

ECUE “L^AT_EX”

Édition scientifique avec L^AT_EX

Gloria Faccanoni

 <https://moodle.univ-tln.fr/course/view.php?id=7459>

Dernière mise-à-jour : Lundi 29 janvier 2024



Le but de ce document est de guider le nouvel utilisateur de \LaTeX pour une prise en main efficace, en évitant autant que possible les mauvaises habitudes. À noter, **il ne constitue pas un manuel**, mais d'un support aux cours/TP. Veuillez noter qu'il est continuellement mis à jour. Si vous découvrez des erreurs, n'hésitez pas à me les signaler, je vous en remercie.



Gloria FACCANONI

IMATH Bâtiment M-117
Université de Toulon
Avenue de l'université
83957 LA GARDE - FRANCE

☎ 0033 (0)4 83 16 66 72

✉ gloria.faccanoni@univ-tln.fr
🌐 <http://faccanoni.univ-tln.fr>

Table des matières

Syllabus	5
Évaluation	7
1 Introduction	9
1.1 Installation d'une distribution, d'un éditeur dédié et d'un pdf viewer	11
1.2 Configuration de Texmaker	11
2 Premiers pas en L^AT_EX	15
2.1 Structure de base d'un fichier L ^A T _E X	15
2.2 Espaces dans le code source	17
2.3 Commentaires	17
2.4 Espaces entre alinéas dans le document final	17
2.5 Le système de packages de L ^A T _E X	18
2.5.1 Pour aller plus loin	20
2.6 Caractères spéciaux	20
2.7 Titre, table des matières et sections	21
2.8 Inclusion de fichiers L ^A T _E X	22
2.9 Étiquettes, références croisées et liens externes	25
2.10 Notes de bas de page ou dans la marge	27
2.11 Listes à puce, énumérations et descriptions	27
2.12 Afficher des codes sources avec listings	29
2.13 Objets flottants : figures et tableaux	31
2.13.1 Insérer des images, les encapsuler dans une figure	32
2.13.2 Insérer des tableaux, les encapsuler dans une table	34
2.14 Mettre son document sur plusieurs colonnes	36
2.14.1 multicol(s)	36
2.14.2 minipage	38
2.15 Le formatage du texte ou comment écrire en <i>italique</i> en gros en petit en chasse fixe...	39
2.16 Définition d'environnements et de commandes personnelles	42
2.17 Pour aller plus loin	43
3 Faire des bibliographies avec L^AT_EX	45
3.1 Étape n°① : création d'une base bibliographique dans un fichier .bib	45
3.2 Étape n°② : appel de la bibliographie dans le fichier .tex	46
3.3 Étape n°③ : compilations	47
3.4 Pour aller plus loin : le package BibL ^A T _E X	48
4 Mathématiques	53
4.1 Documents avec théorèmes, propositions, etc.	53
4.2 Les différents modes mathématiques	55
4.3 Indices et exposantes	57
4.4 Racine carrée, racine n-ième	58
4.5 Texte et espaces dans une formule <i>displaystyle</i>	58
4.6 Symboles d'usage courant	59
4.7 Points de suspension	60

Table des matières

4.8	Fonctions mathématiques	60
4.9	Flèches extensibles	61
4.10	Fractions et coefficients binomiaux	61
4.11	Lettres grecques	62
4.12	Grands opérateurs : intégrales, sommes, produits, etc.	63
4.13	Modules de congruences	64
4.14	Placer au-dessus ou en-dessous	64
4.15	Accents mathématiques	64
4.16	Délimiteurs	65
4.17	Alphabets mathématiques	66
4.18	Tableaux mathématiques	67
4.18.1	Distinction de cas	68
4.18.2	Matrices	68
4.19	Formules sur plusieurs lignes et alignements	69
4.20	Exercices de synthèse	70
4.21	Pour aller plus loin	71
5	Présentations vidéo-projetées	73
5.1	Introduction	73
5.2	Choix du thème et création de la première diapositive	73
5.3	Les blocs	76
5.4	Les listes	80
5.5	Ajouter des colonnes	82
5.6	Les images	83
5.7	La barre de navigation	83
5.8	Mettre en avant des portions de texte	83
5.9	only, uncover, visible etc.	84
5.10	Pour aller plus loin	88
6	Compléments	89
6.1	Dessiner avec TikZ	89

Syllabus

\LaTeX est un système de rédaction de documents largement adopté par les scientifiques pour la réalisation de livres, articles de recherche, présentations, posters, supports de cours (comme celui-ci), feuilles d'exercices, notes de travail... Son omniprésence s'explique non seulement par sa capacité à mettre en forme des équations mathématiques complexes, mais aussi par la qualité professionnelle de ses résultats (aussi bien pour le texte que pour les formules mathématiques), permettant une publication directe. De plus, sa façon de structurer les documents, en séparant le contenu (le sens du texte) de la mise en page, ainsi que la "programmation" de tâches fastidieuses (comme les références croisées), favorisant ainsi une grande productivité.

☞ Durée

Le module est organisé en 5 séances de cours-TP, chacune d'une durée de 3h.

☞ Objectifs

Le but de ce cours n'est en aucun cas de couvrir tous les aspects de \LaTeX : d'une part parce que cela serait impossible, d'autre part parce que certaines de ses fonctionnalités ne seraient sans doute pas immédiatement utiles.

En revanche, les objectifs sont les suivants :

- se familiariser avec la rédaction scientifique et les bases de la composition avec \LaTeX ,
- établir des bases solides de connaissances, si possible exempt de mauvaises habitudes,
- donner des pistes pour être en mesure par la suite de continuer les apprentissages de manière autonome.

☞ Contenu pédagogique

Le programme du cours comprendra :

1. *Introduction* : installation, structure d'un fichier source, types de document, structure du document, gestion automatique de la table des matières et des références.
2. *Mise en page* : prise en charge de la langue française, listes (à puce, énumérations, descriptions), tableaux et figures (flottants), notes de bas de page...
3. *Mathématiques* : mise en forme des formules mathématiques, rédactions de théorèmes, exercices...
4. *Compléments* : gestion automatique de la bibliographie, présentation de codes, présentations vidéo-projetées, dessins avec \LaTeX ...

Évaluation

L'évaluation des connaissances est un DM. Il consiste en la rédaction à la maison d'un document de 6 à 10 pages (hors pages blanches, page de titre ou de table des matières lorsque la classe `scrbook` ou `scrreprt` est utilisée).

Ce document pourrait, par exemple, être inspiré par la démonstration d'un résultat fondamental dans l'un de vos cours. Il faudra alors rédiger l'énoncé et la démonstration, inclure une bibliographie, des illustrations, voire une mini-biographie (avec photo) des mathématiciens/informaticiens impliqués dans ce résultat. Si vous manquez d'idées, vous pouvez vous appuyer sur une page Wikipédia (souvent fournissant tous les éléments requis ci-dessous). Dans tous les cas, veuillez citer correctement vos sources! Je vais principalement évaluer la bonne utilisation de \LaTeX , sans me focaliser spécifiquement sur le contenu. Cependant, **soyez attentifs aux fautes de français** et **évitiez le plagiat** (vous pouvez bien sûr utiliser des textes d'autres personnes, mais veuillez correctement indiquer la source, et assurez-vous de ne pas simplement copier un rapport déjà rédigé par d'autres personnes). Je ne devrais pas à avoir à l'indiquer, mais, bien sûr, **pas de rapport intégralement rédigé par ChatGPT ou analogue**. Vous pouvez lui demander de reformuler des paragraphes que vous avez rédigés, mais rien de plus.

Ce devoir est assez simple, mais il est important de ne pas attendre le dernier jour pour le réaliser. Il est nécessaire de prendre du temps afin de trouver le sujet et éventuellement collecter les informations nécessaires, garantissant ainsi la présence de tous les éléments obligatoires.



Éléments obligatoires

- Un titre (créé avec `\maketitle` ou l'environnement `{titlepage}`) et un résumé (formé avec l'environnement `{abstract}`).
- Une table des matières (élaborée avec `\tableofcontents`) contenant des liens hypertextuels (en utilisant le package `hyperref`).
- Des sections, sous-sections etc. (créées avec `\section`, `\subsection` ...).
- Des listes numérotées (avec l'environnement `{enumerate}` ou analogue) et non numérotées (avec l'environnement `{itemize}` ou analogue).
- Des notes en bas de page et des notes à marge (utilisation des commandes `\footnote`, `\marginpar` ...).
- Des en-têtes et pieds de page personnalisés avec soit le package `scrlayer-scrpage` dont la documentation doit être consultée, soit le package `fancyhdr`.
- Des théorèmes et/ou exercices, exemples etc. (avec utilisation d'un package, par exemple `amsthm`).
- Des formules mathématiques *inline* (avec `$. . $`).
- Des formules mathématiques *display* sans numérotation (avec `\[. . \]`).
- Des formules mathématiques *numérotées* (avec l'environnement `{equation}` ou analogue) avec des références à ces équations (utilisation des commandes `\label` et `\eqref`).
- Des figures/schémas (importés avec la commande `\includegraphics` ou créés, par exemple, avec le package `TikZ`) dans un élément flottant (environnement `{`

figure}) avec une légende (commande `\caption`) et des références à cette figure (commandes `\label` et `\ref`).

- Des tableaux réalisés avec l'environnement `tabular` (ou similaire) dans un élément flottant (environnement `{table}`) avec une légende (commande `\caption`) et des références à ces tableaux (commandes `\label` et `\ref`).
- Des commandes personnelles judicieusement choisies créées à l'aide de `\newcommand`, et des environnements personnels créés avec `\newenvironment`.
- Une bibliographie (créée avec BibTeX ou BibLaTeX) et au moins une référence à une entrée de la bibliographie (avec `\cite`).
- Un algorithme ou du code source (en utilisant un package spécifique, par exemple `listings`).



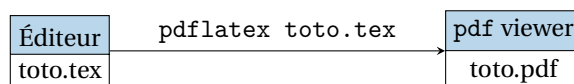
Les éléments suivants seront considérés comme des fautes graves :

- utiliser `$$...$$` au lieu de `\[...\]` ;
- utiliser `{eqnarray}` au lieu de `{align}` (ou analogues) ;
- utiliser `\newline` (et analogues) ou `\\` (en dehors d'un tableau) ;
- utiliser des commandes telles que `\newpage` ou `\clearpage` (et analogues) ;
- utiliser `\nocite*` ;
- imposer le placement d'un élément flottant ;
- maintenir des packages inutilisés et dé-commentés dans le préambule.

1 Introduction

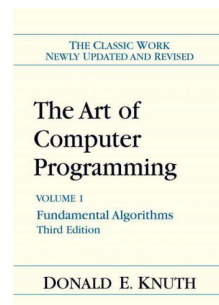
En 1977, Donald Ervin KNUTH est en train d'éditer son livre *The Art of Computer Programming* mais il est mécontent du rendu obtenu. Il décide alors de créer son propre système de composition de texte, nommé $\text{T}_{\text{E}}\text{X}$ (prononcé *tek*, du grec ancien $\tau\acute{\epsilon}\chi\nu\eta$). Son utilisation étant assez ardue, un autre informaticien, Lesli LAMPORT, commence à développer un jeu de *packages* (ensemble de commandes) permettant d'accélérer la saisie de documents en $\text{T}_{\text{E}}\text{X}$. Le mot $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (prononcé *latek*) est un jeu de mot basé sur $\text{T}_{\text{E}}\text{X}$ auquel sont rajouté les initiales du concepteur de $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.¹

Un document en $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ est un fichier source qui doit être compilé avant d'être lisible par tous. Nous compilerons toujours avec `pdflatex` pour produire un fichier pdf (il y a d'autres formats de sortie et d'autres fichiers qui sont produits lors d'une compilation, mais on ne s'y intéressera pas dans ce document).



Contrairement aux logiciels où l'on voit directement à l'écran ce que l'on tape, dits de type WYSIWYG (*What You See Is What You Get*, ce que voyez est ce que vous obtenez), tel les documents produits avec OpenOffice ou Microsoft Word, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ sépare la forme du contenu et il demande au rédacteur de se concentrer uniquement sur la *structure logique* de son document et sur son contenu, tandis que la mise en page du document (césure des mots, alinéas, styles des titres, etc.) est laissée au logiciel. De ce fait, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ requiert un apprentissage initial plus important que celui qui est nécessaire pour les logiciels de type WYSIWYG, du moins pour la mise en page de petits documents simples. Mais une fois cette phase d'apprentissage accomplie, le fait de se concentrer sur le contenu et de laisser à $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ le soin de présenter le document devient très appréciable : la qualité du document produit est élevée (respect des règles typographiques) et la gestion des références bibliographiques (avec une base de donnée $\text{Bib}_{\text{T}_{\text{E}}\text{X}}$), les numérotations et la table des matières sont cohérentes sans qu'on ait à s'en soucier.

¹ Il existe une vaste famille d'outils qui sont désormais dérivés de $\text{T}_{\text{E}}\text{X}$: $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{pdf}_{\text{T}_{\text{E}}\text{X}}$, $\text{pdf}_{\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}}$, $\text{Xe}_{\text{T}_{\text{E}}\text{X}}$, $\text{Lua}_{\text{T}_{\text{E}}\text{X}}$, $\text{Con}_{\text{T}_{\text{E}}\text{X}}$ t... L'ancêtre de tout cela est bien sûr le programme $\text{T}_{\text{E}}\text{X}$ original de KNUTH. Cela prend un document en texte brut et le transforme en un document magnifiquement composé. Cependant, les commandes de $\text{T}_{\text{E}}\text{X}$ sont encore assez basiques et il n'est pas facile de faire des choses compliquées avec elles. Pour y parvenir, Leslie LAMPORT a créé $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ au début des années 1980 pour fournir un langage de travail de plus haut niveau que $\text{T}_{\text{E}}\text{X}$. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ est un ensemble de commandes définies en termes de commandes $\text{T}_{\text{E}}\text{X}$ sous-jacentes. En créant un système de packages standardisé pour $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, Leslie LAMPORT a permis à la communauté de se développer considérablement. Il existe désormais des milliers et des milliers de packages $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ disponibles pour vous permettre de composer n'importe quoi, des sous-figures aux modèles de tricot. Il existe également de nombreuses classes de documents disponibles pour différents types de documents, qu'il s'agisse d'un livre, d'un rapport de laboratoire ou d'un CV. De nombreux éditeurs et revues disposent de leurs propres classes de documents personnalisées qui stylisent leurs documents comme ils le souhaitent. Le programme $\text{T}_{\text{E}}\text{X}$ d'origine et son dérivé $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ génèrent un document dans un format appelé DVI (DeVice Independent format), qui peut ensuite être transformé en fichier PostScript pour l'impression. Cependant, le format PDF est apparu en 1993 et nous pouvons constater aujourd'hui qu'il s'est clairement imposé comme le meilleur format par rapport au PostScript. De nombreuses fonctionnalités améliorent le PDF, telles que des hyperliens entre les sections, une section de métadonnées qui vous permet de voir une table des matières dans la partie gauche de votre visionneuse PDF et la prise en charge d'une gamme plus large et plus moderne de formats d'image. $\text{pdf}_{\text{T}_{\text{E}}\text{X}}$ et $\text{pdf}_{\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}}$ sont des modifications de $\text{T}_{\text{E}}\text{X}$ et $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ qui permettent de sortir directement au format PDF et ainsi de profiter de ces fonctionnalités supplémentaires. En 2004, Jonathon KEW a créé $\text{Xe}_{\text{T}_{\text{E}}\text{X}}$, qui est une autre modification du moteur $\text{T}_{\text{E}}\text{X}$ sous-jacent, cette fois pour prendre en charge une plus large gamme de caractères au-delà des simples chiffres et lettres anglais, et pour inclure la prise en charge des formats de polices modernes. Cela rend l'écriture dans des langues étrangères beaucoup plus facile et permet également d'utiliser des polices dans $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ qui n'étaient traditionnellement disponibles que pour les traitements de texte. $\text{Lua}_{\text{T}_{\text{E}}\text{X}}$ est une tentative d'étendre le programme $\text{T}_{\text{E}}\text{X}$ original avec un langage de programmation plus sensé. Bien qu'en principe vous puissiez faire absolument n'importe quoi à partir de $\text{T}_{\text{E}}\text{X}$, en pratique, c'est un langage de programmation très difficile à utiliser. $\text{Con}_{\text{T}_{\text{E}}\text{X}}$ est un autre système créé en 1990 par Hans HAGEN. Il vise à fournir une interface simple aux fonctionnalités de typographie avancées.



Apprendre à produire des documents avec \LaTeX demande un peu d'investissement, même s'il est possible d'apprendre au fur et à mesure de ses besoins. Cela vient du fait que c'est un système complètement différent des traitements de texte qu'on a l'habitude d'utiliser. Par exemple, pour écrire *différent* en italique, on n'a pas cliqué sur un bouton dans un menu comme on le ferait avec `OpenOffice`, mais en gros, on a saisi dans le fichier source quelque chose comme `\italique{différent}`. Tout ce document est construit selon ce principe, ensuite \LaTeX compile le fichier source et le transforme en un beau document pdf. À première vue, cela semble plus contraignant et moins convivial qu'un traitement de texte, mais alors pourquoi se mettre à \LaTeX ?

- **Pour produire des documents scientifiques de qualité.** C'est pour cela que \LaTeX a été conçu au départ. On peut écrire facilement des formules mathématiques, dessiner des arbres, des molécules, des diagrammes commutatifs, de la musique, etc. Dans chaque communauté de scientifiques, certains ont créé des packages, disponibles sur Internet gratuitement et personnalisables, pour adapter \LaTeX à leurs besoins. À cette adresse vous trouverez quelques exemples : <http://www.tug.org/texshowcase/>
- **Pour la qualité typographique.** Les traitements de texte classiques n'ont pas été conçus avec l'expertise de typographes, contrairement au système \LaTeX . On en est assez vite convaincu à la vue d'un document \LaTeX : espace entre les caractères, césures, arrangement des paragraphes, mais également disposition des figures dans le texte, domaine pour lequel les traitements de texte sont très mal conçus.
- **Pour créer des gros documents.** C'est pour cette qualité de \LaTeX que l'investissement est le plus rentable. On peut lui laisser la gestion de toutes les choses compliquées liées à la production de gros documents (livres, rapports de recherche, mémoires de stage, manuscrits de thèse...). En particulier,
 - il numérote automatiquement les sections, sous-sections, appendices, figures, formules, exercices, théorèmes, notes de bas de page, etc.;
 - il crée tout seul la table des matières, la liste des figures et des tableaux;
 - on peut numérotter très facilement les équations, les formules, les tableaux, les théorèmes, le livres dans la bibliographie... puis faire référence à ces numéros et à la page où ils apparaissent. Même si le document est retouché (par exemple, même si d'autres équations numérotées sont insérées à divers endroits dans le document), le document final restera cohérent;
 - il gère très bien la disposition des figures et des tableaux dans un texte;
 - on peut fusionner très facilement plusieurs documents, cela permet à plusieurs personnes de travailler sur des chapitres différents d'un même document. Par ailleurs, les documents \LaTeX ne prennent que très peu de place sur le disque, contrairement aux documents produits par un traitement de texte.
- **Pour la pérennité.** C'est un critère déterminant pour un livre ou une thèse. Rien ne permet de dire qu'un document écrit avec `Microsoft Word` aujourd'hui puisse être parfaitement lisible (et modifiable) dans 10 ou 15 ans. Au gré des versions, des options disparaissent ou sont créées. Ce n'est pas le cas avec \LaTeX . Les modifications qui y sont apportées ne se font jamais au détriment des utilisateurs.
- **Pour la souplesse.** Le principe même de \LaTeX c'est un noyau commun, qui permet de créer tous les documents simples, et la possibilité de créer des nouveaux modules adaptés à des besoins particuliers. Concevoir ces modules demande beaucoup de talent en programmation mais dans chaque communauté ont été développées des bibliothèques spécifiques qui sont disponibles sur Internet : mathématiques, informatique, chimie, mais aussi partitions de musique, parties d'échecs, russe, grec, etc. On peut donc utiliser simplement \LaTeX en se servant des modules créés par d'autres utilisateurs sans les concevoir soi-même. En revanche, il est très simple de créer des petites commandes adaptées à ses besoins. Par exemple, si j'en ai assez de saisir au clavier «Université de Toulon», je peux créer une commande `\UTLN`. À chaque fois que \LaTeX va lire cette commande, il va automatiquement la traduire en «Université de Toulon».

- **Parce que c'est universel.** Pour échanger des documents produits avec \LaTeX , on peut les transformer en fichier pdf ou même HTML. Peu importe que la personne avec qui on travaille utilise un PC avec Windows, Linux, Mac, Android. Et bien sûr, tout ça est entièrement gratuit.

1.1 Installation d'une distribution, d'un éditeur dédié et d'un pdf viewer

Pour utiliser \LaTeX nous avons besoin d'une distribution \LaTeX (le « compilateur ») et d'un éditeur de texte. Certains éditeurs sont plus adaptés que d'autres et proposent des fonctionnalités plus ou moins intéressantes, telles que la coloration syntaxique, l'auto-complétion de certaines balises, etc. Durant cette formation, nous utiliserons l'éditeur de texte Texmaker qui est disponible sur les trois plate-formes principales (Windows, Linux, Mac).

<https://tug.org/begin.html>

Installation sous Ubuntu

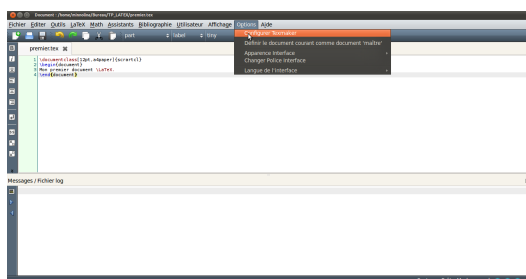
- Installation de la distribution TeXlive : dans *Synaptic* on fait une recherche en rentrant comme mot clé `texlive`.
- Installation de l'éditeur Texmaker : dans *Synaptic* on fait une recherche en rentrant comme mot clé `texmaker`.
- Installation d'un viewer pdf : dans *Synaptic* on fait une recherche en rentrant comme mot clé `Okular` ou `Evince`...

Installation sous Windows

- Télécharger la distribution MiKTeX à l'adresse <https://miktex.org/download>
- Suivre les instructions pour l'installation de la distribution MiKTeX (choisir la version complète) : <https://miktex.org/howto/install-miktex>
- Télécharger et suivre les instructions pour l'installation de l'éditeur Texmaker : <https://www.xmimath.net/texmaker/>
- Installer un viewer pdf (par exemple Acrobat Reader).

1.2 Configuration de Texmaker

- Pour commencer on va démarrer notre éditeur de texte (sous Ubuntu : menu «Applications» → menu «Bureautique» → «Texmaker»). Une nouvelle fenêtre va s'ouvrir, c'est l'éditeur.
- Avant d'écrire le premier document, configurons un raccourci de l'éditeur pour qu'il fasse appelle à `pdflatex` et qu'il lance la visualisation du pdf automatiquement : menu «Option» → «Configurer Texmaker»; une nouvelle fenêtre va s'ouvrir, dans cette fenêtre cliquer sur «Compilation rapide» et cocher «PdfLaTeX + View pdf».

