{\footrulewidth}{0pt}
```

`\headrule` et `\footrule` : lignes sous l'entête et au dessus du pied

La commande `\fancypagestyle{-}{-}` permet de redéfinir un style déjà défini.

Lire la documentation pour plus d'informations.

.....
Le package `titlesec` : le style des pages

Le package **titlesec** permet de (re)définir des styles de page.

La commande `\newpagestyle{nom} [options globales] {code}` et son équivalent

`\renewpagestyle` (re)définissent un style nommé *nom* selon le code donné par *code*.

Les *options globales* s'appliquent aux entêtes et aux pieds.

Le code *code* repose sur deux commandes `\sethead[-][-][-]{-}{-}`
`\setfoot[-][-][-]{-}{-}`.

Ces lots de trois cases sont les parties gauches, centrées, droites des entêtes et des pieds.

Les parties optionnelles concernent les pages paires.

Des commandes permettent d'insérer dans ces arguments des données diverses :

- `\headrule`, `\footrule`, et variantes... pour des filets ;
- `\chapertitle`, `\sectiontitle`... pour les titres de chapitres...
- `\ifthechapter{-}{-}`, `\ifthesection{-}{-}`...

.....

Le package `titlesec` : exemples de styles

Rédefinition du style `plain` :

```
\renewpagestyle{plain}
  [\normalfont\sffamily\bfseries\mathversion{bold}]
  {\setfoot[\today]{}{}{\today}}
  {\sethead[\usepage]{}{}{\usepage}}
```

Définition d'un nouveau style :

```
\newpagestyle{main}
  [\normalfont\sffamily\bfseries\mathversion{bold}]
  {\headrule
  \setfoot[\today]{}{}{\today}
  \sethead[\usepage]{}{}{\ifthechapter
    {\chapertitlename\ \thechapter\ - \chapertitle}
    {\chapertitle}}
  \ifthesection{\thesection\ - \sectiontitle}
    {\chapertitle}%
  {}{\usepage}}
```

.....

Le package `titlesec` : les commandes de section

`titlesec` peut personnaliser l'apparence des commandes de type sections.

- `\titleformat{-}{-}{-}{-}{-}` définit le rendu d'une commande du type `\chapter`, `\section`...
- `\titlespacing*{-}{-}{-}{-}` définit les espacements avant et après les commandes `\chapter`, `\section`...

Les arguments de `\titleformat` désignent des points précis : forme générale du label, formatage général, formatage du label seul, formatage du titre seul, espaces horizontaux...

Les arguments de `\titlespacing*` concernent les espacements horizontaux à gauche et à droite, et les espacements verticaux avant et après.

→ Consulter la documentation pour des explications plus fournies (avec exemples).

.....
Le package `titlesec` : exemples

Redéfinition du style des chapitres :

```
\titleformat{\chapter}[display]
  {\raggedright\normalfont\sffamily\LARGE\bfseries%
   \mathversion{bold}}
  {\chapertitlename\ \thechapter}{0pt}{\huge}

\titlespacing*{\chapter}{0pt}{20pt}{4ex plus 1ex minus 1ex}
\assignpagestyle{\chapter}{empty}
\newcommand{\chapterbreak}{\cleardoublepage}
```

Redéfinition du style des sections :

```
\titleformat{\section}
  {\normalfont\sffamily\large\bfseries\mathversion{bold}}
  {\thesection}{1em}{}