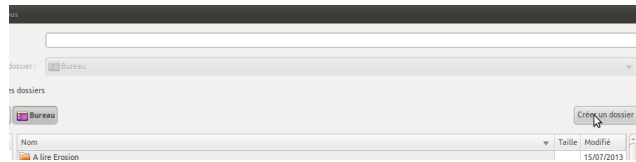


1 Introduction

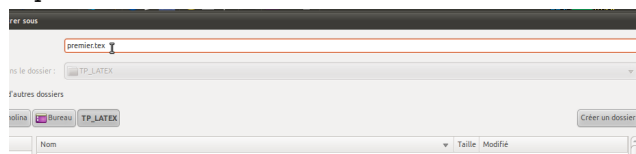
- Créons à présent un nouveau fichier (menu «Fichier» → «Nouveau») et copions le texte suivant :

```
\documentclass[12pt,a4paper]{scrartcl}
\begin{document}
Mon premier document \LaTeX.
\end{document}
```

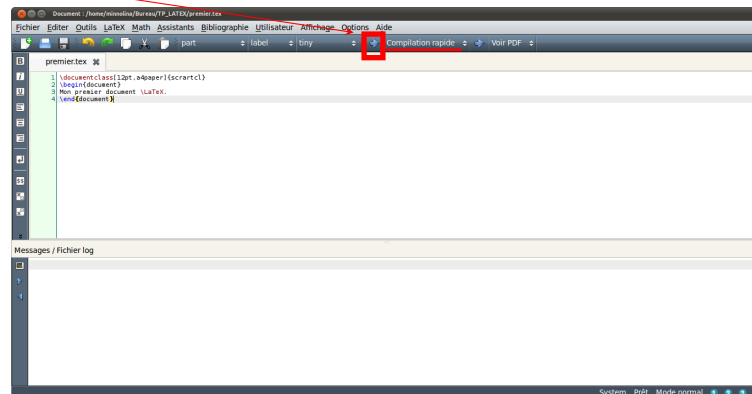
- Enregistrons ce document dans un répertoire TP_LATEX sous le nom premier.tex :
 - menu «Fichier» → «Enregistrer sous»,
 - on clique sur «Bureau» et on va créer un dossier en cliquant sur le bouton «Créer un dossier», on appelle ce dossier «TP_LATEX»



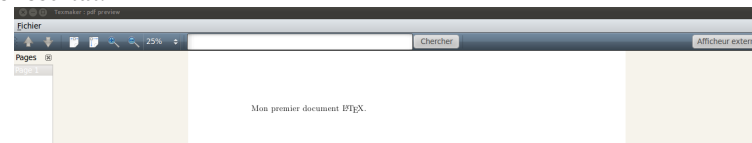
- on appelle le fichier premier.tex



- Maintenant on va compiler notre fichier avec pdf_latex et on lance la visualisation du pdf en cliquant sur la **flèche à gauche** du menu «Compilation rapide»



et on admire le résultat.



Un site de documentation pour Texmaker est disponible ici https://www.xmlmath.net/texmaker/doc_fr.html

Raccourcis clavier par défaut

Menu "Fichier" :

- Nouveau : Ctrl+N
- Enregistrer : Ctrl+S

Menu "Editer" :

- Copier : Ctrl+C
- Couper : Ctrl+X

- Coller : Ctrl+V
- Sélectionner tout : Ctrl+A
- Commenter : Ctrl+T
- Décommenter : Ctrl+U
- Chercher à nouveau : Ctrl+M
- Remplacer : Ctrl+R

Menu "Outils" :

- Quick build : F1
- Pdflatex : F6
- View Pdf : F7
- View log : F10
- Bibtex : F11

2 Premiers pas en L^AT_EX

Un fichier source L^AT_EX est un fichier `.tex` qui contient des balises. Ces éléments servent à structurer le document. Les balises sont introduites par la barre oblique inverse `\` appelée *backslash*. La plupart des balises possède un argument (parfois plusieurs). Le cas échéant, l'argument est placé entre accolades `{ }`. Les balises peuvent aussi posséder des options placées entre crochets `[]` et séparées les unes des autres par des virgules.

Les instructions L^AT_EX se divisent en deux catégories :

① les commandes

```
\nomcommande[option_1,option_2,...]{arg_1}{arg_2}
```

comme par exemple `\documentclass[]{}`, `\tableofcontents`, `\maketitle`, `\section{}`, `\LaTeX` etc. Les commandes peuvent comporter plusieurs paires d'accolades, mais ne comportent pas toujours d'options.

② les environnements

```
\begin{nom_envir}[option_1,option_2,...]
```

```
\end{nom_envir}
```

comme par exemple `\begin{document}... \end{document}`. Ici encore, les environnements ne comportent pas toujours d'options.

Les environnements peuvent être imbriqués, à condition que l'ordre de fermeture soit respecté :

```
\begin{aaa}
  \begin{bbb}
  ...
  \end{bbb}
  ...
\end{aaa}
```

Il existe une quantité inimaginable de commandes et environnements prédéfinis dans L^AT_EX (surtout après l'importation d'un package), voir par exemple <https://ctan.org/pkg>. Nous allons en voir une toute petite sélection parmi les plus utiles.

2.1 Structure de base d'un fichier L^AT_EX

Tout fichier L^AT_EX commence par la *commande*

```
\documentclass[opt_1,opt_2,...]{xxx}
```

On remarque que

- la commande commence par `\` (*backslash*) qui s'obtient avec les touches `AltGr+8`,
- son nom est `documentclass`, à savoir la classe de document,
- ses arguments optionnels `opt_1, opt_2, ...` sont entre
 - un crochet ouvrant `[` qui s'obtient avec les touches «`AltGr+5`»
 - et un crochet fermant `]` qui s'obtient avec les touches «`AltGr+`».

2 Premiers pas en \LaTeX

Les commandes placées entre crochets sont des options. Dans tout ce cours on choisira pour `documentclass` les deux options `10pt`, `a4paper` pour utiliser une police de 10 points et un rendu sur une page de format A4.

- son argument `xxx` est entre
 - une accolade ouvrante `{` qui s'obtient avec les touches «AltGr+8»
 - et une accolade fermante `}` qui s'obtient avec les touches «AltGr+=».

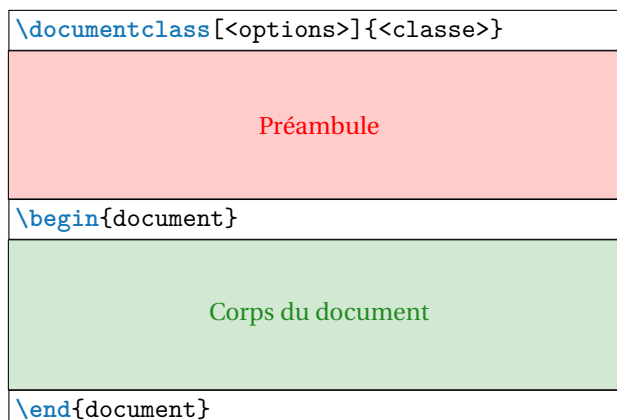
Cette instruction dit à \LaTeX que l'on souhaite utiliser une `class` de document (une mise en forme) particulière, nommée `xxx`. Dans tout ce cours on choisira `scrartcl` qui est la classe la plus courante pour des documents de quelques pages, mais d'autres choix sont possibles (`scrbook` pour des livres, `scrreprt` pour des rapports, `beamer` pour faire des présentations au vidéoprojecteur etc.).¹

L'autre composante absolument obligatoire d'un document \LaTeX est l'*environnement*

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

Ce jeu de deux commandes sert à délimiter tout ce qui sera imprimé dans le document.

La partie entre le `\documentclass` et le `\begin{document}` est appelé le **préambule**; c'est là qu'on met toutes les définitions et packages qu'on charge. La partie entre `\begin{document}` et `\end{document}` est appelé le **corps du document** (c'est ici qu'on tape le texte qu'on veut voir apparaître dans le pdf)



Voici un document \LaTeX absolument minimal et le résultat après compilation :

```
\documentclass[10pt,a4paper]{scrartcl}
\begin{document}
Texte.
\end{document}
```

Texte.

¹ \LaTeX a été écrit par un américain et est particulièrement adapté à cette langue. Pour pouvoir l'utiliser convenablement il nous faut charger des packages qui permettent de l'adapter à notre langue. C'est le cas du package `babel` et de son option `french` qui règle les problèmes linguistiques. Mais pour ce qui est de la mise en page, ce package ne règle pas le problème. En effet les classes standards (`article`, `book` et `report`) ont été conçues pour être imprimées sur le format de papier américain letter (279 × 216 mm) et legal (356 × 216 mm) et non notre format européen A4 (210 × 297 mm). C'est pour cette raison que les marges des documents \LaTeX nous paraissent toujours trop grandes. Pour adapter la mise en page de \LaTeX à nos standards européens, l'allemand Markus KOHM a développé KOMA-Script, qui est un ensemble de paquets mais surtout de classes qui remplacent les classes standard. C'est pour cela que dans ce document on utilise les classes `scrartcl`, `scrbook` et `scrreprt` au lieu des classes `article`, `book` et `report` décrites dans tous les manuels classiques.

2.2 Espaces dans le code source

Dans le fichier source, les espaces et les tabulations sont traitées indifféremment comme une «espace» par \LaTeX . Plusieurs espaces consécutives sont considérées comme une seule espace. L'espace en début d'une ligne est ignoré. Une interruption de ligne est traitée comme une espace.

Une ligne vide entre deux lignes de texte marque la fin d'un paragraphe. Plusieurs lignes vides sont considérées comme une seule ligne vide.

Il n'est pas important si on met une ou plusieurs espaces. Cette ligne appartient au même paragraphe, pas la prochaine.

Une ligne vide démarre un nouveau paragraphe.

Il n'est pas important si on met une ou plusieurs espaces. Cette ligne appartient au même paragraphe, pas la prochaine. Une ligne vide démarre un nouveau paragraphe.

2.3 Commentaires

Lorsqu'on code ou on rédige, il est utile d'annoter/désactiver certaines parties. En \LaTeX , le symbole pour-cent `%` indique le début d'un commentaire : tous les caractères entre `%` et la fin de la ligne sont ignorés par le compilateur.

Pour commenter (resp. dé-commenter) plusieurs lignes avec l'éditeur Texmaker, surligner les lignes à traiter et

pour commenter : appuyer sur les touches `Ctrl + T`,

pour dé-commenter : appuyer sur les touches `Ctrl ++ U`.

2.4 Espaces entre alinéas dans le document final

On appelle *alinéa* la portion de texte comprise entre deux retours à la ligne, appelée aussi *paragraphe*.

Le début d'une alinéa est indiquée par une indentation de la première ligne, c'est-à-dire une espace horizontale entre la marge de gauche et le premier mot du paragraphe. La raison est simple : si aucune indentation est utilisée, seule la longueur de la dernière ligne de l'alinéa précédent donnerait au lecteur un point de repère visuel. Mais dans certains cas il est très difficile de détecter si une ligne est pleine ou non. Parfois on a alors envie d'augmenter l'espace vertical entre les alinéas. Mais ce type de repère visuel présente l'inconvénient de disparaître dans certains cas. Par exemple, après une formule centrée, il serait impossible de détecter si l'alinéa précédente se poursuit ou si une nouvelle alinéa commence. De même, quand on commence à lire en haut d'une nouvelle page, il peut être nécessaire de regarder la page précédente afin de déterminer si une nouvelle alinéa a été commencée ou non. Tous ces problèmes disparaissent lorsqu'on utilise l'indentation. *Une combinaison de l'indentation et de l'espacement vertical entre les alinéas est redondante et doit donc être évitée.* L'indentation est parfaitement suffisante par elle-même. Le seul inconvénient de l'indentation est la réduction de la longueur de ligne. L'utilisation de l'espacement entre les alinéas est donc justifiée lors de l'utilisation de lignes courtes, par exemple dans un journal.

Indépendamment de l'explication ci-dessus, il y a souvent des demandes pour une mise en page de documents avec **un espacement vertical entre les alinéas au lieu de l'indentation**. La classe KOMA-Script fournit un grand nombre d'options pour gérer cet espace vertical : `parskip=full`, `parskip=full-`, `parskip=full*`, `parskip=full+` et `parskip=half`, `parskip=half-`, `parskip=half*`, `parskip=half+`. Les quatre options `full` définissent chacune un espacement vertical entre les alinéas d'une ligne. Les quatre options `half` définissent un espacement vertical d'une demi-ligne. Afin d'éviter qu'un changement d'alinéa passe inaperçu, par exemple après un saut de page, trois options (`-`, `*` et `+`) veillent à ce que la dernière ligne d'une alinéa ne soit pas complètement remplie. Si on met `-`, \LaTeX ne fera rien (la dernière ligne de l'alinéa précédente pourra être remplie) ; si on ne met rien, il laissera un espace de longueur `1em`

en fin de ligne ; si on met + il laissera un tiers de la ligne libre et enfin, si on met *, il laissera au moins un quart de la ligne libre.

⚠ Avertissement sur la personnalisation d'un document

Le but de la typographie est de rendre un document beau et agréable à lire. Pour cela, il ne faut pas que des éléments typographique détournent le lecteur du fond, ni que la mise en page n'entraîne de *fatigue visuelle*. Par exemple une règle couramment admise est que la longueur d'une ligne de texte (en typographie on dit la justification) ne doit pas dépasser 60 à 70 caractères (espace compris). Au-delà l'œil fatigue et il est plus difficile de localiser la prochaine ligne. Pour un texte écrit sur une seule colonne, une justification de 65 caractères est considérée comme idéale. C'est pour éviter la fatigue visuelle que les journaux, qui cherchent pour des raisons d'économie à mettre le plus de texte par page, sont écrit en colonnes pour conserver une justification acceptable. Donc si on décide de modifier la taille du texte, il faut essayer de respecter le plus possible cette règle.

Le *gris typographique* est l'apparence grise que prend le corps de texte, quand on le regarde de loin ou si on plisse les yeux pour voir le texte flou. Plus ce gris est homogène, plus la lecture du texte sera facile et agréable. Au contraire, quand l'homogénéité de ce gris est rompue par un mauvais interlignage, par des «rivières» ou par des «lézardes» (les lignes blanches composées d'espaces, les unes sous les autres, qui traversent les textes mal mis en page), l'attention portée au texte diminue car ces défauts détournent du texte et rendent la lecture difficile. Obtenir un bon gris n'est pas facile, il dépend de la police utilisée, de sa taille, de la justification, de l'interligne...

Avant de se lancer dans des modifications du comportement standard de \LaTeX , il faut bien comprendre que la typographie est une science difficile. Il aurait été plus facile d'écrire un logiciel laissant à l'utilisateur le choix des réglages typographiques (comme la dimension des marges par exemple). Mais les résultats sont souvent calamiteux, car les typographes improvisés que nous sommes ne savent pas ce qui est important ou incorrect, d'autant plus que nous nous sommes habitués à une typographie pauvre. Avec l'avènement des ordinateurs et des traitements de texte, on voit fleurir dans les textes tout un tas de fioritures (changement de fontes, de corps, de casse,...) censées faire beau, mais c'est oublier que la typographie est avant tout au service du texte, que le lecteur doit rester concentré, sans fatiguer.

Si j'ai tenu à évoquer ces questions d'ordre typographique, c'est pour comprendre que l'on ne s'improvise pas typographe et que **dans la plupart des cas vouloir modifier les réglages proposés par défaut par \LaTeX est une mauvaise idée**. Pour fabriquer le texte, \LaTeX utilise des algorithmes qui respectent les règles typographiques. Même si, par la suite, on verra comment régler certaines des paramètres de la mise en page, je conseille de le faire avec parcimonie et plutôt de faire confiance à \LaTeX .

2.5 Le système de packages de \LaTeX

\LaTeX dispose d'un mécanisme d'importation de packages qui permet d'étendre ses possibilités. La syntaxe pour importer un package est

```
\usepackage [option_1,option_2,...]{nom_du_package}
```

qu'il faut mettre dans le **préambule** (les options des packages utilisent des caractères non accentués). Voici le fichier source avec le **préambule** de base qu'on utilisera pour nos documents :

```
\documentclass[10pt,a4paper]{scrartcl}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{textcomp}
\usepackage{amsmath,amssymb}
\usepackage{lmodern}
\usepackage{graphicx}
\usepackage[dvipsnames,svgnames]{xcolor}
```

```

\usepackage{microtype}
\usepackage{hyperref} \hypersetup{colorlinks=true,linkcolor=Brown,urlcolor=Navy,
breaklinks=true,pdfstartview=XYZ}

\begin{document}
Test!
\end{document}

```

Regardons plus en détail chacun de ces packages.

- `\usepackage[utf8]{inputenc}`
Ce package déclare l'encodage du fichier source. Il est chargé avec l'option `utf8` pour dire à \LaTeX que le document source sera encodé en UTF8. On fait ce choix car c'est le codage par défaut des caractères sous Ubuntu, mais si vous souhaitez créer un document sur un système Windows par exemple, il faudra remplacer cette option par `latin1`.
- `\usepackage[T1]{fontenc}` Ce package, qui est chargé avec l'option `T1`, s'occupe de la gestion des accents (pour les pdf). Pour simplifier, disons que ce package permet de s'assurer que les polices de caractères qu'on utilise ont toutes les lettres accentuées dont on aura besoin (si on voulait écrire, par exemple, du polonais, il faudrait utiliser une autre option que `T1`).
- `\usepackage[french]{babel}`
La langue naturel de \LaTeX est l'anglais. Le rôle du package `babel` est de permettre à \LaTeX de parler d'autres langues, à l'aide d'options. Chaque langue possède des règles typographiques qui lui sont propre. On ne coupe pas les mots de la même manière en anglais et en français. Les items d'une listes commencent par • en anglais et — en français. En anglais et en allemand on ne met pas d'espace avant les signes doubles (; : ?!) en français on met une espace insécable avant et une espace normale après pour le : et pour les ; ?! on met avant une espace fine insécable et une espace normale après. Pour charger la prise en compte de la langue française, on utilise l'option `french`. Si on voulait écrire de l'anglais, il faudrait utiliser l'option `english`; pour du grec, l'option `greek`, etc. On peut aussi charger plusieurs langues (ce qui permet de changer au cours du document); dans ce cas, il faut mettre toutes les options voulues au package `babel`, la dernière langue écrite étant celle par défaut. Par exemple, pour un document en français avec quelques passages en anglais, on utiliserait `\usepackage[english,french]{babel}`. L'option `french` permet de respecter toutes les subtilités de la typographie française, pour la traduction automatique des mots (par exemple le mot «chapter» sera traduit automatiquement en «chapitre»), les caractères spécifiques à une langue (comme les guillemets français), etc.
- `\usepackage{textcomp}`
Il permet l'accès à certains caractères supplémentaires, non disponibles dans l'encodage `T1` utilisé par `fontenc`, comme •, ◦, ☞.
- `\usepackage{amsmath,amssymb}`
Ils sont utiles pour rédiger des formules mathématiques (on reviendra dessus au chapitre dédié aux mathématiques). `amssymb` définit nombreux symboles, `amsmath` est pratiquement indispensable. Recommandation : ~~`amsmath`~~ \rightsquigarrow `mathtools` : ce package charge `amsmath` en sous-main et en corrige quelques bogues et limitations.
- `\usepackage{lmodern}`
Ici on fait le choix d'une police de caractères (le package `fontenc` ne fait que garantir la présence des symboles, il ne choisit pas la police). À la place de `lmodern`, on peut utiliser par exemple `fourier` ou `pxfonts` (Palatino) ou encore `txfonts` (Times), `kpfonts` etc.
- `\usepackage{graphicx}`
Il permet d'inclure des images dans le pdf. Recommandation : ~~`graphicx`~~ \rightsquigarrow `graphbox` : ce package charge `graphicx` en sous-main et ajoute des options pour l'alignement vertical (utile pour des images sur la même ligne).

- `\usepackage[dvipsnames,svgnames]{xcolor}`
Il permet d'utiliser les couleurs (voir la documentation <https://www.ctan.org/pkg/xcolor>).
- `\usepackage{microtype}`
Il introduit des améliorations typographiques (toujours charger `microtype` **après** un package changeant de fonte comme `lmodern` ou `fourier`).
- `\usepackage{hyperref}`
Le package `hyperref` (toujours à mettre **en dernier** même si on rajoute des packages supplémentaires), permet de gérer les liens, construit les bookmarks du pdf et rend la table des matières interactive (les entrées de la table des matières sont entourées en rouge et lorsqu'on clique dessus, on est amené à la section en question; le cadre rouge ne s'imprime pas, il est seulement visible à l'écran; avec l'option `colorlinks=true`, `linkcolor=Brown` on n'a plus de cadre mais les entrées sont écrites en Brown). On termine par dire à `hyperref` qu'on autorise les coupures des URL et d'utiliser le zoom standard (par défaut, `hyperref` change le zoom à quelque chose de trop petit pour être lu) : `\hypersetup{pdfstartview=XYZ}`



2.5.1 Pour aller plus loin

Il existe un grand nombre de packages \LaTeX sur CTAN pour différentes disciplines et exigences. La liste complète des packages disponibles ainsi que leur manuel d'utilisation est disponible à l'adresse

<https://ctan.org/>

Cependant, la plupart de ces packages ne doivent pas être installés directement et manuellement à partir de CTAN. Au lieu de cela, vous devez utiliser le gestionnaire de packages de la distribution \LaTeX que vous utilisez pour l'installation.

Avec `TeXLive` (la distribution classique sous Ubuntu), pour accéder rapidement à la documentation d'un package on peut taper dans un terminal (qu'on peut ouvrir avec la combinaison de touches «`ctrl+alt+T`»)

```
texdoc nomDuPackage
```

Par exemple, pour obtenir la documentation de `biblatex` on tapera

```
texdoc biblatex
```

2.6 Caractères spéciaux

Comme nous l'avons vu précédemment, les commentaires sont précédés du symbole `%`. Ainsi, pour écrire ce symbole, il faudra utiliser une syntaxe indirecte, comme pour d'autres symboles dont voici une liste (non exhaustive) :

- le *backslash* `\` pour commencer une commande. Pour afficher le *backslash* on utilise la commande `\textbackslash`.
- les accolades `{ et }` pour définir des groupes. Pour afficher les accolades on utilise les commandes `\{ et \}`.
- le pour-cent `%` pour commencer un commentaire (lors de la compilation du document, les caractères situés après ce symbole seront ignorés). Pour afficher le pour-cent on utilise la commande `\%`.
- un ou plusieurs dollars `$` pour passer en mode mathématique. Pour afficher un dollar on utilise la commande `\$`.
- l'*underscore* `_` pour écrire des indices en mode mathématique. Pour afficher l'*underscore* on utilise la commande `_`.
- le chapeau `^` pour écrire des exposants en mode mathématique. Pour afficher le chapeau on utilise la commande `\textasciicircum`.

- l'*ampersand* `&` pour séparer les colonnes d'un tableau. Pour afficher l'*ampersand* on utilise la commande `\&`.
- le *hash* `#` pour définir des nouvelles commandes. Pour afficher le hash on utilise la commande `\#`.
- le tilde `~` pour produire une espace insécable.² Pour afficher le tilde on utilise la commande `\textasciitilde`.

2.7 Titre, table des matières et sections

Nous allons maintenant fabriquer notre premier vrai document. Le but est d'arriver à produire le code source correspondant au pdf donné à la figure 2.1. Analysons la *structure* de ce document. Il comprend tout d'abord un titre, un auteur et une date. Ensuite, il y a une table des matières, puis le corps du document proprement dit. Ce corps de document comprend une section non numérotée (et qui n'est pas dans la table des matières) dont le titre est «Introduction», du texte, puis une section numérotée dont le titre est «Rappels», du texte, une sous-section numérotée dont le titre est «Condition initiale», du texte, une sous-section numérotée dont le titre est «Problème de Cauchy» puis du texte et pour finir une section numérotée dont le titre est «Exercices».

Une chose importante à comprendre est que \LaTeX est un langage orienté vers la *sémantique* : au lieu de dire qu'on va mettre la phrase «Équations différentielles» en gros caractères et centré, on va dire à \LaTeX que cette phrase est le titre, puis lui dire de l'afficher. Si ensuite on décide que le titre doit être en gras, il suffira de personnaliser l'apparence du titre.

Voici les commandes nécessaires pour générer ces éléments.

Dans le **préambule** :

- Les trois premières commandes, `\title`, `\author` et `\date` sont des commandes à un argument qu'on mettra dans le **préambule**. Voici un exemple d'utilisation :

```
\title{Équations différentielles}
\author{Gloria Faccanoni}
\date{15 janvier 2014}
```

Si on ne met pas la ligne avec `\date` (par exemple on met un % devant l'instruction), la date choisie sera la date du jour de compilation du fichier.

Si on commente la ligne qui importe le paquet `babel`, la date sera en anglais.

- Pour afficher le texte de l'exemple sans devoir le taper on peut charger dans le **préambule** le package `lipsum` et puis utiliser la commande `\lipsum`. Cette commande génère par défaut les paragraphes 1 à 7 du `Lipsum`.³ Pour générer un `Lipsum` contenant les 2 premiers paragraphes, on écrira `\lipsum[1-2]`. Pour afficher la première phrase du deuxième paragraphe, on écrira `\lipsum[2][1]`. Le paquet `blindtext` propose un résultat similaire.

Dans le **corps du document** :

- Après le `\begin{document}` (c'est-à-dire dans le **corps du document**), on utilise `\maketitle` pour imprimer le titre, l'auteur et la date (cette commande ne prend pas d'argument, contrairement aux trois précédentes). Rappelons que ces informations ont été mentionnées dans le préambule. Si la commande `\maketitle` est omise, le document obtenu après la compilation n'inclura pas ces informations.

²L'espace insécable est une espace qui ne peut être coupée par un saut de ligne. Elle s'écrit `~` en \LaTeX (`\`, pour l'espace fine insécable).

³Le *Lorem Ipsum* est simplement du faux texte employé dans la composition et la mise en page avant impression. C'est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du *Lorem Ipsum* sur un texte générique comme «bla bla bla» est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du *Lorem Ipsum* leur faux texte par défaut, et une recherche pour «Lorem Ipsum» vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction.

- Lors d'une publication scientifique (notamment des articles), il est usuel de démarrer celle-ci avec un résumé (*abstract* en anglais), censé donner au lecteur une vue d'ensemble de ce qu'il doit attendre du document. La classe `scrartcl` fournit un environnement `abstract` à cette fin :

```
\begin{abstract}
...
\end{abstract}
```

- Pour afficher la table des matières, on utilise la commande

```
\tableofcontents
```

Comme `\maketitle`, cette commande ne prend pas d'argument. \LaTeX permet de générer automatiquement la table des matières. Pour que la table soit générée, **il faut compiler le document deux fois**. En effet, la première compilation créera un fichier (dont l'extension est `.toc`) dans lequel seront mises les diverses informations à insérer dans le sommaire) et la seconde compilation lira le contenu de ce fichier.

- Pour faire une section numérotée, on utilise la commande `\section` dont l'argument est le titre de section; par exemple :

```
\section{Titre de la section}
```

Si on ne veut pas que la section soit numérotée, on utilise `\section*`; par exemple :

```
\section*{Titre de la section}
```

Les sections non numérotées n'apparaissent pas dans la table des matières. Pour les ajouter, au lieu de la commande `\section*` on utilisera la commande

```
\addsec{Titre de la section}
```

Pour faire une sous-section, on utilise `\subsection`. La syntaxe est la même que pour la commande `\section`. Dans la classe `scrartcl`, les commandes de sectionnement suivantes sont disponibles :

```
\part et \part* et \addpart
\section et \section* et \addsec
\subsection et \subsection*
\subsubsection et \subsubsection*
\minisec
\paragraph
\subparagraph
```

Dans les classes `scrreprt` et `scrbook`, une commande de sectionnement entre `\part` et `\section` est disponible :

```
\chapter et \chapter* et \addchap
```

Attention : les commandes `\addpart`, `\addchap`, `\addsec` et `\minisec` ne sont disponibles qu'avec les classes `scrartcl`, `scrbook`, `scrreport`)

Exercice 1

Reproduire le document de la figure 2.1.

2.8 Inclusion de fichiers \LaTeX

Quand un fichier `.tex` commence à devenir long, il est souvent utile de pouvoir en compiler une seule partie (par exemple juste un chapitre ou une section). Il est donc intéressant de séparer le fichier source en plusieurs fichiers.

Deux commandes permettent d'effectuer ce travail : les commandes `input` et `include`. Elles prennent un unique paramètre qui est le nom du fichier à inclure. Attention : avec `input` il faut ajouter l'extension

Équations différentielles

Gloria Faccanoni

15 janvier 2014

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Table des matières

1	Rappels	2
1.1	Condition initiale	2
1.2	Problème de Cauchy	2
2	Exercices	2

Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1

(a) Page 1

FIGURE 2.1 – Exemple de document de deux pages avec titre, table des matières et sections

1 Rappels

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1.1 Condition initiale

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

1.2 Problème de Cauchy

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

2 Exercices

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

FIGURE 2.1 – Exemple de document de deux pages avec titre, table des matières et sections

dans le nom du fichier à importer, avec `include` il ne faut pas indiquer l'extension. De plus, la commande `include` démarre une nouvelle page avant d'inclure le contenu du fichier.

Fichier ch1.tex

```
\section{Mon chapitre}
Bla bla
```

Fichier ch2.tex

```
\section{Un chapitre}
Bli bli
```

Fichier principale à compiler

```
\documentclass{scrartcl}

\begin{document}

\input{ch1.tex} % ou \include{ch1} NB sans extension
%\input{ch2.tex} non importé car ligne commentée

\end{document}
```

Le fichier `ch1.tex` n'est absolument pas un fichier \LaTeX complet. Lorsque \LaTeX rencontre la commande `input`, il va ouvrir le fichier `ch1.tex` et simplement copier-coller son contenu à la place de `\input{ch1}` comme si on avait écrit le document en un seul fichier comme ci-dessous :

Fichier unique

```
\documentclass{scrartcl}

\begin{document}

\section{Mon premier chapitre}
Bla bla

\end{document}
```

Pour aller plus loin on pourra utiliser (après en avoir lu la documentation) le package `subfiles` disponible à l'adresse <https://ctan.org/pkg/subfiles>



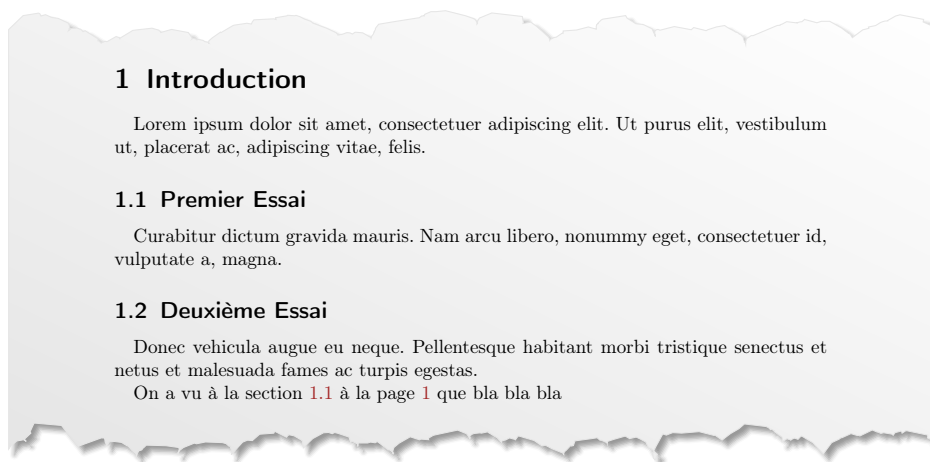
2.9 Étiquettes, références croisées et liens externes

Il est parfois utile dans un document de faire référence à des parties du texte (une section, une équation, un théorème, une entrée dans la bibliographie etc.). Plutôt que d'indiquer à la main les pages concernées (opérations hasardeuse par ailleurs, car le numéro de page peut changer en cours de rédaction), il est plus simple de laisser \LaTeX gérer ces opérations. Ainsi, on dispose des trois commandes suivantes :

- la commande `\label{eti}` insère une étiquette invisible, nommée `eti` ;
- `\ref{eti}` affiche la référence à l'étiquette `eti` ;
- `\pageref{eti}` insère le numéro de la page où se situe l'étiquette `eti`.

Comme tout ce qui est dynamique, il faut deux compilations pour que cela s'affiche correctement. Si on a chargé dans le préambule le package `hyperref`, la référence est cliquable et renvoi à la page où on a mis l'étiquette invisible. Pour que la référence soit correcte il faut compiler **deux** fois.

Voici un exemple



obtenu avec le code suivant :

```
\documentclass[a4paper,10pt]{scrartcl}

\usepackage[latin1]{inputenc} % sous Windows
% \usepackage[utf8]{inputenc} % sous Linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage[dvipsnames,svgnames]{xcolor}
\usepackage{hyperref}\hypersetup{colorlinks=true,linkcolor=Brown,citecolor=ForestGreen}

\usepackage{lipsum}

\begin{document}
\section{Introduction}
\lipsum[1][1-2]

\subsection{Premier Essai}\label{subsec.toto}
\lipsum[1][3-4]

\subsection{Deuxième Essai}
\lipsum[1][5-6]

On a vu à la section-\ref{subsec.toto} à la page-\pageref{subsec.toto} que bla bla bla
\end{document}
```

Pour imprimer les étiquettes dans le pdf (pendant la rédaction), il suffit d'importer dans le **préambule** le paquet showkeys : `\usepackage{showkeys}`. Ne pas oublier de commenter l'importation lors de la phase finale.

Pour afficher un lien internet (cliquable pour aller à la page web voulue), on utilise la commande `\url{}` ou la commande `\href{}{}` du package hyperref :

```
\url{http://www.google.com}
```

```
\href{http://www.google.com}{Google}
```

```
http://www.google.com
Google
```

2.10 Notes de bas de page ou dans la marge

La commande `\footnote` imprime une note de bas de page en bas de la page en cours. Les notes de bas de page doivent être placées après le mot où la phrase auquel elles se réfèrent.⁴ Les notes qui se réfèrent à une (partie de) phrase devraient être placées après une virgule ou un point.⁵

L'alinéa précédente a été obtenue par le code

```
... elles se réfèrent.\footnote{La typographie française demande ... du document.} Les
notes qui se réfèrent à une (partie de) phrase devraient être placées après une virgule
ou un point.\footnote{Remarquez que les notes de bas de page détournent l'attention du
lecteur du corps du document.} La note ...
```

La commande `\marginpar` imprime une note dans la marge de la page en cours. Cette phrase a été obtenue par le code

Exemple de note dans la marge

```
La commande \marginpar imprime une note dans la marge\marginpar{Exemple de note dans la
marge} de la page en cours.
```

2.11 Listes à puce, énumérations et descriptions

Une manière de structurer le texte à l'intérieur d'un paragraphe est d'utiliser les listes.

- L'environnement `itemize` est utilisé pour des listes à puces :

```
Liste des courses:
\begin{itemize}
\item pain
\item lait
\item farine
\end{itemize}
```

```
Liste des courses :
— pain
— lait
— farine
```

Pour redéfinir une seule label d'une listes à puce, on pourra écrire :

```
Liste des courses:
\begin{itemize}
\item pain
\item[ $\bullet$ ] lait
\item farine
\end{itemize}
```

```
Liste des courses :
— pain
• lait
— farine
```

Pour restituer aux marqueurs des listes `itemize` les valeurs standard attribuées par la classe de document ou les extensions utilisées, on pourra écrire dans le **préambule** juste après le `\usepackage` `[french]{babel}` :⁶

```
\frenchsetup{StandardItemLabels=true}
```

On peut sinon choisir le marqueur :

```
\frenchsetup{ItemLabeli=\textbullet}
\frenchsetup{ItemLabelii=\textopenbullet}
\frenchsetup{ItemLabeliii=\textborn}
\frenchsetup{ItemLabeliv=\textleaf}
```

- L'environnement `enumerate` est utilisé pour des énumérations :

⁴La typographie française demande une espace fine avant la marque de renvoi à la note. Celle-ci est insérée automatiquement par le package `babel` si le français est la langue principale du document.

⁵Remarquez que les notes de bas de page détournent l'attention du lecteur du corps du document.

⁶<https://ctan.org/pkg/babel-french>

```
Risotto:
\begin{enumerate}
\item faire ruisser l'oignon et le
beurre,
\item ajouter du vin blanc et le riz et
laisser évaporer,
\item ajouter le bouillon peu à peu,
\item \dots
\end{enumerate}
```

Risotto :

1. faire ruisser l'oignon et le beurre,
2. ajouter du vin blanc et le riz et laisser évaporer,
3. ajouter le bouillon peu à peu,
4. ...

- L'environnement `description` est utilisé pour des descriptions :

```
ECUE Compétences:
\begin{description}
\item[Titre:] Édition scientifique
avec \LaTeX
\item[Durée:] 12h
\end{description}
```

ECUE Compétences :

Titre : Édition scientifique avec \LaTeX
Durée : 12h

On peut imbriquer différents types de listes :

```
\begin{itemize}
\item À faire:
\begin{enumerate}
\item Sortir le chien
\item Rentrer les poubelles
\item Préparer à manger
\end{enumerate}
\item Ce qui sera fait:
\begin{enumerate}
\item Prendre une bière
\item Allumer la TV
\item Dormir
\end{enumerate}
\end{itemize}
```

— À faire :

1. Sortir le chien
2. Rentrer les poubelles
3. Préparer à manger

— Ce qui sera fait :

1. Prendre une bière
2. Allumer la TV
3. Dormir

Exercice 2

Reproduire la liste suivante :

1. Créatures fantastiques
 - a) Dotées d'ailes
 - i. Volent haut
 - Dragons
 - Phénix
 - ii. Volent bas
 - Fées
 - Papillons géants
 - b) Sans ailes
 - i. Terrestres
 - Gobelins
 - Trolls
 - ii. Aquatiques
 - Sirènes

— Krakens

2. Créatures inclassables

Territoire inconnu : Cheshire Cat, licornes multicolores**Dimension parallèle** : Griffons, chimères

Pour personnaliser les listes on pourra utiliser (après en avoir lu la documentation) :

- soit le package `enumitem` disponible à l'adresse <https://ctan.org/pkg/enumitem>
- soit le package `paralist` disponible à l'adresse <https://ctan.org/pkg/paralist>

 **Exercice 3**

Reprendre l'exercice 2 en ajoutant dans le préambule `\usepackage[pointedenum]{paralist}`. Qu'est-ce qui change ?

2.12 Afficher des codes sources avec listings

Pour insérer du code source dans un document \LaTeX , il est possible d'utiliser le package `listings`. On va également pouvoir préciser le langage qui doit être utilisé pour la coloration ou encore la coloration du fond.

Le package `listings` propose la commande `\lstinline` pour des codes sources dans le texte, l'environnement `\lstlisting` pour imprimer plusieurs lignes de code et une commande `\lstinputlisting` qui permet d'inclure un fichier externe.

Dans le préambule on écrira par exemple

```
\usepackage{listings}
\lstset{
  upquote = true,
  columns = flexible,
  basicstyle = \ttfamily,
  keepspaces=true,
  language = python,
  backgroundcolor = \color{Navy!10},
  keywordstyle = \color{ForestGreen}\bfseries,
  % numbers=left, numberstyle=\tiny,
  tabsize = 4,
}
```

Puis dans le corps du document on écrira soit

```
la fonction \lstinline!print(! ...
```

soit

```
\begin{lstlisting}
impaires = lambda L,R : [i for i in range(L,R+1) if i%2!=0]
print(impaires(1,15))
\end{lstlisting}
```

soit

```
\lstinputlisting{source.py}
```

Cette dernière instruction inclut directement le fichier source `.py`. Voici un exemple de résultat après importation d'un fichier python :

2 Premiers pas en L^AT_EX

```
impaires = lambda L,R : [i for i in range(L,R+1) if i%2!=0]
print(impaires(1,15))
```

Comme d'habitude, pour plus d'informations il faut lire la documentation officielle :

<https://ctan.org/pkg/listings>

Voir aussi

<https://borntocode.fr/latex-comment-inserer-et-customiser-du-code-source/>

Problème : lorsqu'on utilise l'encodage UTF-8, l'utilisation de lettres accentuées dans les codes sources pose des problèmes. Nous avons alors trois options : abandonner l'UTF-8 dans l'ensemble du document, renoncer aux caractères accentués dans nos codes, ou adopter une stratégie intermédiaire qui permettra de résoudre ce problème pour les imports via `lstinputlisting`. Cette solution implique le remplacement du package `listings` par le package `listingsutf8`, qui est essentiellement un patch pour le package `listings`.

<https://ctan.org/pkg/listingsutf8>

```
\usepackage{listingsutf8}
\lstset{
  upquote = true,
  columns = flexible,
  keepspaces=true,
  basicstyle = \ttfamily,
  language = python,
  backgroundcolor = \color{Navy!10},
  keywordstyle = \color{ForestGreen}\bfseries,
  % numbers=left, numberstyle=\tiny,
  tabsize = 4,
  % NEW
  inputencoding = utf8, % latin1
  extendedchars=false,
  literate = {é}{\`e}{1}%
             {è}{\`e}{1}%
             {â}{\`a}{1}%
             {ã}{\`a}{1}%
             {ç}{\`c}{1}%
             {¼}{\`oe}{1}%
             {û}{\`u}{1}%
             {É}{\`E}{1}%
             {Ê}{\`E}{1}%
             {À}{\`A}{1}%
             {Ç}{\`C}{1}%
             {¼}{\`OE}{1}%
             {Ê}{\`E}{1}%
             {ê}{\`e}{1}%
             {î}{\`i}{1}%
             {ï}{\`i}{1}%
             {ô}{\`o}{1}%
             {û}{\`u}{1}%
}
```

`inputencoding` spécifie l'encodage; `literate` définit les substitution de caractères. Toutes ces substitutions sont de la forme {caractère dans le listing}{ce par quoi on doit le remplacer}{nombre d'espaces à droite et à gauche de la substitution}.

2.13 Objets flottants : figures et tableaux

La plupart des publications contiennent un nombre important de figures et de tableaux. Ces éléments nécessitent un traitement particulier car ils ne peuvent être coupés par un changement de page. On pourrait imaginer de commencer une nouvelle page chaque fois qu'une figure ou un tableau ne rentrerait pas dans la page en cours. Cette façon de faire laisserait de nombreuses pages à moitié blanches, ce qui ne serait réellement pas beau. La solution est de laisser «flotter» les figures et les tableaux qui ne rentrent pas sur la page en cours, vers une page suivante et de compléter la page avec le texte qui suit l'objet «flottant». \LaTeX fournit deux environnements pour les *objets flottants* adaptés respectivement aux figures (`figure`) et aux tableaux (`table`). Pour faire le meilleur usage de ces deux environnements, il est important de comprendre comment \LaTeX traite ces objets flottants de manière interne. Dans le cas contraire ces objets deviendront une cause de frustration intense car \LaTeX ne les placera jamais à l'endroit où vous souhaitez les voir.

☞ Canevas des objets flottants

<pre> \begin{figure} \centering \includegraphics{myfig} \caption{Titre de la figure} \label{fig.myfig} \end{figure} ... on a montré à la figure~\ref{fig. myfig} que ... </pre>	<pre> \begin{table} \centering \begin{tabular}{...} ... \end{tabular} \caption{Titre du tableau} \label{tab.montableau} \end{table} ... les données de la table~\ref{tab. montableau} ... </pre>
--	---

Tout objet inclus dans un environnement `figure` ou `table` est traité comme un objet flottant. Les deux environnements flottants ont un paramètre optionnel (voir le tableau 2.1) : `\begin{figure} [placement]` ou `\begin{table} [placement]`. Ce paramètre permet de dire à \LaTeX où vous préférez positionner l'objet à flotter si c'est possible. En général, si un objet ne peut pas être placé sur la page en cours, il est placé soit dans la file des figures soit dans la file des tableaux⁷. Quand une nouvelle page est entamée, \LaTeX essaye d'abord de voir si les objets en tête des deux files pourraient être placés sur une page spéciale, à part. Si cela n'est pas possible, les objets en tête des deux files sont traités comme s'ils venaient d'être trouvés dans le texte : il essaye de les placer selon leurs spécifications de placement (sauf `h`, qui n'est plus possible). Tous les nouveaux objets flottants rencontrés dans la suite du texte sont ajoutés à la queue des files. \LaTeX respecte scrupuleusement l'ordre d'apparition des objets flottants. C'est pourquoi un objet flottant qui ne peut être placé dans le texte repousse tous les autres à la fin du document. D'où la règle :

«Si \LaTeX ne place pas les objets flottants comme vous le souhaitez, c'est souvent à cause d'un seul objet trop grand qui bouche l'une des deux files d'objets flottants.»

Essayer d'imposer à \LaTeX un emplacement particulier pose souvent problème : si l'objet flottant ne tient pas à l'emplacement demandé, alors il est coincé et bloque le reste des objets flottants. En particulier, l'utilisation de la seule option `[h]` pour un flottant est une idée à *proscrire*, les versions modernes de \LaTeX changent d'ailleurs automatiquement l'option `[h]` en `[ht]`.

Voici quelques éléments supplémentaires qu'il est bon de connaître sur les environnements flottants.

- La commande `\caption` permet de définir une légende pour l'objet. Un numéro (incrémenté automatiquement) et le mot «Figure» ou «Table» (traduits automatiquement en français par le package `babel`) sont ajoutés par \LaTeX .
- Avec `\label` et `\ref` vous pouvez faire référence à votre objet à l'intérieur de votre texte. La commande `label` doit apparaître *après* la commande `\caption` d'une légende si vous voulez référencer le numéro de cette légende.

⁷ Il s'agit de files FIFO (*First In, First Out*) : premier arrivé, premier servi.

Caractère	Emplacement pour l'objet flottant...
h	<i>here</i> , ici, à l'emplacement dans le texte où la commande se trouve. Utile pour les petits objets.
t	<i>top</i> , en haut d'une page
b	<i>bottom</i> , en bas d'une page
p	<i>page</i> , sur une page à part ne contenant que des objets flottants.
!	ici, sans prendre en compte les paramètres internes (tels que le nombre maximum d'objets flottants sur une page) qui pourraient empêcher ce placement.

TABLE 2.1 – Placements possibles

- Les deux commandes `\listoffigures` et `\listoftables` fonctionnent de la même manière que la commande `\tableofcontents` : elles impriment respectivement la liste des figures et des tableaux. Dans ces listes, la légende est reprise en entier. Si vous désirez utiliser des légendes longues, vous pouvez en donner une version courte entre crochets qui sera utilisée pour la table : `\caption[courte]{Longue sous la figure}`

2.13.1 Insérer des images, les encapsuler dans une figure

L'importation d'une image dans un document se fait en deux étapes : d'abord on indique à \LaTeX le nom du fichier qui contient l'image, puis on la transforme en un objet flottant (appelé *figure*) pour que \LaTeX puisse la placer dans la page avec le meilleur rendu possible. Cela signifie que, si l'image est trop grande pour le peu de place qui reste dans la page, il la placera sur la page d'après. Il ne faut pas forcer le placement des images car \LaTeX est plus performant que vous pour l'arrangement des pages, donc il faut le laisser faire son travail.

Insérer une image : les formats d'images acceptés par une compilation avec `pdflatex` sont JPG, PNG et pdf (le dernier est à préférer pour tout ce qui est vectoriel). Pour inclure une image appelée `courbe.png`, présente dans le même répertoire que le fichier `tex` que vous compilez (très important), il faut charger le package `graphicx` en mettant dans le **préambule**⁸

```
\usepackage{graphicx}
```

puis mettre dans le **corps du document** là où on veut faire apparaître l'image la commande

```
\includegraphics{courbe}
```

Noter qu'il n'y a pas besoin de spécifier l'extension.

La commande `\includegraphics` peut prendre un certain nombre d'options. L'option `width` permet de spécifier la largeur de l'image. Par exemple

```
\includegraphics[width=0.5\textwidth]{courbe}
```

mettra la largeur de l'image à la moitié de la largeur du texte, la hauteur étant changée en conséquence pour garder les proportions de l'image. De même, on peut spécifier la hauteur⁹

```
\includegraphics[height=2cm]{courbe}
```

On peut changer l'orientation de l'image avec l'option `angle`. Par exemple

```
\includegraphics[angle=90]{courbe}
```

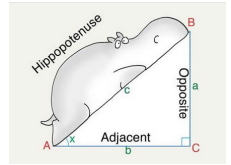
tourne l'image de 90 degrés (sens trigonométrique). On peut couper les bords de l'image avec les options `trim` et `clip`. Par exemple

⁸Recommandation : remplacer le package `graphicx` par le nouveau package `graphbox` : ce package charge `graphicx` en sous-main et ajoute des options pour l'alignement (utile pour aligner des images horizontalement par exemple).

⁹On peut aussi changer la taille de l'image avec l'option `scale`. Par exemple `\includegraphics[scale=0.4]{courbe}` réduira la taille de l'image à 40% de celle de l'original. Cependant il est préférable d'utiliser des dimensions relatives à la page plutôt qu'à l'image.


```
\includegraphics[trim=1cm 2cm 3cm 4cm,clip]{courbe}
```

affichera l'image coupée de 1 cm à partir de la marge gauche, 2 cm de la marge en bas, 3 cm de la marge droite et 4 cm de la marge haute.¹⁰ Pour une description des autres options, voir la documentation du package `graphicx` disponible à l'adresse <https://www.ctan.org/pkg/grfguide>



Exercice 4

Placer une image dans la marge comme ci-contre (la largeur sera fixée à `\marginparwidth`).

Placer l'image dans une figure flottante : il est conseillé de laisser \LaTeX s'occuper automatiquement du placement d'une image grâce à l'environnement `{figure}`. À l'intérieur de cet environnement, il faut mettre la commande `\includegraphics` et rajouter une légende afin de pouvoir faire référence à la figure. Pour cela, on utilise la commande `\caption` ainsi que la commande `\label` pour y faire référence plus tard avec la commande `\ref` (comme pour les numéros de sections, d'équations, etc.). La syntaxe est donc

```
\begin{figure}
\centering
\includegraphics{myfig}
\caption{Titre de la figure}\label{fig.myfig}
\end{figure}
```

Toujours mettre le `\label` après le `\caption`. On peut encourager \LaTeX à positionner l'image à un certain endroit en utilisant un argument optionnel de l'environnement `{figure}` (par exemple `\begin{figure}[t]`). Le paramètre `t` favorisera l'apparition en haut de page, le paramètre `b` en bas de page, le paramètre `h` à l'endroit où est `{figure}` dans le fichier source et `p` sur une page à part avec d'autres images.