\titlespacing*{\section}{0pt}{3ex plus 2.5ex minus .8ex}
  {2ex plus .5ex}
```

16 – La classe beamer

La classe beamer : les bases

La classe **beamer**⁵⁸ est destinée à créer une présentation à l'ordinateur à partir d'un document écrit en **LATEX**.

Les principes généraux de **beamer** sont les suivants :

- Le document produit est du **PDF**, qui est projeté avec n'importe quel logiciel⁵⁹ capable d'afficher un **PDF** en plein écran.
- **beamer** ajoute des commandes spécifiques à **LATEX**.
- Notion de thèmes, modifiables à volonté avec le langage **LATEX**.
- Une présentation à l'ordinateur se fait en mode “paysage”.
- **beamer** bascule sur des fontes sans empattements, y compris pour les mathématiques.
- Le texte est collé à gauche et non pas justifié.
- **beamer** produit automatiquement une table des matières et fournit une barre de navigation avec des hyperliens internes.
- **beamer** ne peut pas produire des effets de transitions riches.
- Avec **Adobe Reader**⁶⁰, on peut utiliser des éléments **FLASH**⁶¹ (animations, vidéos).

La structure d'une présentation

Une présentation sous **beamer** peut être découpée globalement avec des commandes **\part**, **\section** et **\subsection** (**\subsubsection** existe mais l'éviter).

→ **\part** n'est utile que dans des séries d'exposés (cours par exemple).

L'unité élémentaire d'une présentation est la **diapositive**, définie par la commande **\frame[-]{-}** ou l'environnement **\begin{frame}{-}{-} ... \end{frame}**.

→ Dans la suite on ne considère que l'environnement **frame**.

Les éléments suivants peuvent être générés automatiquement :

Une diapositive de titre : il faut renseigner le titre de l'exposé, l'auteur, l'institut, la date...

Des tables des matières : il est possible de créer des diapositives contenant des tables de matières de différents niveaux et aspects.

Des diapositives “section” : elles peuvent marquer l'entrée dans une section.

Une annexe : une annexe peut être créée pour “cacher” des diapositives placées au delà de la fin de l'exposé (réserve pour les questions).

→ Absente des tables des matières, non comptabilisée dans le total des diapositives.

Concept de “pauses” : chaque diapositive peut être subdivisée pour faire apparaître du texte étape par étape (1 étape = 1 page **PDF** = 1 *slide*).

Le compteur **beamerpauses** indexe ces étapes : il augmente d'une unité par étape définie.

.....
Un code source type

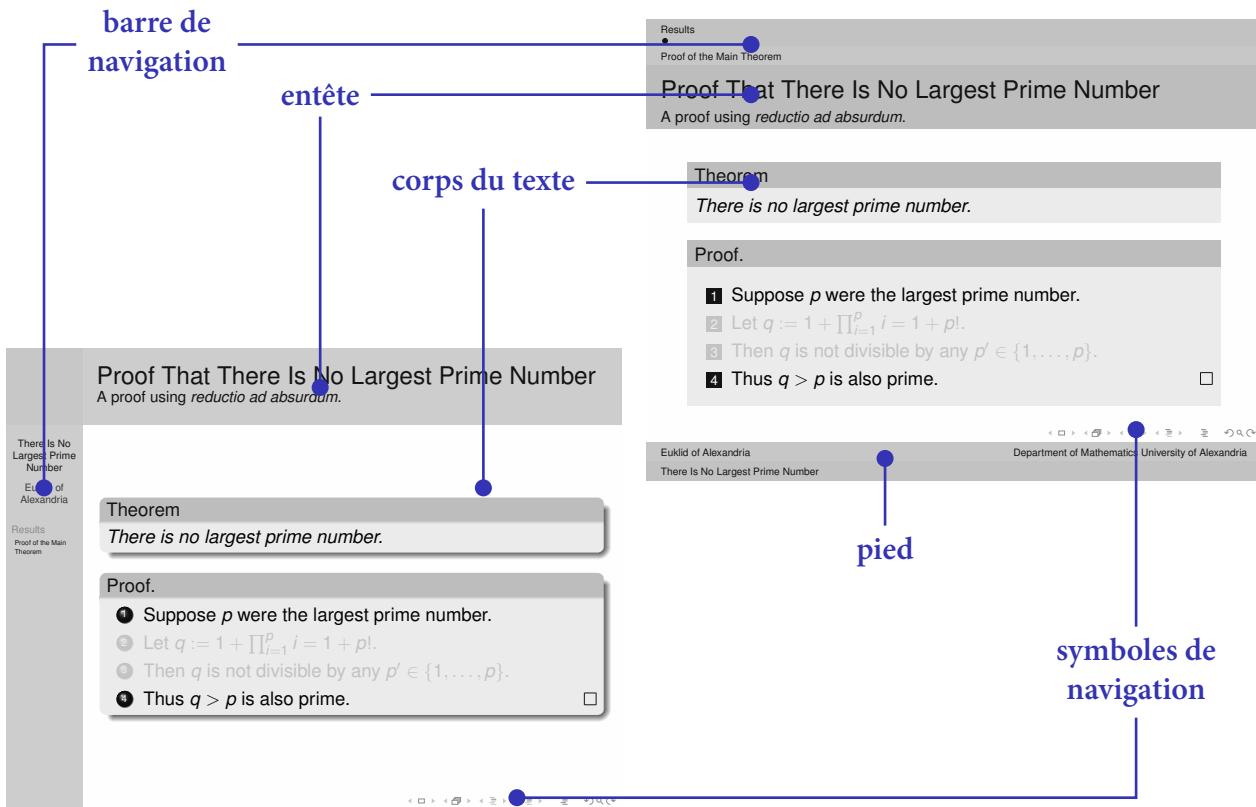
```
\documentclass[10pt]{beamer}
\usepackage{etex}
\usepackage[utf8]{inputenc}
\usepackage[TS1,T1]{fontenc}
\usepackage[english,french]{babel}
\usepackage{lmodern}

\usestheme{Goettingen}
\title[Beamer en action]{Beamer en action}
\author[T. Masson]{Thierry Masson}
\institute[CPT]{Centre de Physique Théorique}
\date[2 avril 2012]{2 avril 2012}

\begin{document}
\section{Une première section}
\begin{frame}{Une première diapositive}
    Bonjour !
\end{frame}
\end{document}
```

beamer charge les *packages* suivants :

geometry
graphicx
xcolor
pgf
hyperref
amssymb
amsmath
amsfonts
amsthm
translator
enumerate

.....
La structure d'une diapositive en images

..... La structure d'une diapositive en théorie

Une diapositive est constituée de différents éléments “physiques” :

La barre de navigation : cette barre contient en général le rappel de la structure logique de la présentation : sections, sous sections, diapositives.

Cette barre peut être horizontale ou verticale selon le thème choisi.

L'en-tête : c'est l'endroit usuel pour placer le titre (et le sous titre) de la diapositive.

Le corps du texte : c'est le contenant de ce qui est présenté dans la diapositive.

Le pied : on peut y placer le titre de la conférence, le nom du présentateur et son institut, la date, le numéro de la diapositive...

Les symboles de navigation : ce sont des symboles cliquables (discrets) permettant la navigation dans la présentation : début, section, table des matières...

Ces éléments ont une apparence et une position déterminées par le thème choisi.

Ils peuvent être personnalisés grâce à des modèles (*templates*) facile à modifier.

De nombreux textes (titre, sections, sous sections...) et symboles insérés dans ces éléments sont des hyperliens internes au document.

→ Accès facile et permanent à diverses parties du document.

..... L'environnement frame

Plusieurs syntaxes sont possibles pour l'environnement **frame** :

```
\begin{frame}{Titre}
...
\end{frame}
```

```
\begin{frame}{Titre}{Sous Titre}
...
\end{frame}
```

```
\begin{frame}
\frametitle{Titre}
\framesubtitle{Sous Titre}
...
\end{frame}
```

Les commandes **\frametitle** et **\framesubtitle** sont optionnelles.

L'option **[plain]** ôte les barres de navigation et le pied de la diapositive.

```
\begin{frame}[plain]{Titre}
...
\end{frame}
```

```
\begin{frame}[plain]
...
\end{frame}
```

→ Gain d'espace pour le contenu de la diapositive (présentation d'un graphique par ex.).

Par défaut, le texte est centré verticalement dans une diapositive.

Les options **t**, **b** et **c** (défaut) de **frame** peuvent modifier ce comportement.

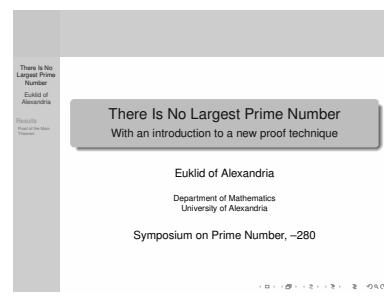
La diapositive de titre

La diapositive de titre peut-être créée par :

```
\begin{frame}[plain]
  \titlepage
\end{frame}
```

On peut écrire plus simplement :

```
\maketitle
```



Les champs à renseigner pour créer la diapositive de titre sont :

`\title[-]{-}` définit le titre de la présentation.

`\subtitle[-]{-}` définit le sous titre de la présentation.

`\author[-]{-}` désigne la liste des auteurs séparés par `\and`.

`\institute[-]{-}` désigne la liste des instituts séparés par `\and`.

`\date[-]{-}` définit la date, et on ajoute souvent le lieu de la conférence.

`\titlegraphic{-}` désigne un graphique de titre.

Les arguments optionnels dans les commandes ci-dessus sont utilisés pour le rappel des données sur chaque diapositive. ➔ Fournir une version plus courte.

La composition de la diapositive de titre dépend du thème choisi.

Il existe des commandes `\partpage`, `\sectionpage` et `\subsectionpage`.

Les tables des matières

Les tables des matières sont composées des titres des sections et des sous-sections.

`\tableofcontents [options]` avec les options :

`sectionstyle=` détermine le style de la section courante et des autres sections :

`sect. courante/autres sect.` avec les valeurs `show`, `shaded`, `hide`.

Si le second est omis, il prend la valeur du premier.

`subsectionstyle=` détermine de même l'aspect des sous sections :

`ss-sect. courante/autres ss-sect. dans la sect. courante/autres ss-sect..`

Si le troisième est omis, il prend la valeur du second.

Si le second est omis aussi, les trois prennent la valeur du premier.

`currentsection` équivaut à

`sectionstyle=show/shaded, subsectionstyle=show/show/shaded.`

`currentsubsection` équivaut à `subsectionstyle=show/shaded`.

`hideallsubsections` équivaut à `subsectionstyle=hide`.

`hideothersubsections` équivaut à `subsectionstyle=show/show/hide`.

Avec ces options, les tables des matières n'affichent pas nécessairement la même chose à différents endroits de la présentation.

D'autres options sont disponibles.

Les tables des matières ont des hyperliens internes au document.

.....
Les colonnes

beamer propose une mise en page (locale) sur plusieurs colonnes.

```
\begin{columns}[t]
  \begin{column}[c]{0.6\textwidth}
    ...
  \end{column}
  \begin{column}[c]{0.35\textwidth}
    ...
  \end{column}
\end{columns}
```

L'option de l'environnement **columns** définit la position horizontale des colonnes qu'il contient : **t**, **b**, **c** (défaut), **T** (alignement haut première ligne).

columns supprime les marges gauche et droite → on dispose de toute la page en largeur.

L'option **onlytextwidth** réduit à la zone de texte.

L'option de l'environnement **column** définit la position horizontale (mêmes options).

On peut placer autant de **column** qu'on veut (peut) dans **columns**.

Les **column** contiennent n'importe quel matériel **LAT_EX** : texte, tableau, graphique...

On peut placer plusieurs **columns** (l'un en dessous de l'autre) dans la même diapositive.

→ À utiliser sans modération pour des vis-à-vis “texte”/“graphique”.

.....
Avancer par étapes

Des commandes et des environnements acceptent un nouvel argument optionnel :