La figure 2.2 a été obtenue par le code suivant :

```
\lipsum[1]
On voit à la figure~\ref{fig.hippopotense} que bla bla

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{hippopotense}
\caption{I put the hippo in hippopotense}\label{fig.hippopotense}
\end{figure}

\lipsum[2]
```

Remarquer que dans le code on insère l'image après le premier paragraphe mais \LaTeX a préféré la faire apparaître en haut de la page. Cela n'est pas gênant car il suffit de faire référence à l'image non pas par une expression du type «comme on voit sur la FIGURE ci-dessous» mais plutôt par une expression du type «comme on voit sur la FIGURE 1» où le numéro est géré automatiquement par un mécanisme d'étiquette/référence.

Le comportement par défaut des images est de flotter, afin que \LaTeX puisse trouver la meilleure façon de les disposer dans votre document et d'en améliorer l'apparence. Si vous jetez un coup d'oeil, vous verrez que c'est ainsi que les livres sont souvent mis en page. La meilleure chose à faire est donc de laisser \LaTeX faire son travail et de ne pas essayer de forcer le placement des figures à des endroits spécifiques. Cela signifie également que vous devez éviter d'utiliser des phrases telles que «dans la figure suivante» qui exige que la figure soit placée à un endroit précis, et utiliser plutôt «dans la figure~\ref{...}», en tirant parti des références croisées de \LaTeX .

Dans certains cas il peut s'avérer nécessaire d'utiliser la commande `\FloatBarrier` du package `placeins` : elle ordonne à \LaTeX de placer tous les objets flottants avant cette commande (par exemple,

¹⁰Pour choisir les marges à rogner, en hase de rédaction on pourra utiliser `\fbox{\includegraphics[trim=1cm 2cm 3cm 4cm]{courbe}} % sans clip`

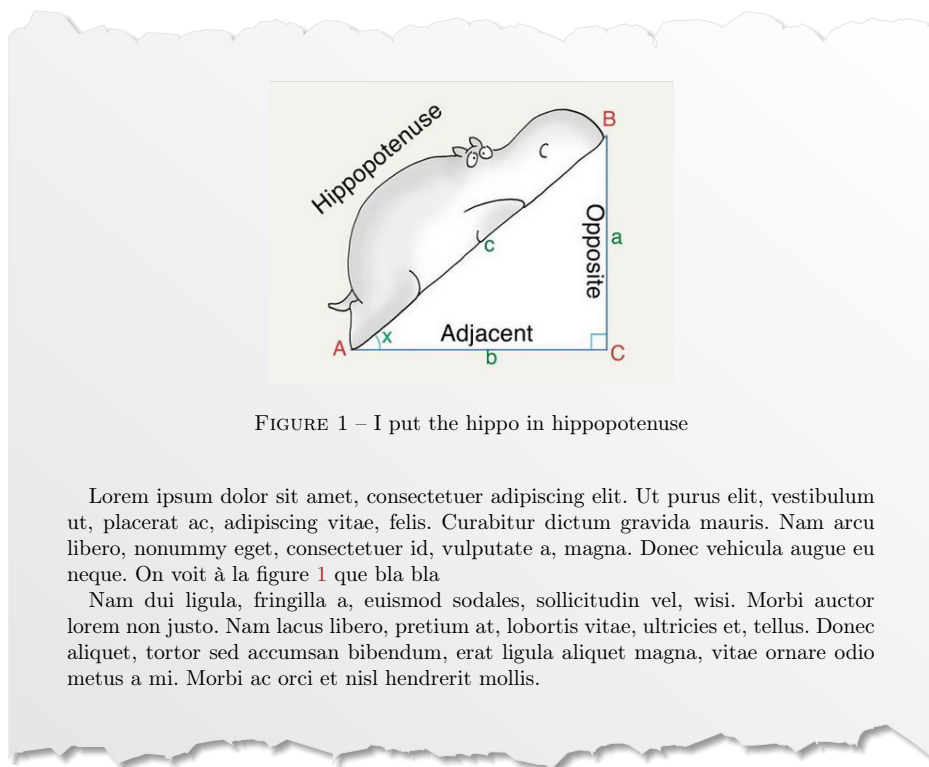


FIGURE 2.2 – Exemple de figure avec référence.

avant de commencer une nouvelle section). Ça reste un objet flottant mais on empêche \LaTeX de le placer au delà de cette commande.

Si, pour une raison quelconque, vous voulez vraiment qu'une figure particulière soit placée "ICI", et non là où \LaTeX veut la placer, vous pouvez utiliser l'option [H] du package float qui transforme la figure flottante en une figure non flottante normale. Notez cependant que dans ce cas vous ne voulez pas forcément ajouter de légende à votre figure, vous n'avez alors pas besoin d'utiliser l'environnement figure du tout! Vous pouvez utiliser juste la commande `\includegraphics` n'importe où dans votre document pour insérer une image.

2.13.2 Insérer des tableaux, les encapsuler dans une table

Le rendu d'un tableau dans un document se fait aussi en deux étapes : d'abord on construit le tableau, puis on le transforme en un objet flottant (appelé table) pour que \LaTeX puisse le placer dans la page avec le meilleur rendu possible, selon le même mécanisme déjà vu pour les images/figures.

Construire un tableau : pour écrire un tableau, on utilise l'environnement `{tabular}`. Cet environnement prend un argument obligatoire qui est la spécification du type des colonnes. Les types de colonnes de base sont c (centré), l (aligné à gauche), r (aligné à droite) et p{5cm} (paragraphe de 5cm de large). Par exemple, pour obtenir un tableau avec trois colonnes, la première étant centrée, la deuxième un paragraphe de 7 cm et la dernière alignée à droite, on écrit

```
\begin{tabular}{cp{7cm}r}
...
\end{tabular}
```

Le texte des différentes colonnes est séparé par un `&` et on passe à la ligne suivante en utilisant `\\`. Par exemple :

Voici un tableau :

```
\begin{tabular}{cp{1cm}r}
chat chien      & lapin poulet & cochon
\\
oiseau poule    & oie vache    & écureuil
\end{tabular}
```

Voici un tableau :

chat chien	lapin	cochon
	poulet	
oiseau poule	oie	écureuil
	vache	

La commande `\\` doit uniquement être utilisée pour passer à la ligne dans les tableaux. Elle ne doit pas être utilisée pour aller à la ligne lorsqu'on tape du texte (il faut laisser une ligne blanche dans le fichier source). L'utilisation abusive de `\\` dans un document que vous me rendrez sera sévèrement sanctionné. (Noter que le tableau est la suite du texte ; pour éviter ceci, le mettre dans un environnement `{center}` ou le rendre un flottant).

Pour rajouter un trait vertical entre deux colonnes, il suffit de mettre un `|` (qu'on obtient avec la combinaison de touches «AltGr+6») dans la spécification des colonnes. Pour tracer un trait horizontal, il y a la commande `\hline`. Cette commande doit être soit au tout début du tableau soit après un `\\`. Voici un exemple :

```
\begin{center}
\begin{tabular}{|c|p{1cm}r|}
\hline
chat chien      & lapin poulet & cochon
\\
\hline
oiseau poule    & oie vache    & écureuil
\\
\hline
\end{tabular}
\end{center}
```

chat chien	lapin	cochon
	poulet	
oiseau poule	oie	écureuil
	vache	

Lorsqu'on fait des tableaux, toujours charger le package `array` (dans le préambule et, comme toujours, avant `hyperref`), qui corrige certains petits problèmes concernant les raccords entre traits horizontaux et verticaux et étend les possibilités pour les tableaux.

\LaTeX traite le contenu d'un environnement `tabular` comme une boîte indivisible, en particulier il ne peut y avoir de coupure de page. Pour réaliser de longs tableaux s'étendant sur plusieurs pages il faut avoir recours aux extensions `supertabular` ou `longtable`.

Parfois les tableaux par défaut de \LaTeX donnent une impression d'étroitesse. Si vous voulez leur donner plus d'extension, vous pouvez le faire en modifiant les valeurs de `arraystretch` et `tabcolsep` comme dans l'exemple suivant.

```
{
\renewcommand{\arraystretch}{1.5}
\renewcommand{\tabcolsep}{0.5cm}
\begin{center}
\begin{tabular}{|l|c|r|}
\hline
chat      & lapin & cochon  \\
\hline
oiseau    & vache & écureuil \\
\hline
\end{tabular}
\end{center}
}
```

chat	lapin	cochon
oiseau	vache	écureuil

Pour aller plus loin, voir le package `booktabs`.

Pour fusionner des cellules d'une même ligne on utilisera `\multicolumn{nombre de cellules à fusionner}{formatage}{contenu}` :



```

\begin{center}
\begin{tabular}{|lcr|}
\hline
chat chien      & lapin          & cochon \\
\hline
\multicolumn{2}{|c}{fusionnées} & vache \\
\hline
oiseau poule    & écureuil       & oie \\
\hline
\end{tabular}
\end{center}

```

chat chien	lapin	cochon
fusionnées		vache
oiseau poule	écureuil	oie

Pour composer des tableaux compliqués, on pourra utiliser le générateur visuel <https://www.tablesgenerator.com/>

Exercice 5

Reproduire le tableau suivant avec des en-têtes de colonnes pertinents :

Vitesse (km/h)	Nom Fantastique	Animal Mythique
300	Pégase	Griffon
250	Drakkar	Chimère
180	Phénix	Basilic

Placer le tableau dans une table flottante : comme pour les images, il est conseillé de laisser L^AT_EX s'occuper automatiquement du placement d'un tableau grâce à l'environnement `{table}`. À l'intérieur de cet environnement, il faut mettre l'environnement `{tabular}` et rajouter une légende afin de pouvoir faire référence au tableau. La syntaxe est donc

```

\begin{table}
\centering
\begin{tabular}{...}
...
\end{tabular}
\caption{Titre du tableau}\label{tab.montableau}
\end{table}

```

Tout ce qu'on a dit pour l'environnement `{figure}` reste valable pour l'environnement `{tabular}`.

2.14 Mettre son document sur plusieurs colonnes

Si on souhaite que tout le document soit écrit sur deux colonnes, on peut utiliser l'option de classe

```
\documentclass[... ,twocolumn]{...}
```

2.14.1 multicol(s)

Parfois on peut être conduit à écrire seulement une partie du document sur plusieurs colonnes, par exemple si on veut mettre côte à côte un texte et une image. La méthode la plus simple consiste à utiliser l'environnement `multicols` qui a besoin d'un paramètre : le nombre de colonnes que l'on veut.

Dans le **préambule** du document, il faut importer le paquet `multicol` (sans `s`)

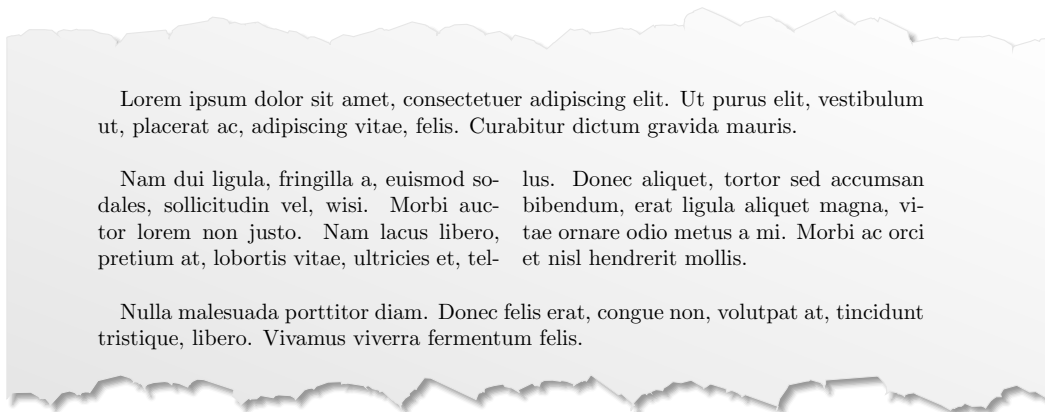
```
\usepackage{multicol}
```

Puis, dans le **corps du document** du document, il faudra utiliser l'environnement `multicols` (avec un `s`) :

```

\begin{multicols}{nb colonnes}
...
\end{multicols}

```

FIGURE 2.3 – Exemple de l'environnement `multicol`

Le nombre de colonnes est compris entre 1 et 10.

L'exemple de la figure 2.3 est obtenu par le code :

```
\documentclass[a4paper,10pt]{scrartcl}

\usepackage{multicol}
\usepackage{lipsum}

\begin{document}
\lipsum[1][1-3]
\begin{multicol}{2}
\lipsum[2][1-5]
\end{multicol}
\lipsum[3][1-3]
\end{document}
```

Exercice 6

Reproduire l'exemple de la figure 2.3. Passer de 2 à 3 colonnes.

On remarque que le texte est réparti automatiquement sur les deux colonnes. On peut forcer \LaTeX à effectuer un changement de colonne avec l'instruction `\columnbreak` suivie d'une ligne vide (voir la figure 2.4) :

```
\documentclass[a4paper,10pt]{scrartcl}

\usepackage{multicol}
\usepackage{lipsum}
\usepackage{graphicx}

\begin{document}
\lipsum[1][1-5]
\begin{multicol}{2}
\lipsum[2][1-5]
\columnbreak

\centering
\includegraphics[width=0.9\columnwidth]{hippopotenus}
\end{multicol}
```

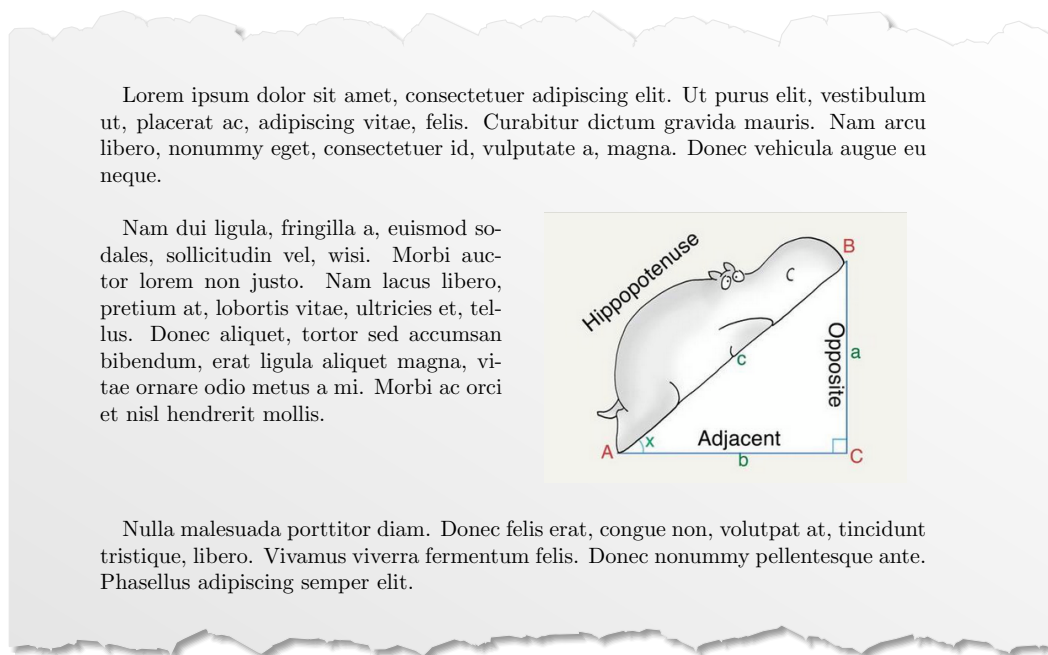


FIGURE 2.4 – Exemple de l’environnement multicol avec la commande `columnbreak`

```
\lipsum[3][1-5]
\end{document}
```

Avec `multicol` les colonnes ont toutes la même largeur, et ce n’est pas forcément ce que l’on veut : on pourrait par exemple vouloir réduire la largeur de la colonne contenant l’image, et donc élargir l’autre. Il y a, bien sûr, une solution : l’environnement `minipage`.

2.14.2 minipage

Pour créer une boîte qui simule la création d’une page avec d’éventuelles notes de bas de page, tableaux, listes, etc. on peut utiliser l’environnement `minipage`

```
\begin{minipage}[pos]{largeur}
...
\end{minipage}
```

Le paramètre `pos` contrôle l’alignement vertical de la boîte par rapport à la base du texte précédent. Trois possibilités : la lettre `c` (par défaut, positionne la minipage de sorte que son centre vertical soit aligné avec le centre des lignes de texte adjacentes), la lettre `t` (aligne la ligne du haut de la minipage sur la ligne de base du texte l’entourant) ou la lettre `b` (aligne la ligne du bas de minipage avec la ligne de base du texte l’entourant).

Voici un exemple :

2.15 Le formatage du texte ou comment écrire en italique en **GROS** en petit en chasse fixe...

Une petite ligne avant la minipage. Dans cet exemple on a créé une minipage faisant 30% de la largeur du texte et contenant un environnement `\texttt{itemize}`.

```
\medskip % espace verticale
```

```
\begin{minipage}[pos=b]{0.3\textwidth}
```

Ma mini-page très étroite qui contient une liste à puce:

```
\begin{itemize}
```

```
\item Bonjour\footnote{Une note}
```

```
\item Au-revoir
```

```
\end{itemize}
```

```
\end{minipage}
```

```
\hspace{2cm} % espace horizontale
```

```
\begin{minipage}[pos=c]{0.3\textwidth}
```

Et une deuxième mini-page juste à côté pour voir l'alignement.

```
\end{minipage}
```

```
\medskip % espace verticale
```

Encore du texte. Les paragraphes avant et après la boîte sont un peu trop collés à la minipage, il faut alors ajouter de l'espace verticalement.

Une petite ligne avant la minipage. Dans cet exemple on a créé une minipage faisant 30% de la largeur du texte et contenant un environnement `itemize`.

Ma mini-page très étroite qui contient une liste à puce :

- Bonjour
- Au-revoir

Et une deuxième mini-page juste à côté pour voir l'alignement.

Encore du texte. Les paragraphes avant et après la boîte sont un peu trop collés à la minipage, il faut alors ajouter de l'espace verticalement.

Avec une minipage on a besoin d'introduire des espacements verticaux. Voici ceux disponibles :

```
\vspace{<longueur>} % espacement vertical
\vspace*{<longueur>} % espacement vertical incompressible
\smallskip % idem que \vspace{\smallskipamount}
\medskip % idem que \vspace{\medskipamount}
\bigskip % idem que \vspace{\bigskipamount}
\\ % saut d'une ligne
\hspace{<longueur>} % espacement horizontal
\hspace*{<longueur>} % espacement horizontal incompressible
```

Remarques importantes sur les minipages :

- il ne peut pas y avoir de saut de page dans une minipage; il faut donc faire attention à ce qu'elle ne soit pas trop longue (sinon, elle sortira de la page);
- on doit obligatoirement spécifier la largeur, quitte à la trouver par tâtonnement, ce qui peut être lourd.

2.15 Le formatage du texte ou comment écrire en italique en gros en petit en chasse fixe...

Dans cette section on va apprendre à modifier les caractéristiques d'une police, c'est à dire à changer de fonte. Avec ces commandes on va pouvoir personnaliser un peu plus les documents. Attention toutefois de ne pas en faire de trop, on risquerait de rendre le document si ce n'est illisible tout du moins très moche.

La police utilisée dans ce polycopié est `fourier`, mais vous pouvez utiliser `lmodern` en chargeant le package `lmodern`. Une police possède 4 caractéristiques que l'on peut faire varier individuellement : la famille, la graisse, la forme, la taille (le corps). On peut donc avoir de nombreuses combinaisons possibles comme par exemple gras et italique. Toutefois toutes les combinaisons ne sont pas possibles et dépendent de la police utilisée. Par exemple en `lmodern` on ne peut pas obtenir des petites capitales en italique, ni en fonte sans serif.

La police c'est l'ensemble des caractères de toutes tailles, graisses, forme, d'une même famille. Une fonte est un ensemble de caractères d'une même police ayant la même graisse, la même taille, la même forme. Par exemple la police de ce paragraphe est `fourier`, la fonte utilisée est `serif corps 10pt` (`fourier`, sans serif, corps 8pt, italique est une autre fonte de la même police.)

Il existe deux types de commandes pour changer de fonte, les commandes à arguments et les commandes déclaratives :

- Les commandes à arguments servent pour modifier un mot ou un petit groupe de mots qui ne dépasse pas la taille d'un paragraphe. Un saut de paragraphe à l'intérieur d'une commande à argument déclenche une erreur de compilation. Elles commencent toutes par `\text` et prennent l'argument entre accolades. Par exemple la commande à argument pour mettre en gras est `\textbf{}` :

```
\textbf{gras}
```

gras

- Les commandes déclaratives agissent à partir de l'endroit où elles sont positionnées dans le texte sur toute la suite du texte jusqu'à l'ordre contraire. Un ordre contraire peut être donné par une autre commande déclarative annulant la première. Par exemple la commande graisse normale annulant la commande gras. La portée d'une commande déclarative peut être limitée par des accolades. `\bfseries` est la commande déclarative pour mettre en gras :

```
{\bfseries Tout ce qui se trouve ici même  
après un paragraphe ou des sections  
\section*{Une section en gras}  
sera en gras jusqu'à cette accolade} et  
ici tout revient normal
```

Tout ce qui se trouve ici même après un paragraphe ou des sections

Une section en gras

sera en gras jusqu'à cette accolade et ici tout revient normal

Une autre façon de limiter la portée d'une commande déclarative est de l'utiliser dans un environnement. Un environnement commence par un `\begin{environnement}` et se termine par un `\end{environnement}`

```
\begin{center}  
\bfseries Tout ce qui se trouve dans cet  
environnement sera centré et en gras.  
\end{center}  
Après la fermeture de l'environnement on  
retrouve une graisse normale.
```

Tout ce qui se trouve dans cet environnement sera centré et en gras.

Après la fermeture de l'environnement on retrouve une graisse normale.

On peut mettre plusieurs commandes déclaratives dans le même environnement ou la même paire d'accolades.

```
{\Large \rmfamily \itshape En gros, roman  
et italique}
```

En gros, roman et italique

2.15 Le formatage du texte ou comment écrire en italique en **GROS** en petit en chasse fixe...

LES FAMILLES DE POLICE

NOM	COMMANDE À ARGUMENT	COMMANDE DÉCLARATIVE	RÉSULTAT
Roman	<code>\textrm{Roman}</code>	<code>{\rmfamily Roman}</code>	Roman
Sans serif	<code>\textsf{Empattement}</code>	<code>{\sffamily Empattement}</code>	Empattement
Chasse fixe	<code>\texttt{typewriter}</code>	<code>{\ttfamily typewriter}</code>	typewriter

LA GRAISSE

NOM	COMMANDE À ARGUMENT	COMMANDE DÉCLARATIVE	RÉSULTAT
Graisse normale	<code>\textmd{Normale}</code>	<code>{\mdseries Normale}</code>	Normale
Écrit en gras	<code>\textbf{En gras}</code>	<code>{\bfseries En gras}</code>	En gras

LA FORME

NOM	COMMANDE À ARGUMENT	COMMANDE DÉCLARATIVE	RÉSULTAT
Écriture droite	<code>\textup{Droite}</code>	<code>{\upshape Droite}</code>	Droite
Normale			Normale
En italique	<code>\textit{Italique}</code>	<code>{\itshape Italique}</code>	<i>Italique</i>
Forme penchée	<code>\textsl{Penchée}</code>	<code>{\slshape Penchée}</code>	Penchée
En petites capitales	<code>\textsc{Petites capitales}</code>	<code>{\scshape Petites capitales}</code>	PETITES CAPITALES
En emphase	<code>\textemph{Emphatisé}</code>	<code>{\em Emphatisé}</code>	<i>Emphatisé</i>

Avec \LaTeX on ne peut pas modifier localement la taille du texte en augmentant le nombre de points (pt) comme on peut faire avec les traitements de texte WYSIWYG. On doit se contenter de taille relative par rapport à la taille normale du texte. Celle-ci est passée comme option à la commande `\documentclass`. \LaTeX accepte 3 tailles différentes : 10pt, 11pt, 12pt. Avec KOMA-Script on peut utiliser d'autres valeurs (14pt par exemple) mais on peut rencontrer des problèmes de compatibilité avec les autres packages. Notons que, de plus, le changement de taille se fait uniquement avec une commande déclarative :

```
{\tiny Texte}
{\scriptsize Texte}
{\footnotesize Texte}
{\small Texte}
{\normalsize Texte}
{\large Texte}
{\Large Texte}
{\huge Texte}
{\Huge Texte}
```

Texte Texte Texte Texte Texte Texte Texte Texte

Dans un manuscrit réalisé sur une machine à écrire, les mots importants sont valorisés en les soulignant; on peut obtenir ce résultat en \LaTeX avec la commande `\underline` :

... valorisés en les `\underline{soulignant}`;
on peut obtenir ...