```
\cmd<étapes>([-]{-})
\begin{env}<étapes>([-]{-}) ... \end{env}
```

où les autres arguments conservent leur place et leur nombre.

\cmd et **env** changent de comportement par rapport aux “étapes de la diapositive”.

<étapes> fait référence au compteur **beamerpauses** (au sens **LAT_EX**).

Le compteur **beamerpauses** est défini à 1 au début de l'environnement **frame**.

Sa valeur maximale est définie par les **\cmd<étapes>** et **\begin{env}<étapes>**.

La syntaxe de **<étapes>** est :

m désigne la valeur **beamerpauses = m**.

m-n désigne les valeurs **beamerpauses** entre **m** et **n**.

m- désigne les valeurs **beamerpauses ≥ m**.

-n désigne les valeurs **beamerpauses ≤ n**.

+ désigne la valeur courante de **beamerpauses** et l'incrémente d'une unité après.

<+-> est la syntaxe la plus utilisée.

Plusieurs occurrences de **+** dans **<étapes>** n'incrémentent que d'une unité au total.

. désigne la valeur (**beamerpauses -1**).

(n) après **+** ou **.** ajoute **n**, qui peut être négatif (*offset*).

<étapes1, étapes2, ...> combine plusieurs spécifications.

.
Exemples de la syntaxe des étapes

Quelques exemples de syntaxe “absolue” :

- `<2-4>` désigne les valeurs 2 à 4 de `beamerpauses`.
- `<2->` désigne les valeurs ≥ 2 de `beamerpauses`.
- `<-4>` désigne les valeurs ≤ 4 de `beamerpauses`.
- `<2-4,6->` désigne les valeurs 2 à 4 et ≥ 6 de `beamerpauses`.

Quelques exemples de syntaxe “relative” :

On suppose `beamerpauses` = 3 lorsqu’on rencontre les motifs `<étapes>`.

- `<+->` est remplacé par `<3->`, puis `beamerpauses` = 4 après.
- `<++(3)>` est remplacé par `<3-6>`, puis `beamerpauses` = 4 après.
- `<+(-2)-+(3)>` est remplacé par `<1-6>`, puis `beamerpauses` = 4 après.
- `<.->` est remplacé par `<2->`, puis `beamerpauses` = 3 après.
- `<+-> ... <.->` est remplacé par `<3-> ... <3->`, puis `beamerpauses` = 4 après.

Vocabulaire : (pour rechercher dans la documentation)

- Chaque page `PDF` est un *slide*.
- Une diapositive est un *frame* (environnement `frame`, commande `\frame`).
- Les *slides* d’un même *frame* sont appelés des *overlays*.
- La valeur maximale de `beamerpauses` pour un *frame* est le nombre d’*overlays*.

.
Dévoiler et cacher du texte

`\uncover<étapes>{texte}` affiche `texte` sur les étapes spécifiées.

En dehors de ces étapes, *la place pour `texte` est réservée*.

Il est possible d’afficher en “transparent” le texte caché (voir `\setbeamercolor{background}{white}`).

`\only<étapes>{texte}` affiche `texte` sur les étapes spécifiées.

En dehors de ces étapes, *la place pour `texte` n’est pas réservée*.

`\visible<étapes>{texte}` équivaut à `\uncover` sans l’effet de transparence.

`\invisible<étapes>{texte}` est l’inverse de `\visible`.

`\onslide<étapes>` affiche le texte qui suit sur les étapes spécifiées.

`\onslide<étapes>{texte}` est équivalent à `\uncover`,

`\onslide+<étapes>{texte}` est équivalent à `\visible`,

`\onslide*{étapes}{texte}` est équivalent à `\only`.

⚠ Cette commande change de signification au sein de l’environnement `overprint`.

`\alt<étapes>{texte1}{texte2}` affiche `texte1` sur les étapes spécifiées et `texte2` sinon.

`\temporal<étapes>{texte1}{texte2}{texte3}` affiche `texte1` avant les étapes spécifiées, `texte2` sur les étapes spécifiées, et `texte3` après la dernière étape spécifiée.

Les arguments de ces commandes peuvent contenir du texte, des tableaux, des images et même des définitions de commandes... Elles peuvent s’emboîter.

Des environnements (presque) équivalents sont fournis : `uncoverenv`, `onlyenv`, `visibleenv`, `invisibleenv`, `altnav`.

.....

Quelques commandes gérant les étapes

beamer définit ou redéfinit des commandes compatibles avec la syntaxe **<étapes>** :

\frame, **\begin{frame}** : **<étapes>** désignent les seules étapes retenues.

\textbf, **\textit**, **\textsf**, **\texttt**, **\textbf** acceptent **<étapes>** comme premier argument :
\textbf{<2-4>{En gras de 2 à 4}}

\color{étapes}[modèle]{spécifications} active la couleur définie par le modèle sur les étapes spécifiées.

\includegraphics{étapes}[-]{-} n'insère le graphique qu'aux étapes spécifiées.

\label{étapes}{label} insère le label **label** sur la page correspondant à l'étape spécifiée.

⚠ étapes ne doit contenir qu'une seule étape !

theorem, **definition**, **proof...** : ces environnements acceptent la syntaxe **<étapes>**.

\pause permet de gérer les étapes d'une diapositive en marquant des “pauses”.

.....

Les listes

beamer retient les trois sortes de liste de **LAT_EX** : **itemize**, **enumerate** et **description**.

L'apparence de ces listes et de leurs items peut varier selon le thème choisi.

Les items des listes acceptent l'argument **<étapes>** :

```
\begin{itemize}
  \item<1-> Un
  \item<2-> Deux
\end{itemize}
```

```
\begin{enumerate}
  \item<+-> Un
  \item<+-> Deux
\end{enumerate}
```

```
\begin{description}
  \item<1-3>[Un]
  \item<2-4>[Deux]
\end{description}
```

Autre syntaxe :

```
\begin{itemize}[<+->]
  \item Un
  \item Deux
  \item Trois
\end{itemize}
```

```
\begin{enumerate}[<+->]
  \item Un
  \item<1-> Deux
  \item Trois
\end{enumerate}
```

→ Raccourci qui fait apparaître les items les uns après les autres sauf mention contraire.

L'enumération reprend les options du package **enumerate** (déjà chargé) :

```
\begin{enumerate}[<+->][(i)] ... \end{enumerate}
```

.....
Mettre en valeur du texte

Les thèmes installent une couleur dominante, appelée “couleur de structure”.

\structure<>{-} met du texte dans la couleur de structure.

\structure{Couleur structure} → Couleur structure

\alert<>{-} met en valeur du texte en le colorant en rouge par défaut.

\alert{En alerte !} → En alerte !

beamer propose de mettre en valeur des blocs de texte de plusieurs lignes :

\begin{block}{Un titre}
Du texte en boîte
\end{block}

\begin{alertblock}{Un titre}
Du texte en alerte
\end{alertblock}

\begin{exampleblock}{Un titre}
Du texte en exemple
\end{exampleblock}

Ces trois types de blocs ont chacun une couleur assignée (modifiable).

⚠ Le terme “bloc” est excessif : dans certains thèmes il n'y a pas de bloc dessiné...

Ces trois environnements acceptent un argument <> :

\begin{block}<> ... \end{block}

→ N'affiche le bloc que selon les spécifications de <>, sa place est réservée.

.....
Les espaces réservés

Ajouter ou ôter du texte produit une recomposition de la diapositive → non souhaitable.

\begin{overlayarea}{<dimh>}{<dimv>} ... \end{overlayarea}

réserve une zone de travail de largeur <dimh> et de hauteur <dimv>.

→ Dans cette zone, des \only ne changeront pas la mise en page globale.

\begin{overlayarea}{\textwidth}{3cm}
\only<1> {Lorem ipsum dolor sit amet.}
\only<2>{Fusce pretium ullamcorper neque sit amet luctus.}
\end{overlayarea}

\begin{overprint}[-] ... \end{overprint} réserve un espace vertical de hauteur minimale (calculée) adaptée à son contenu.

La largeur vaut \textwidth sauf mention contraire dans l'argument optionnel.

Les étapes du contenu sont définies par la commande \onslide<> qui se comporte un peu comme \item<>. → Les \onslide<étapes> doivent être disjoints.

\begin{overprint}
\onslide<1> Lorem ipsum dolor sit amet.
\onslide<2> Fusce pretium ullamcorper neque sit amet luctus.
\end{overprint}

..... Les mathématiques

Les mathématiques dans **beamer** sont les mêmes que dans **LAT_EX**.

Les *packages* **amsmath** et **amsthm** sont chargés par défaut (sauf avec l'option **noamsthm**).

Les environnements **theorem**, **corollary**, **definition**, **definitions**, **fact**, **example**, **examples** sont définis et se présentent comme des blocs.

\newtheorem et **\theoremstyle** sont disponibles (syntaxe de **amsthm**).

```
\theoremstyle{plain}
\newtheorem{proposition}{theorem}{Proposition}
\theoremstyle{example}
\newtheorem{remark}{theorem}{Remarque}
```

Ces environnements acceptent automatiquement un argument **<>** :

\begin{definition}<>{-} ... \end{definition}

→ Apparition du bloc en entier.

→ Insérer des **\uncover<>{-}** à l'intérieur pour affiner les étapes.

..... Boutons et hyperliens internes

Il est possible de définir des hyperliens internes entre pages.

La syntaxe est celle du *package* **hyperref** :

\hypertarget{étapes}{label interne}{texte} définit un but pour un hyperlien interne, de clé interne *label interne*, de but la *page PDF correspondant à <étapes>*.

En dehors des spécifications de **<étapes>**, **texte** n'est pas affiché.

\hyperlink{étapes}{label interne}{texte} rend *texte* cliquable dans les spécifications **<étapes>** et renvoie vers le but défini par *label interne*.

Pour rendre les choses plus jolies, on peut placer des boutons dans **\hyperlink** :

\beamerbutton{texte du bouton} dessine un bouton.

\beamergotobutton{texte du bouton} bouton avec une flèche vers la droite.

\beamerskipbutton{texte du bouton} bouton avec une double flèche vers la droite.

\beamerreturnbutton{texte du bouton} bouton avec une flèche vers la gauche.