... valorisés en les soulignant; on peut obtenir ...

Toutefois, dans un ouvrage imprimé on préfère les *mettre en valeur*¹¹. Les commandes de mise en valeur sont `\emph` et `\em`. La plupart du temps la mise en évidence du mot ou du groupe de mot se fera en le mettant en italique. Mais dans le cas d'un texte en italique `\emph{}` et `\em` mettront le texte en écriture droite.

Voici un texte `\emph{normal}` avec le mot `\emph{normal}` en emphase.

`{\itshape Voici un texte en \emph{italique}`
avec le mot `\emph{italique}` en emphase.)

Voici un texte *normal* avec le mot *normal* en emphase.
Voici un texte en italique avec le mot italique en emphase.

¹¹ *Emphasize* en anglais.

2.16 Définition d'environnements et de commandes personnelles

Il est possible de définir des commandes et des environnements personnels.

- **Exemple de définition d'une commande personnelle** : supposons de vouloir mettre les noms de famille en petites capitales, pour cela on doit utiliser la commande `\textsc` qui prend un argument et le met en petites capitales. Voici un exemple :

```
L'histoire de la théorie de l'intégration est jalonnée de noms célèbres comme
Augustin Louis \textsc{Cauchy}, Bernhard \textsc{Riemann}, Henri
\textsc{Lebesgue} ou encore Arnaud \textsc{Denjoy}.
```

```
L'histoire de la théorie de l'intégration est jalonnée de noms célèbres comme Augustin Louis
CAUCHY, Bernhard RIEMANN, Henri LEBESGUE ou encore Arnaud DENJOY.
```

Le problème de précéder ainsi est qu'il ne sera pas aisé de changer d'avis sur la composition des noms propres si on souhaite plus tard, disons par exemple ne plus afficher les prénoms, ou ne plus mettre les noms en petites capitales ou même rajouter les noms dans l'index. Une habitude importante à prendre avec \LaTeX est de définir des commandes permettant d'avoir accès au sens. Ici, ce que l'on compose, c'est des noms propres; il est donc parfaitement naturel de vouloir avoir une commande `\nompropre` qui prend deux arguments, le prénom et le nom et les compose de la façon voulue. Dans l'exemple précédent, on voudrait que `\nompropre{Augustin Louis}{Cauchy}` donne Augustin Louis CAUCHY.

Voyons comment définir de nouvelles commandes en \LaTeX .

- Pour définir une commande qui ne prend pas d'argument, on utilise la syntaxe suivante (de préférence dans la **préambule**) :

```
\newcommand{\SNCF}{S.N.C.F}
```

Ceci définit une commande `\SNCF` qui imprimera S.N.C.F. Notons que les espaces après une commande sont ignorés par \LaTeX . Si on veut obtenir une espace, on doit terminer la commande par des accolades et laisser une espace comme dans cet exemple :

```
bla \SNCF bla
```

```
bla S.N.C.Fbla
bla S.N.C.F bla
```

```
bla \SNCF{} bla
```

Ce serait une très mauvaise idée de mettre une espace dans la définition de la commande, car on aurait toujours une espace, y compris avant une ponctuation. Pour gérer ces deux situations on peut utiliser le package `xspace` :

```
\usepackage{xspace}
\newcommand{\SNCF}{S.N.C.F\xspace}
```

```
bla \SNCF bla
```

```
bla S.N.C.F bla
bla S.N.C.F.
```

```
bla \SNCF.
```

La commande `teste` ce qui suit la commande : si c'est une ponctuation ou `{ ou }`, elle ne fera rien; dans les autres cas, elle ajoute une espace. Une conséquence de ce fonctionnement est qu'une `\footnote` suivant `\SNCF` va produire une espace inopportune. Elle peut être évitée en tapant `{}` : `bla \SNCF{} \footnote{Toto}`.

- Pour avoir des commandes avec argument, il faut spécifier entre crochet, juste après le nom, le nombre d'arguments. On accède ensuite aux différents arguments avec `#1` pour le premier, `#2` pour le deuxième, etc. Par exemple

```
\newcommand{\guillemets}[1]{\og #1\fg}
```

crée une commande `\guillemets` à un argument telle que

```
\guillemets{science sans conscience n'est que ruine de l'âme}
```

produise «science sans conscience n'est que ruine de l'âme».

On veillera à n'utiliser `\newcommand` que dans le **préambule**. Pour des raisons de lisibilité, il vaut mieux mettre toutes les `\newcommand` après le chargement de tous les packages.

Exercice 7

Créer une commande à 2 arguments `\nompropre` qui imprime le prénom suivi du nom en petites capitales et l'utiliser pour reproduire l'exemple initiale de la section.

Exercice 8

Modifier la commande précédente pour qu'elle n'affiche plus que le nom de famille.

Exercice 9

Modifier la commande précédente pour qu'elle affiche le nom de famille en couleur.

- **Exemple de définition d'un environnement personnel** : on veut écrire un environnement `{citationFR}` pour mettre en page une citation comme suit : le texte cité se trouve dans un environnement `{quotation}` entouré par des guillemets. Pour définir notre environnement on utilise la syntaxe suivante :

```
\newenvironment{citationFR}{\begin{quotation}\og}{\fg\end{quotation}}
```

qui crée un environnement `{citationFR}` sans arguments tel que

```
\begin{citationFR}
\lipsum[1][1-5]
\end{citationFR}
```

produit

« Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. »

2.17 Pour aller plus loin

Voici quelques packages que vous pouvez utiliser pour personnaliser vos documents (avec parcimonie) :

- la documentclass KOMA-Script :
 - Guide complète <https://komascript.de/~mkohm/scrguien.pdf>
 - Traduction et adaptation du guide <https://archives.framabook.org/koma-script/index.html>
- gestion des marges :
 - typearea (package automatiquement chargé par les classes koma-script comme scrartcl) <https://www.ctan.org/pkg/typearea.html>
 - geometry <https://www.ctan.org/pkg/geometry.html>
- gestion des entêtes et pieds de page :
 - scrlayer-scrpage voir le chapitre 5 de la documentation de KOMA-Script
 - fancyhdr <https://www.ctan.org/pkg/fancyhdr.html>
- tableaux complexes :
 - array <https://www.ctan.org/pkg/array.html>



2 Premiers pas en L^AT_EX

- booktabs <https://www.ctan.org/pkg/booktabs.html>
- longtable <https://www.ctan.org/pkg/longtable.html>
- tabularx <https://www.ctan.org/pkg/tabularx.html>
- tcolorbox <https://www.ctan.org/pkg/tcolorbox>
- tabu <https://ctan.org/pkg/tabu>
- Quelques conseils : <https://people.inf.ethz.ch/markusp/teaching/guides/guide-tables.pdf>

3 Faire des bibliographies avec \LaTeX

La bibliographie d'un document est une liste structurée d'ouvrages cités dans le corps du texte et placée à la fin d'un article ou un rapport.

Pour produire une bibliographie avec \LaTeX , il est bien-sûr possible de confectionner une bibliographie à la main, mais il est beaucoup plus efficace de recourir au programme Bib \TeX .

Pour la réaliser il faut un endroit où sont stockées les données bibliographiques (auteur, titre, année de publication,...), un système pour faire des références dans le texte, et un modèle pour la mise en forme de la bibliographie.

La démarche pour la création d'une bibliographie est la suivante :

- ① création de la base de données bibliographique dans un fichier `.bib`,
- ② importation de quelques données dans le fichier `.tex` avec références croisées,
- ③ étapes de compilation $\LaTeX \rightarrow \text{Bib}\TeX \rightarrow \LaTeX$.

3.1 Étape n°① : création d'une base bibliographique dans un fichier `.bib`

Qui dit bibliographie dit, avant tout, base de données bibliographiques, c'est à dire un fichier où tu stockes toutes les données bibliographiques (l'auteur, le titre, l'année de publication, le nombre de pages...) et où \LaTeX ira chercher les renseignements pour construire la bibliographie.

Une base bibliographique est un simple fichier annexe dont l'extension est nécessairement `.bib` et qui respecte une syntaxe à respecter scrupuleusement.

Ce fichier ne comporte ni commande `\documentclass`, ni **preamble**, ni environnement `document` ; il contient juste une liste de documents. Certains de ces documents seront ensuite importés dans le document `.tex` par une référence.

Chaque entrée dans la base correspond à un document :

- qui commence par un `@` suivi du type d'ouvrage, par exemple un article, un livre, un manuel, etc. (la casse du type de document est indifférente : `@article = @Article = @ARTICLE`)
- qui est identifié de manière unique par un identifiant, appelé *clé*,
- caractérisé par :
 - des attributs : titre, auteur, date, etc.
 - des données : affectées aux attributs

Voici un exemple de fichier `.bib` constituant une mini base bibliographique de 3 entrées :

```
@Article{E103,
  author = {Loughran, Ellen},
  title = {Tentative {B}eginnings: {M}ontaigne {R}ewrites {H}is {E}arly {E}ssays},
  journaltitle = {Neophilologus},
  date = {2003},
  volume = {87},
  number = {3},
  pages = {371383}
}
```

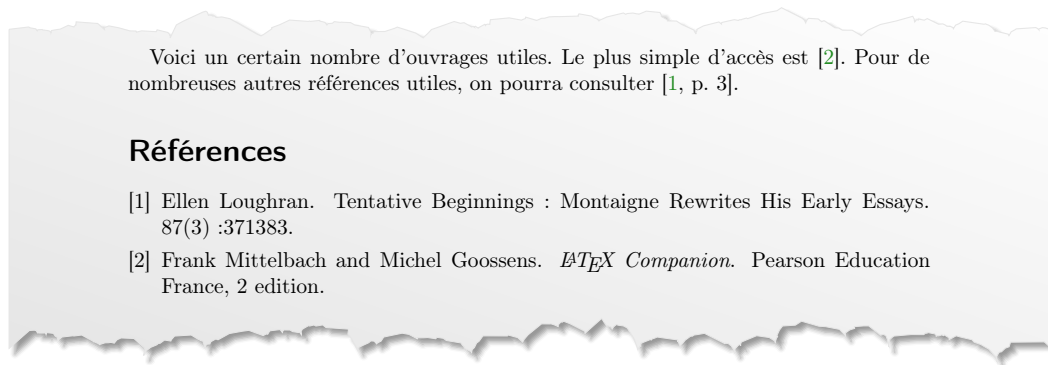


FIGURE 3.1 – Exemple de document avec bibliographie.

```
@Book{lcf,
  author = {Mittelbach, Frank and Goossens, Michel},
  title = {\LaTeX{} {C}ompanion},
  publisher = {Pearson Education France},
  date = {200602},
  location = {Paris},
  pagetotal = {1116},
  edition = {2}
}

@Manual{Leh,
  title = {The \texttt{biblatex} package},
  subtitle = {Programmable bibliographies and citations},
  author = {Lehman, Philipp},
  version = {1.7},
  date = {20111113},
  url = {http://tug.ctan.org/pkg/biblatex}
}
```

Pour créer sa base de donnée on pourra utiliser `google scholar` comme suit :

- chercher sur <https://scholar.google.fr> les livres/articles que vous voulez insérer dans votre bibliographie
- cliquer sur les deux chevrons >> après la description du livre choisi
- cliquer sur les guillemets
- choisir le format `bibtex`
- copier le tout dans un fichier `biblio.bib`

Des interfaces graphiques permettent de gérer des base sous le format d'un fichier `bib`, notamment le logiciel multi-plateforme `Jabref` <https://www.jabref.org/>

3.2 Étape n°② : appel de la bibliographie dans le fichier `.tex`

Analysons le document à la figure 3.1.

1. Dans les deux premières lignes, il y a des références aux éléments de la bibliographie. Ces références se font grâce à la commande `\cite`. Chaque élément de la bibliographie est identifié par une clef, disons `lcf` et c'est cette clef que l'on passe en argument à `\cite` :

```
Le plus simple d'accès est \cite{lcf}.
```

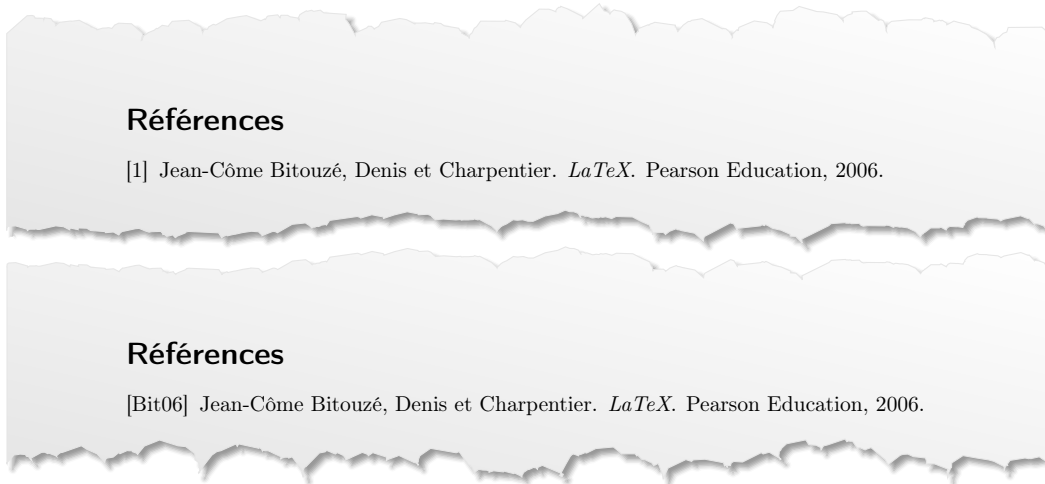


FIGURE 3.2 – Exemple d’un même document avec le style `plain` (en haut) puis avec le style `alpha` (en bas).

Cela imprimera automatiquement l’étiquette utilisée dans la bibliographie pour cette référence (typiquement, ce sera un numéro, disons [2], mais ce pourrait aussi être les initiales, disons [MG], selon le style bibliographique choisi) et le lien sera cliquable. Si jamais on veut spécifier un numéro de page ou un théorème particulier, `\cite` peut prendre un argument optionnel :

```
on pourra consulter \cite[p.~3]{E103}.
```

donnera [1, p. 3] si l’étiquette est [1] et [Lou, p. 3] si c’est [Lou].

Attention : seul les documents cités seront affichés dans la bibliographie.¹

2. À l’endroit où l’on veut faire apparaître la bibliographie on écrira les commandes `\bibliographystyle` et `\bibliography` (la première spécifie le style de la bibliographie et la seconde le fichier de bibliographie). Par exemple, les lignes

```
\bibliographystyle{plain}
\bibliography{biblio}
```

feront que le style `plain` sera utilisé et que c’est le fichier `biblio.bib` qui sera utilisé pour générer la bibliographie.

À la figure 3.2 on voit un exemple d’un même document avec le style `plain` et avec le style `alpha`.² Si on écrit un document en français, on veillera à utiliser un style adapté à la langue française (comme par exemple le style `plain-fr`).

3.3 Étape n°③ : compilations

Un fichier de bibliographie `.bib` ne doit jamais être compilé. Il faut faire un certain nombre de compilations et toutes ces opérations doivent s’appliquer au document `.tex`.

- **une première compilation** avec `pdflatex` (touche F6 avec Texmaker) ; cette compilation écrit les informations utiles à la construction de la bibliographie dans le fichier `.aux` ;
- ensuite **on compile** avec `bibtex` (touche F11 sous Texmaker) ; cette compilation produit un fichier `.bbl` qui contient un environnement `{thebibliography}` et qui sera incluse dans le document à la prochaine compilation ;

¹ Il est possible, mais déconseillé, d’importer tous les documents du fichier `bib` en écrivant avant l’importation de la bibliographie la commande `\nocite{*}`

² On pourra utiliser par exemple `plain`, `alpha`, `abbrv`, `acm`, `ieetr`, `siam`.

3 Faire des bibliographies avec L^AT_EX

- enfin **on compile deux fois** de suite avec `pdflatex` (touche F6 deux fois avec Texmaker
- Ne pas oublier de mettre à jour l’affichage avec la touche F7 (pour visualiser le fichier pdf produit).

Automatisation du processus de compilation avec Texmaker :
Menu “Options” → “configurer Texmaker” → “compil rapide” :
sélectionner le mode PdfLaTeX + Bib(la)tex + PdfLateX (×2) + Voir Pdf.

Attention, dans Texmaker, il faut parfois modifier les options de compilation comme indiqué ici <https://www.xm1math.net/doculatem/biblatex.html> : au lieu de `biber %.aux` écrire juste `biber %` (ou `bibtex %` selon le backend choisi).

Exercice 10

Reproduire le document donné à la figure 3.1.

Exercice 11

Écrire un document qui contient au moins trois livres OU articles de votre base de données bibliographiques.



3.4 Pour aller plus loin : le package BibL^AT_EX

Pour une mise en page personnalisée des citations et de la bibliographie on pourra utiliser le package `biblatex` :

- le package officiel <https://ctan.org/pkg/biblatex>
- “BibLaTeX expliqué à Mademoiselle Michu, étudiante en Sciences Humaines” <https://bertrandmasson.free.fr/index.php?article27/>

Brièvement, que faut-il modifier pour utiliser `biblatex` ?

1. Les citations s’obtiennent toujours avec la commande `\cite{<clé>}` :

```
Le plus simple d'accès est \cite{lcf}.
```

2. À l’endroit où l’on veut faire apparaître la bibliographie on remplace les lignes

```
\bibliographystyle{plain}  
\bibliography{biblio} % nom du fichier .bib
```

par la commande

```
\printbibliography
```

3. Dans le **préambule** on ajoutera

```
\usepackage{csquotes}  
\usepackage[<options>]{biblatex} % dans les options on indiquera le style choisi  
\addbibresource{biblio.bib} % extension .bib obligatoire
```

Exercice 12

Reproduire le document ci-dessous en utilisant le package BibL^AT_EX.

Voici un certain nombre d'ouvrages utiles. Le plus simple d'accès est [2]. Pour de nombreuses autres références utiles, on pourra consulter [1, p. 3].

Références

- [1] Ellen LOUGHRAN. « Tentative Beginnings : Montaigne Rewrites His Early Essays ». In : *Neophilologus* 87.3 (2003), p. 371383.
- [2] Frank MITTELBACH et Michel GOOSSENS. *L \TeX Companion*. 2^e éd. Paris : Pearson Education France. 1116 p.

Biblatex Cheat Sheet

For further details, explanations, hints, caveats, examples and alternatives to the backend Bibter, see the [Biblatex manual](#). For a list of *contributed* styles and extensions, see ctan.org/topic/biblatex.

Basic Setup

```

Compilation sequence: pdflatex → biber → pdflatex (×2).
\documentclass[language=option]{class}
...
\usepackage{inputenc}
\usepackage[babel,csoquotes=xpatch]{recommended}
\usepackage[backend=biber, options]{biblatex}
\addbibresource{options}<{first resource}>
\addbibresource{options}<{second resource}>
...
\begin{document}
... \printbibliography{<options>}
... \printbibliography{<options>}
... \end{document}
    
```

Common Package Options

```

{style} = {style} style of bibliography and citations
{citetyle} = {style} bibliography style
{refsegment} = {division} citation style
{auto-cite} = {division} new resection at document division
{sortcites} = {style} behaviour of \autocite etc.
{maxnames} = {boolean} whether to sort multiple citations
{minnames} = {integer} truncate longer name lists
{backref} = {boolean} no. of names in truncated name lists
{mincrossrefs} = {integer} whether to print 'back references'
{sorting} = {sort order} minimum number of cross references
{indexing} = {boolean} bibliography sort order
whether to enable indexing support
    
```

Sources of Bibliographical Data

```

\addbibresource{options}{(resource)} add to default resource list
\addbibresource{options}{(resource)} add to global resource list
Options:
location = local local file
           = remote HTTP/FTP
           = bibtex BibTeX
           = ris RIS
           = zotero rdff/xml Zotero RDF/XML
           = endnotexml EndNote XML
    
```

BibTeX Databases

A BibTeX database file is a plain text file with extension .bib. It consists of entries of the following form:

```

|-----|
|<entrytype>{<key>, e.g. |@book|tolkien-hobbit,
|field = <value>,      author = {Tolkien, J. R. R.},
|<field> = <value>,    title = {The Hobbit},
|<field> = <value>,    date = {YYYY-MM-DD},
|...|
|-----|
    
```

(*entrytype*) partially determines which fields are required and which optional. (*key*) is a unique identifier used in citation commands to reference the entry. (*field*) is the name of a database field and determines the expected format of (*value*). (*value*) is the value of the relevant (*field*) for the entry.

BibTeX Database Entry Types

Material from journals, magazines & newspapers:	Material from single-authored or co-authored books:
@article journal, magazine or newspaper article	@book book part with own title
@periodical whole issue of a periodical	@suppbook supplementally published as standalone book
@supperperiodical supplemental material in periodical	@book single-volume book by author(s) of whole multi-volume book
@imbook	Material from edited anthologies:
@suppbook	@inccollection contribution to anthology
@book	@collection supplemental material in anthology
@mvbook	@mvcollection single-volume edited anthology
	@mvcollection multi-volume collection
	Material from conference proceedings:
	@inproceedings article in conference proceedings
	@proceedings single-volume conference proceedings
	@mvproceedings multi-volume conference proceedings
	Material from works of reference:
@reference article in a reference work	@reference single-volume work of reference
@mvreference multi-volume reference work	
Material from technical & institutional publications:	
@manual technical or other documentation	
@report institutional report or white paper	
@patent patent or patent request	
@thesis work completed to fulfill degree requirement	
Material from online, informal & other sources:	
@online inherently online source	
@booleet informally published book	
@unpublished work not formally published	
@misc last resort (check manual first!)	
Special entries for database management:	
@set (static) entry 'set'	
@data data-container (cannot be cited)	

(*resource*) must be one of:
 (filename).bib local database
 http://.../(filename).bib remote
 ftp://.../(filename).bib remote
 \bibliography{(filename),(filename),...} adds ++ local BibTeX files.

Citations

Standard commands:	bare
<code>\cite{(pre)}{(post)}{(key)}</code>	parenthetical
<code>\mfootcite{(pre)}{(post)}{(key)}</code>	footnote (\footnote)
<code>\footciteext{(pre)}{(post)}{(key)}</code>	footnote (\footnotetext)
Common commands:	textual
<code>\textcite{(pre)}{(post)}{(key)}</code>	context-dependent
<code>\smartcite{(pre)}{(post)}{(key)}</code>	year/title only
<code>\cite*{(pre)}{(post)}{(key)}</code>	year/title only
<code>\parencite*{(pre)}{(post)}{(key)}</code>	superscript
<code>\supercite{(pre)}{(post)}{(key)}</code>	style-dependent
Style-independent commands:	style-dependent
<code>\m\autocite{(pre)}{(post)}{(key)}</code>	author list
<code>\c\autocite*{(pre)}{(post)}{(key)}</code>	compressed author list
Text commands:	(short) title
<code>\citeauthor{(pre)}{(post)}{(key)}</code>	(full) title
<code>\citetitle{(pre)}{(post)}{(key)}</code>	year
<code>\citeyear{(pre)}{(post)}{(key)}</code>	date
<code>\citedate{(pre)}{(post)}{(key)}</code>	URL
<code>\citeurl{(pre)}{(post)}{(key)}</code>	
Multi-volume commands:	cite by volume + page
<code>\volcite{(pre)}{(vol)}{(page)}{(key)}</code>	parenthetical
<code>\m\volcite{(pre)}{(vol)}{(page)}{(key)}</code>	footnote (\footnote)
<code>\c\volcite{(pre)}{(vol)}{(page)}{(key)}</code>	footnote (\footnotetext)
<code>\mvolcite{(pre)}{(vol)}{(page)}{(key)}</code>	context-dependent
<code>\m\mvolcite{(pre)}{(vol)}{(page)}{(key)}</code>	textual
<code>\c\mvolcite{(pre)}{(vol)}{(page)}{(key)}</code>	style-dependent
Standalone citation commands:	full reference
<code>\fullcite{(pre)}{(post)}{(key)}</code>	full reference in footnote
<code>\footfullcite{(pre)}{(post)}{(key)}</code>	
Inclusion in bibliography without citation:	inclusion only
<code>\nocite{(key)}</code>	with notes
<code>\nocite*{(pre)}{(post)}{(key)}</code>	with parenthetical notes
<code>\pnotecite{(pre)}{(post)}{(key)}</code>	with footnote notes
<code>\mnotecite{(pre)}{(post)}{(key)}</code>	
Author-year and author-title styles only.	
<code>\caption{command(s)}</code>	also provided, e.g. \Textcite, \Autocites.
<code>\multicite{command(s)}</code>	available.
e.g. <code>\cites{(multipost)}{(pre)}{(post)}{(key)}{(pre)}</code>	
<code>{(post)}{(key)}...</code>	
<code>\numcites{(key)}...</code>	Numerical styles only.
<code>\starred{style}</code>	available to include extra year information.