```
\hyperlink<3->{labelA}{\beamergotobutton{C'est par là !}}
```

De nombreux buts sont déjà définis par **beamer** auxquels on peut accéder avec les commandes :

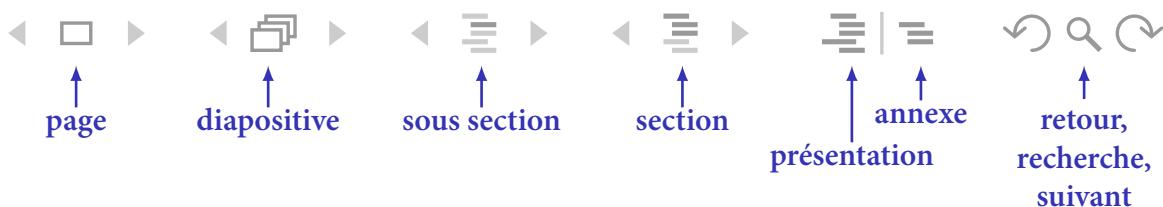
\hyperlinkslideprev{texte}, **\hyperlinkslidenext{texte}**,

\hyperlinkframestart{texte}, **\hyperlinkframeend{texte}**,

\hyperlinkframestartnext{texte}, **\hyperlinkframeendprev{texte}**...

La barre de navigation

Par défaut, une barre de navigation est insérée en bas à droite des diapositives. Elle permet de naviguer dans le document grâce à des hyperliens internes.



page est une page PDF,
diapositive symbolise l'ensemble des pages qui composent une unique diapositive,
sous section est un début de sous section,
section est un début de section,
présentation désigne la fin de la présentation (avant l'annexe),
annexe est le début de l'annexe.

retour, recherche et suivant permettent de naviguer directement dans le fichier PDF.

Les flèches à droite et à gauche des symboles font naviguer d'une entité à une autre.

On peut ôter cette barre en insérant dans le préambule :

```
\setbeamertemplate{navigation symbols}{}
```

Interrompre et reprendre une diapositive

On peut interrompre une diapositive, en insérer d'autres, puis la continuer plus loin.

Tout le code de la diapositive interrompue est donné dans son environnement frame.

- Les spécifications <étapes> désignent son affichage la première fois.
- Un label interne frametlabel lui est attribué pour l'utiliser plus tard avec la commande :

```
\againframe<étapes>{frametlabel}
```

Possible d'utiliser plusieurs fois \againframe<->{-} pour la même diapositive.

```
\begin{frame}<1-5>[label=diapo]{Titre}
  Contenu total
\end{frame}

  autres diapositives

\againframe<6-7>{diapo}

  autres diapositives

\againframe<8->{diapo}
```

⚠ Éviter des valeurs de beamerpauses qui se chevauchent (“multiply-defined labels”).

.
Quelques trucs

Il est possible d'écraser verticalement une diapositive qui serait trop longue :

```
\begin{frame}[squeeze]{Titre}
  Texte un peu trop long
\end{frame}
```

```
\begin{frame}[shrink=5]{Titre}
  Texte un peu trop long
\end{frame}
```

`squeeze` réduit certains espaces verticaux à zéro.

`shrink` réduit globalement le contenu de la diapositive du pourcentage donné.

→ Le contenu de la diapositive est réellement réduit, y compris la taille des caractères !

Ajouter automatiquement une table des matières à chaque début de section :

```
\AtBeginSection[]{
\begin{frame}
  \frametitle{\insertsection}
  \tableofcontents[currentsection,hideothersubsections]
\end{frame}
}
```

`\AtBeginSection` installe du matériel à chaque début de section.

Il existe aussi `\AtBeginPart`, `\AtBeginSubsection` et `\AtBeginLecture`

.
Autres fonctionnalités à explorer

Le package **TikZ** est compatible avec **beamer** (même créateur : Till Tantau).

Il fonctionne avec la commande `\uncover<>{-}`.

Il est possible de créer des graphiques dynamiques.

Il est possible de relier entre eux des éléments dans une diapositive.

→ Lire la documentation de **TikZ** pour plus de renseignements.

Position absolue : avec le package **textpos**, il est possible de placer des objets de façon absolue dans une diapositive.

→ Voir la documentation de ce package.

Lire la documentation de **beamer** pour plus de renseignements.

.....
Le package multimedia

Le package **multimedia** est fourni avec **beamer**.

Il permet d'insérer des vidéos et du son dans des documents **PDF**.

\movie[-]{poster}{fichier vidéo} :

- Le code *poster* représente du texte ou une image (\includegraphics).
- Il symbolise l'élément multimédia sur la page.
- En cliquant sur *poster*, le fichier multimédia *fichier vidéo* se lance.
- Les options gèrent le comportement de la vidéo et l'aspect de *poster*.

⚠ Certains visualiseurs **PDF** ne sont pas capables de gérer des fichiers multimédia.

```
\usepackage{multimedia}
...
\movie[externalviewer, width=120pt]{%
  \includegraphics[width=120pt]{image.jpg}%
  {video.mp4}}
```

Avec l'option **externalviewer**, le visualisateur **PDF** délègue au système d'exploitation le soin de trouver un lecteur multimédia.

- Il faut régler ce problème au niveau du système.
- Grande portabilité de cette solution.

La documentation de **multimedia** est insérée dans celle de **beamer**.

.....
Du FLASH dans beamer

⚠ Ce qui suit ne fonctionne qu'avec **Adobe Reader** version 9 ou 10 et **pdflatex**.

Ça ne semble pas fonctionner sous **LINUX**...

Le package **flashmovie**⁶² permet d'insérer des animations **FLASH**⁶¹ dans **LATEX**.

Avec **beamer**, il faut l'appeler *avant* le \documentclass (problème de compatibilité) :

```
\RequirePackage{flashmovie}
\documentclass[10pt]{beamer}
...
\flashmovie[width=280pt, height=80pt, auto=1]{animation.swf}
```

Insertion de vidéos (.mp4, .flv) grâce à des lecteurs **FLASH** : **JW Player**⁶³ ou **FLV Player**⁶⁴ :

```
\flashmovie[width=200pt, height=110pt, image=preview.jpg,%
           engine=jw-player, auto=0, controlbar=1]{video.mp4}
```

Le fichier **player_flv_maxi.swf** du lecteur **FLV Player** est déjà installé.

⚠ **JW Player** est la solution la plus compatible avec **beamer**.

Seul le fichier **player.swf** de **JW Player** est nécessaire.

Le fichier **PDF** contient les fichiers **FLASH** et les vidéos. → Un seul fichier à distribuer.

Lire la documentation dans le fichier **flashmovie.sty** lui-même.

Autre package semblable : **media9**⁶⁵.

..... Généralité sur les thèmes

beamer fournit des thèmes pour modifier l'aspect de la présentation.

Un thème appelé **default** est installé automatiquement (pas nécessaire de le charger).

Les thèmes sont classés en 4 catégories, selon ce qu'ils modifient :

Thèmes extérieurs modifient l'aspect des éléments extérieurs de la présentation : entête et pied, barres de navigations, symboles de navigation, titre des diapositives...

`\useoutertheme[-]{-}` dans le préambule charge un tel thème.

Thèmes intérieurs modifient l'aspect des éléments intérieurs de la présentation : page de titre, listes, blocs, théorèmes, figures et tables, *footnotes*, entrées bibliographiques...

`\useinnertheme[-]{-}` dans le préambule charge un tel thème.

Thèmes de couleurs modifient la couleur des éléments.

`\usecolortheme[-]{-}` dans le préambule charge un tel thème.

Thèmes couleurs extérieurs (noms : animaux marins...).

Thèmes couleurs intérieurs (noms : fleurs...).

Thèmes couleurs complets (noms : animaux volants...).

Thèmes de fontes modifient la typographie des éléments.

`\usefonttheme[-]{-}` dans le préambule charge un tel thème.

On peut installer des *packages* de polices.

Les thèmes extérieurs et intérieurs ne touchent pas aux couleurs et à la typographie.

..... Les thèmes clés en main

Des **thèmes de présentation** sont aussi définis.

Ils regroupent de façon cohérente des ensembles de 4 thèmes ci-dessus.

`\usetheme[-]{-}` dans le préambule charge un tel thème.

Ces thèmes de présentation portent des noms de villes, sauf **default** et **boxes**.

boxes est conçu pour ajouter des boîtes dans la barre de navigation et le pied.

```
\usetheme[headheight=2.5ex]{boxes}
  \addheadbox{section in head/foot}%
    {\footnotesize\quad\insertsection}
  \addfootbox{structure}%
    {\tiny\quad \insertshorttitle, \insertshortauthor}
```

On peut surcharger un thème clés en main en appelant un “sous thème” :