Multiple, Divided & Filtered Bibliographies

Bibliography section Document part with its own bibliography.
Bibliography segment Document part corresponding to a sub-division of a global bibliography.

See package options `refsection` and `refsegment` for automated creation according to document division. Finer-grained control is also possible:

```
\begin{refsection}
[<resource>...% replace default list
...
\end{refsegment}
```

Bibliography category Topic or source type corresponding to a sub-division of a global bibliography.

```
\DeclareBibliographyCategory{<category>} new category
\addtocategory{<category>}{<key>} add entry to category
```

Printing Bibliographies

```
\printbibliography{<options>} typeset the bibliography
\printbiblist{<options>}{<name>} typeset bibliography list (<name>)
e.g. shorthand
```

Options:

```
env = <name> e.g. bibliography
heading = <heading> e.g. subbibliography, (sub)bibintoc
title = <title>
prenote = <name> } define start/end notes with
postnote = <name> } \defbibnote{<name>}{<text>}
section = <integer> for refsection <integer>
segment = <integer> for refsegment <integer>
categoryn = <category> only entries in <category>
keywordn = <keyword> only entries with keyword (<keyword>)
typea = <entrytype> only entries of type (<entrytype>)
n A negated filter is available as not- e.g. notcategory=<category>.
\bibbysection{<options>} all refsection bibliographies
\bibbysegment{<options>} all refsegment bibliographies
\bibbycategory{<options>} bibliographies for all categories
```

Biber

```
biber [options] file[.bcb] biber [options] -tool <datasource>
```

By default, Biber reads a .bcb and produces a .bbl which L^AT_EX needs to produce a document's citations and bibliography. But Biber also has a powerful "tool" mode. The manual explains the details but `biber -help` is a more comprehensible starting point.

To produce a document-specific .bib:

```
biber --output_format=bibtex --output_resolve <filename>.bcb
```

Copyright ©2017 Clea F. Rees ReesC21@cardiff.ac.uk Rev. 66.44 2017-06-24

International Standard Audiovisual Number
 International Standard Book Number
 International Standard Music Number
 International Standard Technical Report Number
 International Standard Serial Number
 International Standard Work Code
 record of work's abstract
 for annotated bibliographies
 local link
 library name, call number or similar
 fall-back label
 special designator, overrides label in citations
 override default introduction of shorthand

Special fields for non-printable data:
 arbitrary TeX code
 separated list of keywords
 per-entry options
 citation key aliases
 another entry key, related options for options
 relationship identifier for related
 override value of relatedtype
 list of entry keys in @set
 another entry key
 another entry key
 entry key for @xdata container
 babel/polyglossia language identifier
 polyglossia options for langid
 gender of author or editor
 modify sorting
 sort key, overrides everything except presort
 replaces author or editor when sorting
 sortkey if entry has shorthand
 replaces title when sorting
 replaces year (from date) when sorting

Special fields for non-printable data:

```
execute arbitrary TeX code
keywords separated list of keywords
options per-entry options
ids citation key aliases
related another entry key, related options for options
relatedtype relationship identifier for related
relatedstring override value of relatedtype
entryset list of entry keys in @set
crossref another entry key
xref another entry key
xdata entry key for @xdata container
langid babel/polyglossia language identifier
langdopts polyglossia options for langid
gender gender of author or editor
presort modify sorting
sortkey sort key, overrides everything except presort
sortname replaces author or editor when sorting
sortshorthand sortkey if entry has shorthand
sorttitle replaces title when sorting
indexsorttitle replaces year (from date) when sorting
```

Built-In Styles

```
citestyle numeric numeric
numericc,v numeric
alphanumeric alphabetic
authoryearc,ib,ic author-year
authorarticlec,ib,ic,ic,ic author-year
authorarticlec,ib,ic,ic author-year
verboseb,ib,ia,n verbose
verbose=tradl/2/3t full references on first citation
readinga 'traditional' footnote citations
draft reading list
debug show entry keys
c-comp option (compact) ib ibid option (use ibidem) ic icomp
option (compact & ibidem) in inote option (notes & ibidem).
n -note option (full citations as footnotes). t -terse option (omit title
if unique). ic -icomp option (compact & terse).
ib -ibid option (compact, terse & ibidem).
tr The three use different scholarly abbreviations in different ways.
v -verb option (verbose). t Equivalent to citestyle=authorarticle.
```

B_ATEX Database Fields

INDIVIDUALS
 author^d author(s) of title, author type specifies kind
 bookauthor author(s) of book title
 editor^a editor(s), editor type specifies role
 editors/b/c secondary editor(s), editors/b/c type for roles
 afterword author(s) of afterword
 annotator author(s) of annotations
 commentator author(s) of commentary
 forward author(s) of forward
 introduction author(s) of introduction
 translator translator(s) of (book)title
 of patent of patent

ORGS
 institution university or similar
 organization manual/website publisher or event sponsor
 publisher^o publisher(s)
 title title
 if different from title
 indextitle title of book
 booktitle^u title of multi-volume book
 maintitle^u or journal^p
 issuetitleⁿ title of journal special issue
 eventtitle^a title of conference or event
 reprinttitle title of a reprint of the work
 series^e publication series

TITLES
 volume volume of journal or multi-volume book
 number numbered issue of journal or book in series
 part number of physical part of logical volume
 issue non-number issue of journal
 volumes number of volumes for multi-volume work
 edition as (<integer>) rather than ordinal
 version revision number for software or manual
 pubstate publication state

VOLUMES & VERSIONS
 pages page list or range
 pagetotal total number of pages
 (book)pagination pagination format of (book)title
 date^o publication date as (YYYY-MM-DD)
 erendate conference or event date as (YYYY-MM-DD)
 uridate access date for url as (YYYY-MM-DD)
 location^o or address, where published
 venue of event
PLACES DATES
 url URL
DIGITAL
 doi Digital Object Identifier
 eid electronic identifier of @article
 eprint archive-specific electronic identifier
 eprinttype type of identifier, eprintclass for further details
TYPES
 type of @manual, @patent, @report or @thesis
 entrysubtype for finer-grained specification of type
 addendum miscellaneous data printed at end of entry
 note miscellaneous data printed within entry
MISC.
 howpublished non-standard publication details
 language^o language of work
^a An -addon field is available e.g. nameaddon, eventtitleaddon.
^o An orig- field is available e.g. origdate, origlanguage.
^s A short- field is available e.g. shortauthor, shorttitle.
^u A -subtitle field is available e.g. subtitle, mansubtitle.

4 Mathématiques

\LaTeX est aujourd'hui le standard utilisé dans l'édition mathématique (que ce soit pour des articles de recherche, des livres, des photocopiés de cours, des feuilles de TD, etc.). Le but de ce chapitre est d'apprendre à taper des formules simples avec \LaTeX . Au début, cela fait beaucoup de commandes à mémoriser, mais avec la pratique, on finit par connaître la plupart des commandes et cela permet de taper les formules relativement rapidement. Ne pas hésiter à utiliser les menus de symboles de *Texmaker* dans le panneau de gauche.

4.1 Documents avec théorèmes, propositions, etc.

Les environnements pour écrire des théorèmes, des corollaires, des lemmes et autres propositions ainsi que les démonstrations, les exercices etc. ne sont pas définis par défaut.

Pour apprendre à produire un document qui contient ces environnements, on va analyser l'exemple donné à la figure 4.1. On remarque que ces environnements peuvent être regroupés en quatre catégories :

- les théorèmes, corollaires, lemmes, propositions ont leur nom composé **en gras** puis leur contenu composé *en italique*;
- les définitions ont leur nom composé **en gras** et leur contenu composé en droit;
- les remarques ont leur nom composé *en italique* et leur contenu composé en droit;
- les démonstrations sont comme les remarques, sauf qu'elles impriment un carré à la fin de la démonstration.

On note d'ailleurs que la numérotation est automatique et choisie de telle façon que le numéro de section est imprimé avant le numéro du théorème et remis à 1 après chaque changement de section. De plus, la numérotation de tous ces environnements se suit (la première proposition après la définition 1.1 est la proposition 1.2, pas la proposition 1.1).

- Pour composer des théorèmes, on a besoin de charger le package `amsthm` dans le **préambule**. Comme tous les autres packages, il vaut mieux le charger avant `hyperref` ; il vaut mieux également le charger avant `lmodern` car si jamais on change ce package pour un autre (par exemple `txfonts`), il peut y avoir des incompatibilités :

```
\usepackage{amsthm}
```

- On doit maintenant définir un environnement qui permettra d'écrire le théorème proprement dit. Dans le **préambule** on écrit

```
\newtheorem{theoreme}{Théorème}[section]
```

Le premier argument est le nom de l'environnement ¹ ; il ne doit pas comporter d'accents. Si on utilise `theoreme`, on tapera plus tard dans le **corps du document** `\begin{theoreme}` ; si on utilise `theo`, il faudra taper `\begin{theo}`. Un nom court est plus rapide à taper, mais rend le fichier source moins lisible. Le deuxième argument est ce que l'on veut qui soit imprimé, ici «Théorème» (avec majuscule et accents). Le troisième argument, entre crochets car optionnel, permet de rajouter le numéro des sections avant le numéro des théorèmes et réinitialise ce numéro lors de chaque changement de section. Si on utilisait la classe de document `scrbook` au lieu de `scrartcl`, on voudrait probablement mettre `chapter` ici au lieu de `section`.

¹ Faire attention, tous les noms ne sont pas permis. Une erreur fréquente est de vouloir appeler une définition `def` ; cela provoque une erreur car c'est un nom réservé. De même, `th` est déjà pris (c'est le caractère θ).

1 Rappels

Définition 1.1. Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero.

Proposition 1.2. *Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis.*

Démonstration. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna.

Notons le petit carré automatiquement inséré pour signaler la fin de la preuve (il s'appelle QED, acronyme de "quod erat demonstrandum"). \square

Corollaire 1.3. *Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.*

Exercice 1. *Mauris ut leo. Cras viverra metus rhoncus sem.*

Exercice 2. *Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat.*

2 Approfondissements

Définition 2.1. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo.

Lemme 2.2. *Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi.*

Démonstration. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. \square

Théorème 2.3. *Cras nec ante. Pellentesque a nulla.*

Démonstration. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. \square

Remarque. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Exercice 3. *Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero.*

FIGURE 4.1 – Exemple d'un document avec théorèmes, propositions, etc.

Ensuite, on veut définir par exemple un corollaire qui sera numéroté de la même façon que `theoreme`. La syntaxe est

```
\newtheorem{corollaire}[theoreme]{Corollaire}
```

La seule chose qui change par rapport à ce qu'on a utilisé pour définir l'environnement `theoreme` est l'argument optionnel qui est désormais entre les deux arguments entre accolades. Cela veut dire qu'il faut adopter la même numérotation que `theoreme`.

Tous les autres environnements s'obtiennent de la même façon, sauf pour la remarque qui est non numéroté; pour elle, on utilise

```
\newtheorem*{remarque}{Remarque}
```

Finalement, si on voulait un environnement numéroté indépendamment de tout («Exercice 1» puis «Exercice 2», etc.), on utiliserait

```
\newtheorem{exo}{Exercice}
```

- Il faut enfin sélectionner un style; cela se fait avec la commande `\theoremstyle`. Le style `plain` compose le nom de l'environnement en gras et son contenu en italique; le style `definition` compose le nom de l'environnement en gras et son contenu en droit; le style `remark` compose le nom de l'environnement en italique et son contenu en droit.

```
\theoremstyle{plain}
  \newtheorem{theoreme}{Théorème}[section]
  \newtheorem{proposition}[theoreme]{Proposition}
  \newtheorem{corollaire}[theoreme]{Corollaire}
  \newtheorem{lemme}[theoreme]{Lemme}
  \newtheorem{exo}{Exercice}
\theoremstyle{definition}
  \newtheorem{definition}[theoreme]{Définition}
\theoremstyle{remark}
  \newtheorem*{remarque}{Remarque}
```

Il n'y a pas de styles pour les démonstrations, elles sont produites directement en utilisant dans le corps du document l'environnement `{proof}` :

```
\begin{proof}
Bla bla bla bla bla.
\end{proof}
```

Pour personnaliser l'apparence de ces environnements on pourra s'appuyer sur l'un des package suivants :

- `ntheorem` <https://ctan.org/pkg/ntheorem>
- `thmtools` <https://ctan.org/pkg/thmtools>
- `bclogo` <https://ctan.org/pkg/bclogo>

Exercice 13

En utilisant les outils qu'on vient de décrire, reproduire le document de la figure 4.1.

4.2 Les différents modes mathématiques

Pour afficher des formules mathématiques avec \LaTeX , il faut lui indiquer qu'il s'agit d'un environnement mathématique. Il existe deux modes mathématiques :

- ① les formules dans le texte (appelée *textstyle* ou *inline*),
- ② les formules en évidence (dites *displaystyle*). Ces dernières peuvent être sans ou avec numérotation.

Pour des mises en forme un peu évoluées, le package `amsmath` est pratiquement indispensable. Mieux encore, le package `mathtools` qui charge `amsmath` et en corrige quelques bogues et limitations.

- **Formule dans le texte**

Pour mettre une formule dans le texte, il suffit de mettre la formule entre deux dollars :

```
Soit $y=f(x)$ alors
```

```
Soit  $y = f(x)$  alors
```

- **Formule en évidence, sans numérotation**

Pour mettre une formule en évidence, il faut mettre la formule entre `\[` et `\]` :

```
Soit
\[
y=f(x)
\]
alors
```

```
Soit

$$y = f(x)$$

alors
```

On peut aussi utiliser l'environnement `{equation*}` :

```
Soit
\begin{equation*}
y=f(x)
\end{equation*}
alors
```

```
Soit

$$y = f(x)$$

alors
```

Attention : laisser des lignes blanches en mode `displaystyle` produit une erreur de compilation.

Nota Bene : pour mettre une formule en évidence, l'utilisation des doubles dollars `$$. . . $$` au lieu de `\[. . . \]` (qu'on peut trouver dans certains vieux manuels et vieux codes qui se transmettent de génération en génération. . .) sera considéré comme une faute grave et sévèrement sanctionné.²

- **Formule en évidence, avec numérotation**

Pour une équation numérotée, il faut utiliser l'environnement `{equation}` :

```
Soit
\begin{equation}
y=f(x)
\end{equation}
alors
```

```
Soit

$$y = f(x) \tag{4.1}$$

alors
```

Pour faire référence à une équation numérotée, il faut placer un `\label` avec une étiquette, par exemple :

```
Soit
\begin{equation}\label{eq.fonction.f}
y=f(x)
\end{equation}
alors
```

```
Soit

$$y = f(x) \tag{4.2}$$

alors
```

puis utiliser `\eqref{eq.fonction.f}` pour imprimer (4.2) avec les parenthèses ajoutées automatiquement et `\pageref{eq.fonction.f}` pour imprimer le numéro de page où se trouve l'équation (ici 56). Comme la table des matières, ces commandes requièrent **deux compilations** successives pour fonctionner correctement. Se rappeler que les commandes `\label`, `\ref` et `\pageref` ne sont pas limitées aux équations, mais fonctionnent aussi pour des sections ou des théorèmes (en fait, pour tout ce qui est numérotés automatiquement).

²Voir le document "Liste des péchés des utilisateurs de \TeX " pour plus de détails <https://ctan.org/tex-archive/info/l2tabu/french>

Exercice 14

Reproduire le document suivant :

Soit f une fonction vérifiant

$$f(x) = 2x + 1. \quad (1)$$

On a $f(x) - 1 = 2x$ d'après la formule (1).

Propriétés des *modes mathématiques* :

- les espaces saisis au clavier sont ignorés (cf. page 58) ;
- tous les caractères alphabétiques sont en italique mais différentes de l'italique en *mode texte*.

`\Frown{}` $y = f(x)$ est différentiable`\Frown{}` $y = f(x)$ est différentiable\$`\Smiley{}` $y = f(x)$ est différentiable

☹ $y = f(x)$ est différentiable
 ☹ $y = f(x)$ est différentiable
 ☺ $y = f(x)$ est différentiable

Nota Bene : la non utilisation du mode mathématique pour une formule (resp. l'utilisation du mode mathématique pour du texte) sera considéré comme une faute grave et sévèrement sanctionné.

On verra plus loin comment ajouter des espaces ou écrire du texte dans un environnement mathématique.

4.3 Indices et exposantes

Deux opérations fondamentales en mode mathématique sont la mise en exposant et la mise en indice.

- Pour obtenir un indice, il faut utiliser le caractère `_` (*underscore*) qui s'obtient avec la touche «8». Par exemple, taper `x_1` donnera x_1 . Attention, taper `x_12` ne donne pas x_{12} mais x_12 : seul le premier caractère tapé après `_` est mis en indice ; pour obtenir x_{12} , il faut taper `x_{12}`.
- Pour obtenir un exposant, il faut utiliser le caractère `^` (accent circonflexe qui s'obtient en tapant deux fois la touche «^»). Par exemple, `x^2` donne x^2 . De même que pour les indices, `x^23` donne x^23 tandis que `x^{23}` donne x^{23} .

On peut bien sûr combiner les deux, dans l'ordre que l'on veut : `x_1^2` ou `x^2_1` donnent x_1^2 .

Exercice 15

Taper les formules suivantes :

$$(x^2)^3 = x^{2^3}$$

$$F_n = 7^{2^n} + 1$$

Remarquer la différence dans les formules suivantes :

`$(x^2)^3$` et `$(x^2)^3$` et `$(\left(x^2\right))^3$` et `$(\left(x^2\right))^3$`

$(x^2)^3$ et $(x^2)^3$ et $(x^2)^3$ et $(x^2)^3$

4.4 Racine carrée, racine n-ième

La racine carrée s'obtient par `\sqrt{...}` et la racine n-ième par `\sqrt[n]{...}`.

```
\[
\sqrt{1+x} + \sqrt[3]{1+x}
\]
```

$$\sqrt{1+x} + \sqrt[3]{1+x}$$

Exercice 16

Taper les formules suivantes :

$$u_{n+1} = \sqrt[n]{1+u_n}$$

$$x_5 = \sqrt{1 + \sqrt{2 + \sqrt{3 + \sqrt{4 + \sqrt{5}}}}}$$

4.5 Texte et espaces dans une formule *displaystyle*

La commande `\text{}` permet d'insérer du texte dans une formule *displaystyle* :

```
\[
y = x^2 \text{ et donc } x = \pm\sqrt{y}
\]
```

$$y = x^2 \text{ et donc } x = \pm\sqrt{y}$$

Exercice 17

Taper la formule :

$$(\sqrt{x})^2 = x \text{ mais } \sqrt{x^2} \neq x \text{ en général.}$$

Espaces en mode mathématique

Les espaces s'obtiennent via des commandes :

- 2 cadratins avec la commande `\qquad`

```
\[a \qquad b\]
```

$$a \quad b$$

- 1 cadratin avec la commande `\quad`

```
\[a \quad b\]
```

$$a \quad b$$

- inter-mot avec la commande `\`

```
\[a \ b\]
```

$$a \ b$$

- épaisse avec la commande `\;`

```
\[a \; b\]
```

$$a \; b$$

- moyenne avec la commande `\:`

```
\[a \: b\]
```

$$a \: b$$

- fine avec la commande `\,`

<code>\[a \, b\]</code>	ab
• normale	
<code>\[a b\]</code>	ab
• fine négative avec la commande <code>\!</code>	
<code>\[a \! b\]</code>	ab

 Texte en indice

Il est fréquent que du texte soit utilisé en indice, par exemple $f_{\text{opt}}(x)$. Si les indices ou les exposants ont un rôle descriptif, ils sont à saisir en tant que texte et donc en argument de la commande `\text` :

Ne pas écrire <code>\f_{opt}(x)</code> mais <code>\f_{\text{opt}}(x)</code>	Ne pas écrire $f_{opt}(x)$ mais $f_{\text{opt}}(x)$
Ne pas écrire <code>\f_{\text{i}}(x)</code> mais <code>\f_i(x)</code>	Ne pas écrire $f_i(x)$ mais $f_i(x)$

4.6 Symboles d'usage courant

Certains symboles s'obtiennent directement au clavier :

<code>\$ () [] = + - / < > , ; : ! \$</code>	<code>()[] =+-/<>,;:!</code>
--	------------------------------------

La plupart des symboles s'obtiennent via des commandes.³

<code>\infty</code>	<code>\forall</code>	<code>\exists</code>	<code>\nexists</code>	<code>\partial</code>	<code>\pm</code>	<code>\mp</code>
∞	\forall	\exists	\nexists	∂	\pm	\mp
<code>\times</code>	<code>\neq</code>	<code>\leq</code>	<code>\geq</code>	<code>\approx</code>	<code>\simeq</code>	<code>\equiv</code>
\times	\neq	\leq	\geq	\approx	\simeq	\equiv
<code>\in</code>	<code>\subset</code>	<code>\cup</code>	<code>\cap</code>	<code>\setminus</code>	<code>\emptyset</code>	<code>\ell</code>
\in	\subset	\cup	\cap	\setminus	\emptyset	ℓ
<code>\to</code>	<code>\mapsto</code>	<code>\implies</code>	<code>\impliedby</code>	<code>\iff</code>	<code>\nearrow</code>	<code>\searrow</code>
\rightarrow	\mapsto	\implies	\impliedby	\iff	\nearrow	\searrow

Pour la négation d'un symbole, on peut utiliser `\not`. Par exemple, `\not\subset` fournit $F \not\subset E$.

Exercice 18

Taper les formules suivantes :

$$y = x^2 \iff x = y^{1/2}$$

$$x > 0 \implies x^2 \neq 0$$

$$x \in X \setminus Y \implies x \notin Y$$

³L'application Web [detexify](#) permet de dessiner un symbole à la souris et obtenir la commande \LaTeX correspondante.

4.7 Points de suspension

COMMANDE	<code>\cdot</code>	<code>\dots</code>	<code>\cdots</code>	<code>\ddots</code>	<code>\vdots</code>
RÉSULTAT	⋮	⋮

La différence entre `.` et `\cdot` ainsi qu'entre `\dots` et `\cdots` est qu'ils ne sont pas sur la même ligne horizontale :

```
bla bla . bla bla  $\cdot$ 
```

```
bla bla  $\dots$  bla bla  $\cdots$ 
```

```
bla bla . bla bla .
bla bla ... bla bla ...
```

4.8 Fonctions mathématiques

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

Pour mettre des bornes à ces objets, il suffit d'utiliser les commandes d'indice et/ou d'exposant. **Le placement des indices et exposants change** selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle*.

```
 $\lim_{x \to 0} f(x)$ 
\[
\lim_{x \to 0} f(x)
\]
```

```
limx→0 f(x)
limx→0 f(x)
```

On peut aussi définir de nouveaux opérateurs avec

```
\DeclareMathOperator{\cotan}{cotan}
```

qui permettra d'utiliser `\cotan` pour obtenir `cotan`. Il y a aussi une variante étoilée pour les objets du type `\lim` ou `\max` qui prennent des bornes. Par exemple

```
\DeclareMathOperator*\supess{sup\,ess}
```

définira une commande `\supess` imprimant `supess` et se comportant comme `\lim` vis-à-vis des indices.

Exercice 19

Taper les formules suivantes :

$$\cos^2(x) + \sin^2(x) = 1$$

$$2^{\ln(x)} = x^{\ln(2)}$$

4.9 Flèches extensibles

Le package `amsmath` dispose de deux flèches extensibles, `\xleftarrow[]{}{}` et `\xrightarrow[]{}{}`. Pour mettre une formule dessus, on utilise l'argument obligatoire tandis que pour mettre une formule dessous, on utilise l'argument optionnel :

```
\[
f(x) \xrightarrow{\text{d'après (H)}} a
\]
\[
f(x) \xrightarrow[x\to 0]{} a
\]
\[
f(x) \xrightarrow[x\to 0]{\text{d'après (H)}} a
\]
```

$$f(x) \xrightarrow{\text{d'après (H)}} a$$

$$f(x) \xrightarrow[x\to 0]{} a$$

$$f(x) \xrightarrow[x\to 0]{\text{d'après (H)}} a$$

4.10 Fractions et coefficients binomiaux

Pour les fractions, on utilise la commande `\frac{}{}` qui prend deux arguments, dans l'ordre le numérateur et le dénominateur. Le rendu change selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle* :

```
Bla bla $\frac{1}{2}+1$ bla bla
\[
\frac{1}{2}+1
\]
```

Bla bla $\frac{1}{2} + 1$ bla bla

$$\frac{1}{2} + 1$$

Exercice 20

Taper les formules suivantes :

$$x^{1/3} = x^{\frac{1}{3}} = \sqrt[3]{x}$$

$$\sqrt{2} = 1 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \ddots}}}$$

Il peut être utile, dans certaines situations, de forcer le style d'une fraction (afficher une fraction dans le texte comme si elle était mise en évidence et vice-versa). Pour forcer le rendu *displaystyle* on peut utiliser la commande `\dfrac{}{}` et pour forcer le rendu *inline* on peut utiliser la commande `\tfrac{}{}`. Voici des exemples

4 Mathématiques

Dans le texte: style prédéfini `\frac{1}{2}+1`, style inline `\tfrac{1}{2}+1`, style display `\dfrac{1}{2}+1`.

Les mêmes mais dans des formules display:

```
\[
\frac{1}{2}+1,\quad \tfrac{1}{2}+1,\quad \dfrac{1}{2}+1.
\]
```

Dans le texte : style prédéfini $\frac{1}{2} + 1$, style inline $\frac{1}{2} + 1$, style display $\frac{1}{2} + 1$.
 Les mêmes mais dans des formules display :

$$\frac{1}{2} + 1, \quad \frac{1}{2} + 1, \quad \frac{1}{2} + 1.$$

Pour les coefficients binomiaux on utilise la commande `\binom{}{}` (et les variantes `\dbinom{}{}` et `\tbinom{}{}`).

```
\[
\binom{n}{k}
\]
\[
\tbinom{n}{k}
\]
 $\binom{n}{k}$ 
 $\dbinom{n}{k}$ 
```

$$\binom{n}{k}$$

$$\binom{n}{k}$$

4.11 Lettres grecques

Pour afficher les lettres grecques, il suffit de précéder le nom de la lettre par un *backslash*; par exemple `\alpha` donne α . Voici une liste complète des lettres grecques disponibles sous \LaTeX :

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

Exercice 21

Taper la formule :

$$\frac{\pi^2}{6} + \gamma = \Gamma(n) + \sqrt[n]{1+\alpha}$$

4.12 Grands opérateurs : intégrales, sommes, produits, etc.

Le rendu des grands opérateurs change selon qu'il s'agit d'une formule *inline* ou d'une formule *displaystyle*. Certaines de ces commandes prennent, tout comme max ou lim, des bornes. Le principe est le même : on utilise des indices ou des exposants pour les taper. Le placement des indices et exposants dépend de si la formule est mise en évidence ou pas.

```
Bla bla  $\int_a^b f(x) dx$  bla bla  $\sum_{i=0}^n u_n$ 
\[
\int_a^b f(x) dx
\]
\[
\sum_{i=0}^n u_n
\]
```

```
Bla bla  $\int_a^b f(x) dx$  bla bla  $\sum_{i=0}^n u_n$ 
```

$$\int_a^b f(x) dx$$

$$\sum_{i=0}^n u_n$$

COMMANDE	<code>\int</code>	<code>\iint</code>	<code>\iiint</code>	<code>\sum</code>	<code>\prod</code>
RÉSULTAT <i>inline</i>	\int	\iint	\iiint	Σ	Π
RÉSULTAT <i>displaystyle</i>	\int	\iint	\iiint	Σ	Π

COMMANDE	<code>\bigcup</code>	<code>\bigcap</code>	<code>\bigsqcup</code>	<code>\bigoplus</code>	<code>\bigotimes</code>	<code>\coprod</code>
RÉSULTAT <i>inline</i>	\cup	\cap	\sqcup	\oplus	\otimes	\amalg
RÉSULTAT <i>displaystyle</i>	\cup	\cap	\sqcup	\oplus	\otimes	\amalg

Pour mettre plusieurs lignes dans les indices, il faut utiliser `\substack{}`; à l'intérieur de l'argument de `\substack`, on passe à la ligne avec `\\`.

```
\[
L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}
\]
```

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$$

Exercice 22

Taper les formules suivantes :

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\int_0^1 -\frac{\ln(1-t)}{t} dt \approx 1,64493$$

$$\max_{\substack{x,y \in E \\ x \cdot y = 0}} \varphi(x)$$

4.13 Modules de congruences

Selon l'apparence voulue, il y a trois façon d'écrire les modules de congruence :

`$a \equiv b \pmod m$`

`$a \equiv b \pmod m$`

`$a \equiv b \pmod m$`

$$\begin{aligned} a &\equiv b \pmod m \\ a &\equiv b \pmod{m} \\ a &\equiv b \pmod{m} \end{aligned}$$

Il y a aussi la commande `\bmod` qu'on peut utiliser dans le contexte suivant

`$\gcd(n,m \bmod n)$`

$$\gcd(n, m \bmod n)$$

Exercice 23

Reproduire la formule suivante :

$$\sum_{n \equiv a \pmod p} \frac{1}{n} < +\infty.$$

4.14 Placer au-dessus ou en-dessous

Les commandes `\underset{en-dessous}{symbole}` et `\overset{au-dessus}{symbole}` permettent de placer du matériel arbitraire en-dessous ou au-dessus de n'importe quel symbole.

```
\[
\overset{\text{déf}}{=}
\]
\[
x \underset{x \to 0}{\in} o(x^2)
\]
```

$$\begin{aligned} & \overset{\text{déf}}{=} \\ x & \underset{x \rightarrow 0}{\in} o(x^2) \end{aligned}$$

4.15 Accents mathématiques

Pour mettre un accent sur une lettre seule on pourra utiliser l'un de ces commandes :

<code>\tilde{a}</code>	<code>\vec{a}</code>	<code>\hat{a}</code>	<code>\check{a}</code>	<code>\mathring{a}</code>	<code>\dot{a}</code>	<code>\ddot{a}</code>
\tilde{a}	\vec{a}	\hat{a}	\check{a}	\mathring{a}	\dot{a}	\ddot{a}
<code>\dddota</code>	<code>\ddddota</code>	<code>\acute{a}</code>	<code>\breve{a}</code>	<code>\grave{a}</code>	<code>\bar{a}</code>	
\dddot{a}	\ddddot{a}	\acute{a}	\breve{a}	\grave{a}	\bar{a}	

Il existe cependant des accents extensibles pour couvrir plusieurs lettres :

<code>\widetilde{abc}</code> \widetilde{abc}	<code>\widehat{abc}</code> \widehat{abc}	<code>\overline{abc}</code> \overline{abc}	<code>\underline{abc}</code> \underline{abc}
<code>\overbrace{abc}</code> \overbrace{abc}	<code>\underbrace{abc}</code> \underbrace{abc}	<code>\overrightarrow{abc}</code> \overrightarrow{abc}	<code>\underrightarrow{abc}</code> \underrightarrow{abc}
<code>\overleftarrow{abc}</code> \overleftarrow{abc}	<code>\underleftarrow{abc}</code> \underleftarrow{abc}	<code>\overleftrightarrow{abc}</code> \overleftrightarrow{abc}	<code>\underleftrightarrow{abc}</code> $\underleftrightarrow{abc}$

Pour des vecteurs (extensibles), mieux utiliser la commande `\vv{}` du package `esvect` :

`\vec{u}`, `\vv{u}`, `\vec{AB}`, `\vv{AB}`

\vec{u} , \vec{u} , \vec{AB} , \vec{AB}

Pour `\underbrace` il est possible de placer du matériel en-dessous en utilisant `_` et pour `\overbrace` du matériel au-dessus en utilisant `^`.

```
\[
\underbrace{x^3 + x^2 + x + 1}_{\to 0}
\qquad
\overbrace{x^3 + x^2 + x + 1}^{\to 0}
\]
```

$\underbrace{x^3 + x^2 + x + 1}_{\to 0}$ $\overbrace{x^3 + x^2 + x + 1}^{\to 0}$

Exercice 24

Écrire la formule suivante :

$$\overrightarrow{OM} = \underbrace{O + \vec{u}}_{\text{point+vecteur}}$$

4.16 Délimiteurs

COMMANDE	RÉSULTAT	COMMANDE	RÉSULTAT	COMMANDE	RÉSULTAT
<code>(</code>	$($	<code>\lvert</code>	$ $	<code>\lceil</code>	\lceil
<code>)</code>	$)$	<code>\rvert</code>	$ $	<code>\rceil</code>	\rceil
<code>[</code>	$[$	<code>\lVert</code>	$\ $	<code>\langle</code>	\langle
<code>]</code>	$]$	<code>\rVert</code>	$\ $	<code>\rangle</code>	\rangle
<code>\{</code>	$\{$	<code>\lfloor</code>	\lfloor	<code>/</code>	$/$
<code>\}</code>	$\}$	<code>\rfloor</code>	\rfloor	<code>\backslash</code>	\backslash

Il est important de comprendre que, même si `\lvert` et `\rvert` se ressemblent, ils ne peuvent pas être interchangeables : `\lvert` doit toujours être utilisé pour ouvrir et `\rvert` pour refermer. Par exemple, $|x|$ se tape `\lvert x \rvert`. Les seuls délimiteurs à n'être ni ouvrant ni fermant mais médian sont le slash `/` et l'anti-slash `\`.

Exercice 25

Taper les formules suivantes :

$$\|x\| = 1 \iff \langle x, x \rangle = 1$$

$$|\{1, 2, \dots, n\}| = n$$

$$\lfloor x^2 + \epsilon \rfloor = \lceil \sqrt{y} + \delta \rceil$$

Pour avoir des délimiteurs qui sont de la même taille que ce qu'ils entourent, il faut précéder le délimiteur ouvrant par `\left` et le délimiteur fermant par `\right`.

```
\[
\left(1+\frac{1}{n}\right)^n
\]
```

$$\left(1 + \frac{1}{n}\right)^n$$

S'il n'y a qu'un délimiteur, il faut mettre un délimiteur "vide" indiqué par un point :

```
$PV=nRT$ donc
\[
\left.\frac{\partial P}{\partial T}\right|_V=\frac{nR}{V}
\]
```

$$PV = nRT \text{ donc}$$

$$\left.\frac{\partial P}{\partial T}\right|_V = \frac{nR}{V}$$

Exercice 26

Taper les formules suivantes :

$$\left[\sum_{n=1}^N u_n\right]^2 = N^2 + N + 1$$

$$\left[1 + \left(\int_0^{\sqrt{2}} f\right)^2\right] = \gamma$$

big Big BBig etc, voir «courte intro»

On peut aussi utiliser `\middle` pour mettre un délimiteur médian au milieu d'un couple `\left` et `\right`, comme dans la définition d'un ensemble :

```
\[
\left\{ x \middle| x^2 < \frac{1}{2} \right\}
\]
```

$$\left\{ x \middle| x^2 < \frac{1}{2} \right\}$$

On note qu'ici la barre verticale est trop proche de ce qu'il y a autour d'elle et il faut rajouter à la main deux petits espaces avec la commande `\` :

```
\[
\left\{ x \ , \middle| \ , \ x^2 < \frac{1}{2} \right\}
\]
```

$$\left\{ x \ , \middle| \ , \ x^2 < \frac{1}{2} \right\}$$

Pour écrire plus simplement les ensembles, on peut utiliser le package `braket` : dans le **préambule** on ajoute

```
\usepackage{braket}
```

et dans le corps du document on utilise la commande `\Set{}`. L'exemple précédent s'écrit alors

```
\[
\Set{ x | x^2 < \frac{1}{2} }
\]
```

$$\left\{ x \middle| x^2 < \frac{1}{2} \right\}$$

4.17 Alphabets mathématiques

Voici un résumé des alphabets mathématiques disponibles après importation dans le **préambule** des packages `amssymb` et `amsfonts` (on remplacera la lettre *C* par une lettre majuscule quelconque et la lettre *x* par une lettre minuscule quelconque) :

COMMANDE	RÉSULTAT	COMMANDE	RÉSULTAT
<code>\mathbb{C}</code>	\mathbb{C}	<code>\mathrm{x}</code>	x
<code>\mathcal{C}</code>	\mathcal{C}	<code>\mathbf{x}</code>	\mathbf{x}
<code>\mathfrak{C}</code>	\mathfrak{C}	<code>\mathhtt{x}</code>	x
		<code>\mathsf{x}</code>	x
		<code>\mathit{PGL}</code>	PGL

À noter que le résultat obtenu ici avec la commande `\mathcal{C}` n'est pas le même que dans nos salles de TP (qui est obtenu avec la police par défaut Computer Modern du package `lmodern`). En effet, pour cet ouvrage j'utilise l'extension `fourier` qui modifie le résultat. Si l'on utilise la police par défaut et que l'on souhaite faire des majuscules manuscrites, on peut utiliser le package `{mathrsfs}` et la commande `\mathscr{C}`.

Ne pas confondre `\mathrm` et `\text` : le premier est pour mettre des maths en romain tandis que le second est pour insérer du texte dans une formule.

Attention : pour mettre en gras une lettre grecque, on ne peut pas utiliser `\mathbf{}` mais il faut utiliser `\boldsymbol{}` :

```
\[
\sigma \quad \mathbf{\sigma} \quad \boldsymbol{\sigma}
\]
```

σ σ σ

Exercice 27

Reproduire le texte suivant :

Soit $f: \mathbb{R} \rightarrow \mathbb{R}$ une fonction d'ensemble de définition \mathcal{D}_f et de courbe représentative \mathcal{C}_f .
Soit \mathcal{F} une famille libre de vecteurs et $\mathbf{x} \in \mathcal{F}$.

Exercice 28

Taper les formules suivantes :

$$\{a + ib \in \mathbb{C} \mid a < b\}$$

$$\mathcal{L}f = \int_a^b f dt$$

4.18 Tableaux mathématiques

Pour composer des tableaux mathématiques on utilise l'environnement `{array}` qui fonctionne de manière similaire à l'environnement `{tabular}` :

```
\[
\begin{array}{|c|c|}
\hline
f(x) & f'(x) \\
\hline
x^n & nx^{n-1} \\
\ln(x) & \frac{1}{x} \\
\hline
\end{array}
\]
```

$f(x)$	$f'(x)$
x^n	nx^{n-1}
$\ln(x)$	$\frac{1}{x}$

Exercice 29

Reproduire la formule suivante :

$$\left. \begin{array}{l} a \in \mathbb{C} \\ a \notin \mathbb{R} \end{array} \right\} \Rightarrow a \in \mathbb{C} \setminus \mathbb{R}$$

4.18.1 Distinction de cas

Pour définir une fonction par morceaux on peut utiliser l'environnement `{array}` mais le package `amsmath` met à disposition un environnement dédié, l'environnement `{cases}`, qui s'utilise de la façon suivante

```
\[
H(x) =
\begin{cases}
0 & \text{si } x < 0, \\
1 & \text{si } x \geq 0.
\end{cases}
\]
```

$$H(x) = \begin{cases} 0 & \text{si } x < 0, \\ 1 & \text{si } x \geq 0. \end{cases}$$

4.18.2 Matrices

On tape les matrices comme des tableaux, sauf qu'on n'utilise plus l'environnement `{array}`, mais les environnements `{matrix}`, `{pmatrix}`, etc.

```
\[
\begin{matrix} a & b \\ c & d \end{matrix}
\]
\begin{pmatrix} a & b \\ c & d \end{pmatrix}
\]
\begin{bmatrix} a & b \\ c & d \end{bmatrix}
\]
\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}
\]
\begin{vmatrix} a & b \\ c & d \end{vmatrix}
\]
\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}
\]
```

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

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

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

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

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

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

Les matrices peuvent avoir jusqu'à 10 colonnes; si jamais il y a besoin d'en avoir plus (par exemple, 15 colonnes), rajouter dans le **préambule**, après avoir chargé tous les packages, la ligne

```
\setcounter{MaxMatrixCols}{15}
```

Exercice 30

Reproduire la matrice suivante :

$$M = \begin{pmatrix} m_{1,1} & \dots & m_{1,n} \\ \vdots & \ddots & \vdots \\ m_{n,1} & \dots & m_{n,n} \end{pmatrix}$$

4.19 Formules sur plusieurs lignes et alignements

Dans une formule *displaystyle* l'«aller à la ligne» est interdit (car le changement de paragraphe est interdit en mode mathématique). On dispose alors d'environnements spécifiques pour écrire une formule sur plusieurs lignes : les environnements `{gather}`, `{multline}` et `{align}` du package `amsmath`.

- L'environnement `{gather}` (ou sa variante non numérotée `{gather*}`) est fait pour écrire plusieurs équations les unes en-dessous des autres et centrée sur chaque ligne :

```
\begin{gather}
A+B = C+D \\
A = C+D-B
\end{gather}
```

$$A + B = C + D \quad (4.3)$$

$$A = C + D - B \quad (4.4)$$

Ne jamais mettre de `\\` sur la dernière ligne d'un `{gather}`.

- L'environnement `{multline}` (ou sa variante non numérotée `{multline*}`) permet de découper une formule trop grande en plusieurs morceaux sans contrôle précis de l'alignement. Voici un exemple :

```
\begin{multline}
A+B+C+D+E+F+G+H+I+J \\
+A+B+C+D+E+F+G+H+I+J \\
+A+B+C+D+E+F+G+H+I+J \\
+A+B+C+D+E+F+G+H+I+J
\end{multline}
```

$$\begin{aligned} A + B + C + D + E + F + G + H + I + J \\ + A + B + C + D + E + F + G + H + I + J \\ + A + B + C + D + E + F + G + H + I + J \\ + A + B + C + D + E + F + G + H + I + J \end{aligned} \quad (4.5)$$

Ne jamais mettre de `\\` sur la dernière ligne d'un `{gather}`.

- L'environnement `{align}` (ou sa variante non numérotée `{align*}`) permet d'aligner plusieurs signes d'égalité à l'intérieur d'une même formule :

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

$$A + B = C + D \quad (4.6)$$

$$A = C + D - B \quad (4.7)$$

```
\begin{align*}
(x+y+z)^2 &= (x+y)^2+z^2+2(x+y)z \\
&= x^2+y^2+2xy+z^2+2(x+y)z \\
&= x^2+y^2+z^2+2xy+2xz+2yz
\end{align*}
```

$$\begin{aligned} (x + y + z)^2 &= (x + y)^2 + z^2 + 2(x + y)z \\ &= x^2 + y^2 + 2xy + z^2 + 2(x + y)z \\ &= x^2 + y^2 + z^2 + 2xy + 2xz + 2yz \end{aligned}$$

Ne jamais mettre de `\\` sur la dernière ligne d'un `{align}`.

On peut au besoin mettre plusieurs équations en colonne :

```

\begin{align*}
A & \& = B & \& E & \& = F & \& I & \& = J \\
& \& = C+C & \& & \& = G & \& & \& = K \\
& \& = D & \& & \& & \& & \&
\end{align*}

```

$$\begin{array}{lll}
 A = B & E = F & I = J \\
 = C + C & = G & = K \\
 = D & &
 \end{array}$$

Finalement, on peut vouloir numéroter uniquement certaines lignes d'une équation. Il y a alors la commande `\notag` qui permet de désactiver la numérotation sur une ligne. Voici un exemple avec une seule label :

```

\begin{align}
A & \& = B & \notag \\
& \& = C & \notag \\
& \& = D & \label{eq.no}
\end{align}

```

$$\begin{array}{ll}
 A = B \\
 = C \\
 = D
 \end{array} \tag{4.8}$$

Le package `amsmath` définit d'autres environnements, vous pouvez les utiliser (après avoir lu la documentation du package et compris dans quels cas il faut utiliser un environnement plutôt qu'un autre).⁴

Exercice 31

Reproduire les formules suivantes :

$$\begin{array}{l}
 f: \mathbb{R} \rightarrow \mathbb{R} \\
 x \mapsto x^2
 \end{array}$$

$$\begin{array}{l}
 g: \mathbb{R} \rightarrow \mathbb{R} \\
 x \mapsto \sqrt{x}
 \end{array}$$

et

$$\begin{aligned}
 \left| \int_a^b (f+g) \right| &= \left| \int_a^b f + \int_a^b g \right| \\
 &\leq \left| \int_a^b f \right| + \left| \int_a^b g \right| \\
 &\leq \int_a^b |f| + \int_a^b |g|
 \end{aligned}$$

4.20 Exercices de synthèse

Exercice 32

Reproduire les formules suivantes :

$$\sqrt{\sum_{n=0}^{+\infty} u_n} = \left(\int_a^b f dt \right)^2 + \gamma \times \frac{\pi^2}{6}$$

$$\lim_{x \rightarrow 0^+} f(x) = \sin \left| \frac{1 - \sqrt[2]{3}}{2} \right| \quad \text{mais} \quad f(0) = \ln^2(3)$$

⁴Attention : l'utilisation de l'environnement `eqnarray` pour mettre en forme une formule sur plusieurs lignes (comme on peut trouver dans certains vieux manuels), sera considéré comme une faute grave et sévèrement sanctionné. Cet environnement a plusieurs défauts bien illustrés dans ce document : [Avoid eqnarray!](http://tug.org/pracjourn/2006-4/madsen/) disponible à l'adresse <http://tug.org/pracjourn/2006-4/madsen/>.

$$\lim_{\substack{x \rightarrow 0 \\ x > 0}} \frac{[(\sin(x))^{\cos(x)} - (\cos(x))^{\sin(x)}]}{x^3}$$

$$\forall x \in \mathbb{R}, \quad f(x) = 0 \iff x^{1/3} + \ln(\tan(x)) = \omega$$

Exercice 33

En utilisant des commandes personnelles judicieusement choisies, taper les formules suivantes :

$$|\{x \in \mathbb{Z} \mid x^2 < 2\}| = \bigcup_{n \in \mathbb{N}^*} \left| \left\{ x \in \mathbb{Z} \mid x^2 < 2 - \frac{1}{n} \right\} \right|$$

$$\mathbb{S}^1 = \{x \in \mathbb{R}^2 \mid \|x\| = 1\}$$

$$\sum_{n=1}^{+\infty} \frac{1}{n^2 C_{2n}^n} = \frac{\zeta(2)}{3} = \frac{\pi^2}{18}$$

$$S(x) = \int_0^x \frac{\sin(x)}{x} dx \quad \text{avec} \quad \lim_{x \rightarrow +\infty} S(x) = \frac{\pi}{2}$$

4.21 Pour aller plus loin



- Pour plus d'information sur la rédaction des mathématiques avec \LaTeX on pourra consulter le fichier «Mathmode» écrit par H. VOSS disponible en pdf à l'adresse <https://texdoc.org/serve/mathtools/0>
- Pour tous les symboles disponibles en \LaTeX , voir «*The Comprehensive LATEX Symbol List*» disponible en pdf à l'adresse <https://www.ctan.org/pkg/comprehensive>
- Vous pouvez dessiner le symbole et detexify vous dira quel package charger et la commande pour obtenir ce symbole : <http://detexify.kirelabs.org/classify.html>

Faire attention au(x) package(s) nécessaire(s) à un symbole, tous les packages ne sont pas forcément compatibles avec notre préambule ni forcément installés.

Récapitulatif sur les caractères ambigus

Il y a un certain nombre de caractères ambigus dans un pdf qui sont générés par des commande \LaTeX différentes. Par exemple, dans les deux formules « $d|n$ » et « $|x|$ », la barre verticale a trois significations distinctes donc doit être tapée différemment à chaque fois pour obtenir un espacement correct.