```
\usetheme{Warsaw}
\usecolortheme{crane}
```

Voir la documentation de **beamer**, et les sites

Beamer Theme Matrix⁶⁶ et **Latex Beamer Themes**⁶⁷.

Télécharger le **Beamer Cheat Sheet**⁶⁸ sur mon site.

.....
Conseils divers

- La mise en page sous forme de listes est très efficace pour segmenter les idées.
- Le découpage d'une diapositive en étapes permet de synchroniser parole et visuel.
Mais éviter les éléments cachés sous forme “transparent”...
- Les couleurs :
 - Ne surtout pas abuser de couleurs diverses et variées.
→ Codifier une fois pour toute le choix des couleurs.
 - Ne pas jouer trop finement avec les effets de couleurs : les vidéoprojecteurs ne sont pas fidèles en matière de rendu des couleurs et des contrastes.
→ Tester certaines couleurs avant de les adopter (ou de les rejeter).
 - Penser aux daltoniens...
- Si le fond des diapositives est coloré, prévoir des figures avec fond transparent.
→ Sinon encadrer l'image pour marquer ses bords.
- Ne pas surcharger une diapositive :
 - dans son contenu ;
 - dans les éléments structurants, informatifs, graphiques et visuels qui la composent.
- Faire la chasse aux polices à empattement, surtout dans le mode mathématique.
→ Éradiquer les `\mathrm{-}` de toutes les commandes personnelles !
- Inutile de numéroter les équations.

URLs des liens cités dans le texte

- ⠁<http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
- ⠁<http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>
- ⠁<http://home.gna.org/latexrefman/>
- ⠁<http://www.ctan.org/characterization/primary/>
- ⠁<http://en.wikibooks.org/wiki/TeX>
- ⠁<http://detexify.kirelabs.org/classify.html>
- ⠁<http://www.tug.org/pracjourn/>
- ⠁<http://latex-community.org/>
- ⠁<http://texblog.net/>
- ⠁<http://science.thilucmic.fr/spip.php?article5>
- ⠁<http://www.latex-community.org/forum/>
- ⠁[http://fr.wikipedia.org/wiki/Bloc-notes_\(Windows\)](http://fr.wikipedia.org/wiki/Bloc-notes_(Windows))
- ⠁<http://fr.wikipedia.org/wiki/Emacs>
- ⠁[http://fr.wikipedia.org/wiki/Pico_\(logiciel\)](http://fr.wikipedia.org/wiki/Pico_(logiciel))
- ⠁<http://fr.wikipedia.org/wiki/Textedit>
- ⠁<http://fr.wikipedia.org/wiki/Word>
- ⠁<http://fr.wikipedia.org/wiki/Openoffice>
- ⠁http://fr.wikipedia.org/wiki/Publication_assistée_par_ordinateur
- ⠁<http://fr.wikipedia.org/wiki/Indesign>
- ⠁<http://fr.wikipedia.org/wiki/QuarkXPress>
- ⠁<http://fr.wikipedia.org/wiki/Scribus>
- ⠁<http://fr.wikipedia.org/wiki/Html>
- ⠁<http://fr.wikipedia.org/wiki/TeX>
- ⠁<http://www.latex-project.org/>
- ⠁[http://fr.wikipedia.org/wiki/DVI_\(TeX\)](http://fr.wikipedia.org/wiki/DVI_(TeX))
- ⠁<http://fr.wikipedia.org/wiki/Pdf>
- ⠁<http://www.tug.org/texlive/>
- ⠁<http://www.tug.org/texworks/>
- ⠁<http://pages.uoregon.edu/koch/texshop/>
- ⠁http://www.xm1math.net/texmaker/index_fr.html
- ⠁<http://www.gnu.org/software/emacs/>
- ⠁<http://aquamacs.org/>
- ⠁<http://www.texniccenter.org/>
- ⠁<http://www.winedt.com/>
- ⠁<http://www.gnu.org/software/auctex/>
- ⠁<http://jabref.sourceforge.net/>
- ⠁<http://home.gna.org/kbibtex/>
- ⠁<http://aspell.net/>
- ⠁<http://www.tug.org/mactex/>
- ⠁<http://bibdesk.sourceforge.net/>
- ⠁<http://code.google.com/p/mactlmgr/>
- ⠁<http://cocoaspell.leuski.net/>
- ⠁<http://www.ctan.org/search.html>
- ⠁<http://www.ctan.org/>
- ⠁<http://fr.wikipedia.org/wiki/Windows-1252>
- ⠁<http://fr.wikipedia.org/wiki/MacRoman>
- ⠁http://fr.wikipedia.org/wiki/ISO_8859-1
- ⠁<http://www.unicode.org/>
- ⠁<http://www.tug.org/pracjourn/2006-1/robertson/>
- ⠁<http://www.gust.org.pl/projects/e-foundry/tex-gyre/>
- ⠁<http://www.tug.dk/FontCatalogue/>
- ⠁http://www.ctan.org/tex-archive/info/Free_Math_Font_Survey/survey.html
- ⠁[http://fr.wikipedia.org/wiki/Point_\(unité\)](http://fr.wikipedia.org/wiki/Point_(unité))
- ⠁http://fr.wikipedia.org/wiki/Point_pica
- ⠁http://fr.wikipedia.org/wiki/Point_Didot
- ⠁http://fr.wikipedia.org/wiki/Unité_Cicéro
- ⠁<http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf>
- ⠁<https://bitbucket.org/rivanvx/beamer/wiki/Home>
- ⠁http://en.wikipedia.org/wiki/List_of_PDF_software
- ⠁http://fr.wikipedia.org/wiki/Adobe_Acrobat
- ⠁http://fr.wikipedia.org/wiki/Adobe_Flash

URLs des liens cités dans le texte

- ⠁⁶²<http://www.ctan.org/tex-archive/macros/latex/contrib/flashmovie>
- ⠁⁶³<http://www.longtailvideo.com/players/jw-flv-player/>
- ⠁⁶⁴<http://flv-player.net/>
- ⠁⁶⁵<http://www.ctan.org/tex-archive/macros/latex/contrib/media9>
- ⠁⁶⁶<http://www.hartwork.org/beamer-theme-matrix/>
- ⠁⁶⁷<http://latex.simon04.net/>
- ⠁⁶⁸<http://science.thilucmic.fr/spip.php?article35>