- Le caractère |

SIGNIFICATION	EXEMPLE	CODE \LaTeX
Valeur absolue	$ -x $	$\$ \backslash lvert -x \backslash rvert \$$
Divise	$d n$	$\$ d \backslash mid n \$$
Tel que	$x \sin(x) = 0$	$\$ x \backslash mid \backslash sin(x)=0 \$$
Tel que (dans un ensemble)	$\{x \sin(x) = 0\}$	$\$ \backslash Set\{x \mid \backslash sin(x)=0\} \$$
Restreint à	$f _{\mathbb{N}}$	$\$ f _{\backslash N} \$$

- Le caractère ||

SIGNIFICATION	EXEMPLE	CODE \LaTeX
Norme	$\ -x\ $	$\$ \backslash lVert -x \backslash rVert \$$
Divise exactement	$p^2 n$	$\$ p^2 \backslash parallel n \$$
Parallèle à	$(AB) (CD)$	$\$ (AB) \backslash parallel (CD) \$$

4 Mathématiques

- Le caractère :

SIGNIFICATION	EXEMPLE	CODE L ^A T _E X
Deux points	$f: X \rightarrow Y$	<code>\$f\colon X\to Y\$</code>
	$\mathcal{C}: x^2 + y^2 = 1$	<code>\$\$\mathcal{C}\colon x^2+y^2=1\$</code>
Indice... dans	$[G: H]$	<code>\$\$[G:H]\$</code>
Tel que	$\{x: \sin(x) = 0\}$	<code>\$\$\Set{x : \sin(x)=0}\$</code>
Point projectif	$[x_1 : \dots : x_n]$	<code>\$\$[x_1 : \dots : x_n]\$</code>

5 Présentations vidéo-projetées

Beamer est une classe comme `scrartcl`, `scrreprt`... dont le but est de réaliser des présentations, c'est à dire une série de diapositives destinées à être projetées. Beamer présente plusieurs avantages sur d'autre logiciel comme Impress de la suite OpenOffice (LibreOffice) ou PowerPoint de Microsoft. Tout d'abord c'est du \LaTeX donc une qualité typographique irréprochable. De plus, tous ce qu'on a appris sur \LaTeX est utilisable dans les présentations. Le document de sortie est un pdf donc sera lisible sans déformation sur n'importe quel type de machine et avec n'importe quel OS (Linux, Windows, Mac, Android).

5.1 Introduction

Comme toutes les autres classes, beamer se charge par

```
\documentclass[options]{beamer}
```

Il faut savoir que beamer charge automatiquement les packages `hyperref`, `xcolor` et `enumerate`. Si on veut leurs adjoindre des options on doit procéder de la façon suivante :

```
\documentclass
[
hyperref={pdfpagemode=FullScreen,colorlinks=false},
xcolor={table,dvipsnames,svgnames}
]{beamer}
```

L'option `pdfpagemode=FullScreen` d'`hyperref` permet de lancer automatiquement en mode plein écran la présentation.

Il y a un bug dans la version 3.65 de beamer lorsqu'on passe des options à `xcolor`. L'instruction `\PassOptionsToPackage` contourne ce bug. Les anciennes versions ainsi que les suivantes aussi ont été corrigées.

```
\PassOptionsToPackage{table,dvipsnames,svgnames}{xcolor}
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}
```

Ratio

Les dimensions de la diapositive dans beamer sont de 128 mm sur 96 mm (ratio de 4 : 3). Il est cependant préférable de modifier ce comportement pour avoir un ratio plus adapté aux écrans d'aujourd'hui, à savoir 16 : 9. Pour cela, on utilise l'option de classe `aspectratio` :

```
\documentclass[aspectratio=169]{beamer}
```

D'autres valeurs possibles sont 1610 pour 16 : 10, 149 pour 14 : 9, 54 pour 5 : 4, 43 pour 4 : 3 et 32 pour 3 : 2.

5.2 Choix du thème et création de la première diapositive

Le but de ces notes est de réaliser le plus simplement possible une présentation et pas de montrer toutes les possibilités de beamer, on va donc charger un modèle. On trouvera une galerie des thèmes disponibles¹ sur ces sites

¹Par exemple parmi les thèmes complets on a :

— Pittsburgh, Rochester, Bergen, Boadilla, Madrid, AnnArbor, CambridgeUS,

5 Présentations vidéo-projetées

<https://hartwork.org/beamer-theme-matrix/>
<http://mcclnews.free.fr/latex/beamergalerie.php>

Chaque diapositive est créée avec l'environnement `{frame}`. Voici un exemple à la figure 5.1 (avec le thème Madrid) obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete} % une police qui va bien pour l'écran
\usepackage[latin1]{inputenc} % si windows
%\usepackage[utf8]{inputenc} % si linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usetheme{Madrid}

\title{Le titre de la présentation}
\author{G. \textsc{Faccanoni}}
\institute{IMATH-UTLN}

\begin{document}

\begin{frame}[plain]
\maketitle
\end{frame}

\begin{frame}
Le texte de ma diapo.
\end{frame}

\end{document}
```

À la figure 5.2 les mêmes diapositives avec un aspectratio 16:9.

À la figure 5.3 les mêmes diapositives avec un aspectratio 16:9 et le thème CambridgeUS.

-
- Montpellier, Antibes, JuanLesPins,
 - Goettingen, Hannover, Marburg, Berkeley, Paloalto
 - Singapore, Szeged, Berlin, Ilmenau, Dresden, Darmstadt, Frankfurt
 - Malmoe, Copenhagen, Luebeck, Warsaw

5.2 Choix du thème et création de la première diapositive

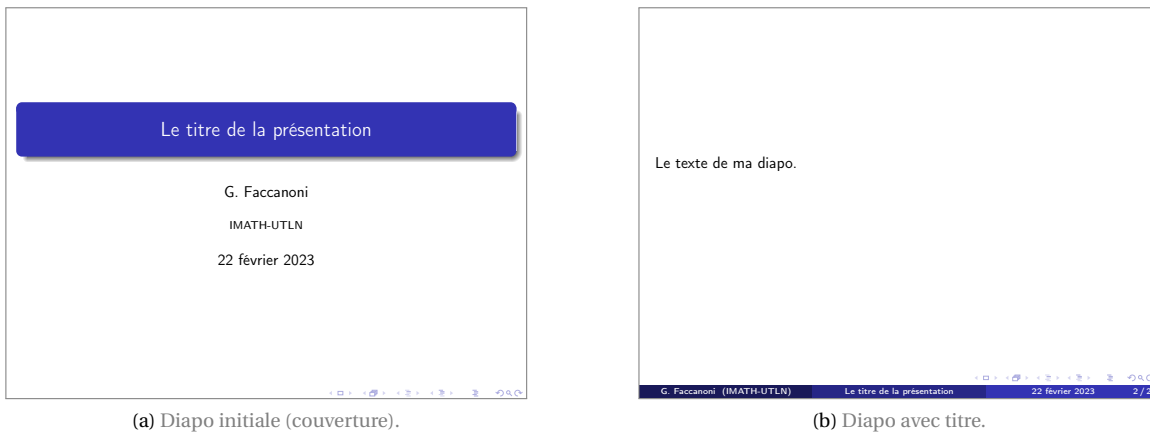


FIGURE 5.1 – Exemple de diapositives beamer (thème Madrid).



FIGURE 5.2 – Exemple de diapositives beamer (thème Madrid) avec un aspectratio 16:9.



FIGURE 5.3 – Exemple de diapositives beamer (thème CambridgeUS) avec un aspectratio 16:9.

5 Présentations vidéo-projetées

On peut utiliser le même sectionnement que sous \LaTeX pour regrouper plusieurs diapositives par section ou sous-section. Voici un exemple à la figure 5.4 obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usetheme{Antibes}

\begin{document}

\begin{frame}
\frametitle{Outline}
\tableofcontents
\end{frame}

\section{Titre de la section}

\subsection{Titre de la sous-section 1}

\begin{frame}
\frametitle{Titre à la diapo}
Première diapo de le sous section 1
\end{frame}

\begin{frame}
\frametitle{Titre à la diapo}
Deuxième diapo de le sous section 1
\end{frame}

\subsection{Titre de la sous-section 2}

\begin{frame}
\frametitle{Titre à la diapo}
Première diapo de le sous section 1
\end{frame}

\end{document}
```

Il **ne faut pas confondre section et titre de la diapositive**. Le sectionnement permet de regrouper plusieurs diapositives sous un même entête. Si le titre de la diapositive apparaît sur la diapositive elle-même, le sectionnement pour être visible doit être utilisé avec un thème affichant le sommaire (comme Hannover ou Antibes) ou afficher une diapositive avec la table de matière avant chaque début de section.

Attention : un environnement frame ne doit contenir ni sections ni sous-sections.

5.3 Les blocs

En plus de tous les objets \LaTeX (images, tableau, listes...) on peut insérer dans les diapositives des objets propres à beamer comme par exemple les block. Leur aspect dépend du thème choisi. Voici un exemple à la figure 5.5a avec le thème Warsaw obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}
```

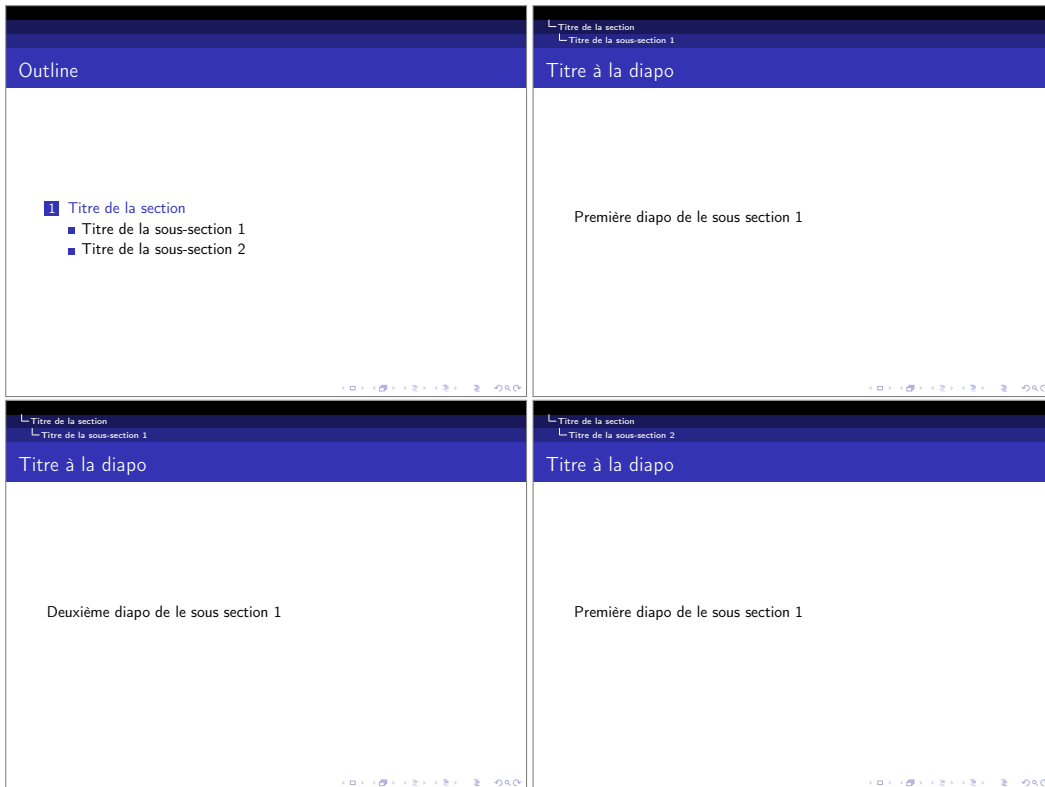


FIGURE 5.4 – Exemple de diapositives regroupées par sections

```

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usetheme{Warsaw}

\begin{document}

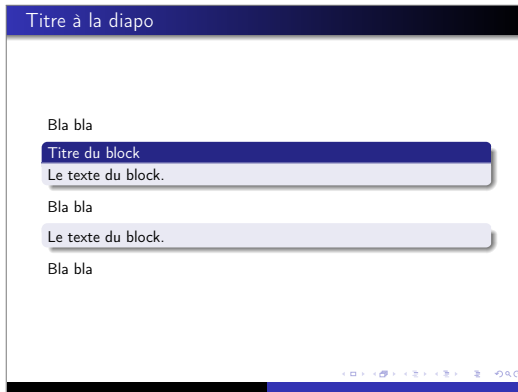
\begin{frame}
\frametitle{Titre à la diapo}
Bla bla
\begin{block}{Titre du block}
Le texte du block.
\end{block}
Bla bla
\begin{block}{}
Le texte du block.
\end{block}
Bla bla
\end{frame}

\end{document}

```

Il existe deux autres type de blocs, dont la seule différence réside dans la couleur : le bloc `alert` et le bloc `example`. Voici un exemple à la figure 5.5b avec le thème `Warsaw` obtenu avec le code

5 Présentations vidéo-projetées



(a) Blocs avec ou sans titre.



(b) Blocs de type alert et exemple.

FIGURE 5.5 – Exemple de diapositives avec blocs (thème Warsaw).

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

Bla bla

\begin{alertblock}{Titre du block}
Le texte du block.
\end{alertblock}

Bla bla

\begin{alertblock}{}
Le texte du block.
\end{alertblock}

Bla bla

\begin{exampleblock}{Titre du block}
Le texte du block.
\end{exampleblock}

Bla bla

\begin{exampleblock}{}
Le texte du block.
\end{exampleblock}
```

```
\end{frame}

\end{document}
```

Il existe également plusieurs type de blocs prédéfini : les définitions, les exemples, les démonstrations, les théorèmes. La traduction des titres de ces blocs n'est pas prise en compte par babel mais par le package translator qui est automatiquement chargé par beamer mais auquel il faut passer l'option French en mettant dans le **préambule** le deux instructions `\uselanguage{French}\languagepath{French}`. Voici un exemple à la figure 5.6 avec le thème Warsaw obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\usetheme{Warsaw}
\uselanguage{French}
\languagepath{French}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

Ela bla

\begin{definition}
Le texte de la définition.
\end{definition}

Ela bla

\begin{example}
Le texte de l'exemple.
\end{example}

Ela bla

\begin{theorem}
Le texte du théorème.
\end{theorem}

\begin{proof}
Le texte de la démonstration.
\end{proof}

\end{frame}

\end{document}
```

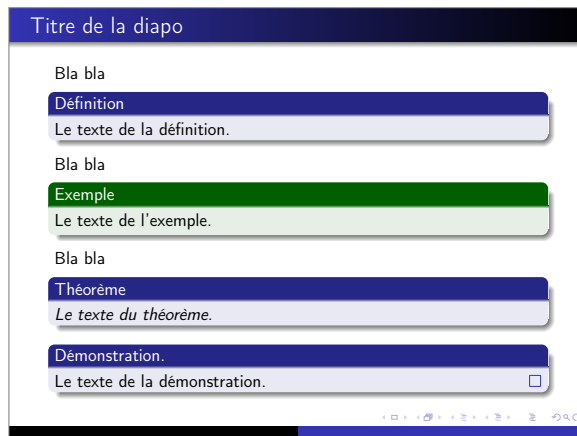


FIGURE 5.6 – Exemple de diapositives avec blocs prédéfinis (thème Warsaw).

5.4 Les listes

L'aspect des listes est modifié par beamer et dépend du thème employé. beamer chargeant automatiquement le package enumerate, on peut employer directement les options de ce dernier pour modifier les énumérations. Voici à la figure 5.7 plusieurs listes utilisant respectivement les environnements itemize, enumerate et description avec les valeurs par défaut et le thème Warsaw obtenu avec le code

```

\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Exemple avec \texttt{itemize}}

\begin{itemize}
\item item 1
  \begin{itemize}
\item item 1.1
\item item 1.2
  \begin{itemize}
\item item 1.2.1
\item item 1.2.2
  \end{itemize}
\end{itemize}
\end{itemize}
\item item 2
\end{itemize}

\end{frame}

```

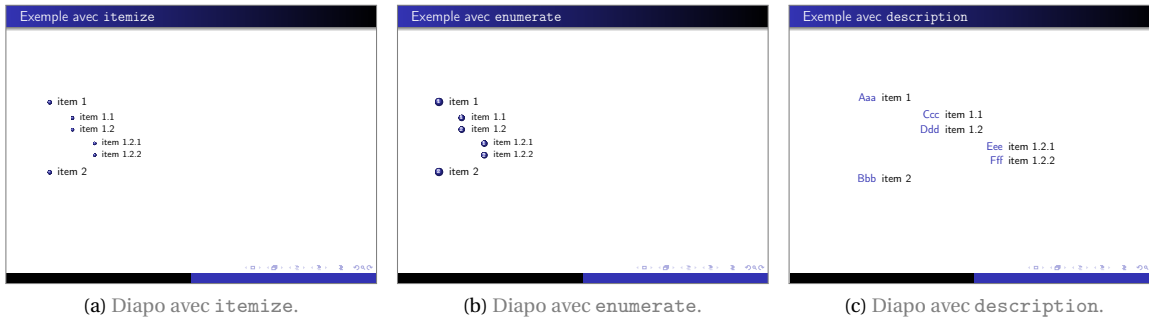



FIGURE 5.7 – Exemple de diapositive avec listes (thème Warsaw).

```

\begin{frame}
\frametitle{Exemple avec \texttt{enumerate}}

\begin{enumerate}
\item item 1
  \begin{enumerate}
\item item 1.1
\item item 1.2
  \begin{enumerate}
\item item 1.2.1
\item item 1.2.2
\end{enumerate}
\end{enumerate}
\end{enumerate}
\item item 2
\end{enumerate}

\end{frame}

\begin{frame}
\frametitle{Exemple avec \texttt{description}}

\begin{description}
\item[Aaa] item 1
  \begin{description}
\item[Ccc] item 1.1
\item[Ddd] item 1.2
  \begin{description}
\item[Eee] item 1.2.1
\item[Fff] item 1.2.2
\end{description}
\end{description}
\end{description}
\item[Bbb] item 2
\end{description}

\end{frame}

\end{document}

```

5.5 Ajouter des colonnes

On peut créer des colonnes de cette manière (attention au «s» de l'environnement externe columns) :

```
\begin{columns}
  \begin{column}{dimension de la première colonne}
  Le texte de la colonne
  \end{column}
  \begin{column}{dimension de la deuxième colonne}
  Le texte de la colonne
  \end{column}
\end{columns}
```

On utilise autant de `\begin{column}{dimension}... \end{column}` que l'on souhaite de colonnes. Pour les dimensions, mieux vaut utiliser des valeurs relatives comme `{0.5\textwidth}`. Voici un exemple à la figure 5.8 avec le thème Warsaw obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{marvosym}

\usetheme{Warsaw}

\begin{document}

\begin{frame}
\frametitle{Exemple avec 2 colonnes}

On peut écrire avant \dots
\bigskip

\begin{columns}

\begin{column}{0.4\textwidth}
\includegraphics[width=\textwidth]{hippopotense}
\end{column}

\begin{column}{0.6\textwidth}
\begin{itemize}
\item bla
\item bla
\end{itemize}
\end{column}

\end{columns}
\bigskip

\dots et après \Smiley

\end{frame}

\end{document}
```

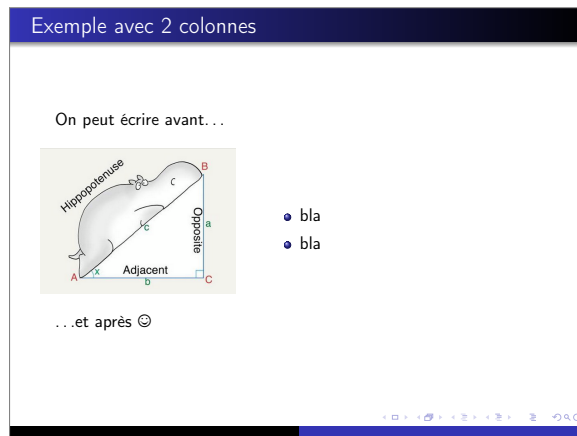


FIGURE 5.8 – Exemple de diapositives avec deux colonnes (thème Warsaw).

5.6 Les images

Pour placer des images on utilisera `\includegraphics` (éventuellement dans un environnement `center` et/ou dans une colonne). Il est conseillé d'utiliser des dimensions relatives comme des pourcentage de `\textwidth` ou `\paperwidth`. Étant donné qu'il ne s'agit pas de flottants, ça n'a aucun sens d'utiliser l'environnement `figure`.

5.7 La barre de navigation

beamer place dans le coin en bas à droite une barre de navigation qui permet de se déplacer dans la présentation avec la souris mais d'habitude on préfère naviguer avec le clavier; pour enlever cette barre il suffit de mettre dans le **préambule**

```
\setbeamertemplate{navigation symbols}{}
```

Pour modifier la barre de navigation pour n'en garder que quelques boutons voir la documentation du package beamer.

Conseil : si le thème choisi ne montre pas le numéro de la diapositive, on pourra utiliser la place de la barre de navigation pour le faire apparaître :

```
\setbeamertemplate{navigation symbols}{\insertframenumber/\inserttotalframenumber}
```

5.8 Mettre en avant des portions de texte

Pour souligner un ou des mots qui paraissent importants on peut toujours utiliser les commandes \LaTeX `\emph{}` ou `\textbf{}` pour mettre en emphase ou en gras. Toutefois, beamer possède deux autres instructions mieux adaptés (et qui dépendent du thème et de la couleur du thème choisie) : les commandes `\structure{}` et `\alert{}` ainsi que les deux environnements du même type `\begin{structureenv}...` `\end{structureenv}` et `\begin{alertenv}...` `\end{alertenv}`. Voici un exemple à la figure 5.9 avec le thème Warsaw obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
```

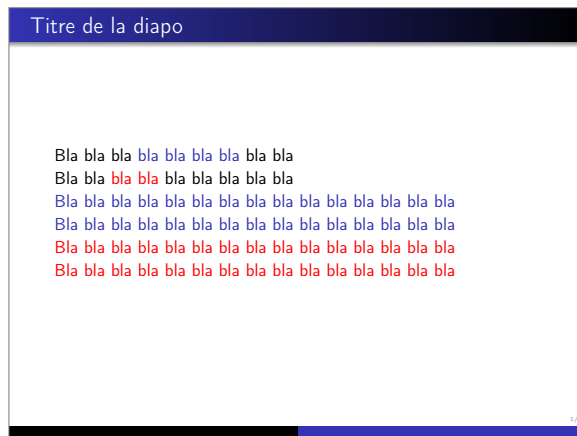


FIGURE 5.9 – Exemple de diapositives avec des mots mis en évidence (thème Warsaw).

```
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\usetheme{Warsaw}
\setbeamertemplate{navigation symbols}{\insertframenumber/\inserttotalframenumber}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}
Bla bla bla \structure{bla bla bla bla} bla bla

Bla bla \alert{bla bla} bla bla bla bla bla

\begin{structureenv}
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
\end{structureenv}

\begin{alertenv}
Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla

Bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla
\end{alertenv}

\end{frame}

\end{document}
```

5.9 only, uncover, visible etc.

Un frame est constitué d'une suite d'une ou plusieurs *layer* qui peuvent être superposées. Certaines commandes supportent des spécifications de superposition : une spécification est le numéro (ou la liste des numéros ou une plage de numéros) des layers auxquelles la commande s'applique. Les numéros sont

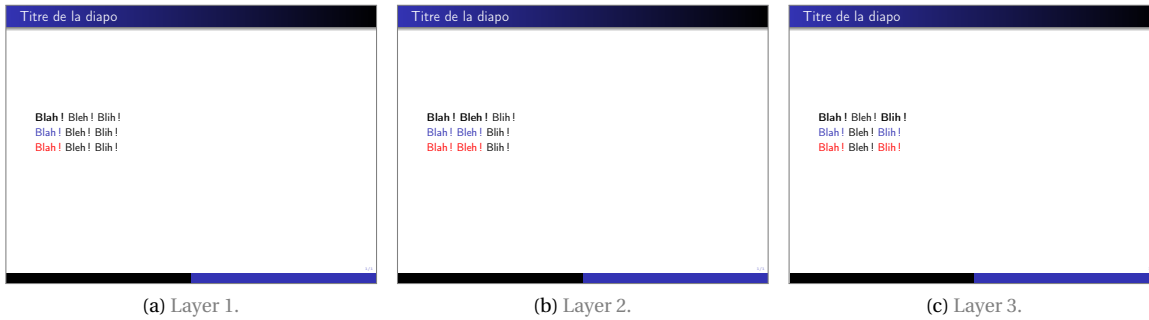


FIGURE 5.10 – Exemple d’un même frame avec trois layers.

indiqués entre < et > après le nom de la commande.

Voici à la figure 5.10 un exemple du même frame avec trois layers obtenu avec le code

```

\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\usetheme{Warsaw}
\setbeamertemplate{navigation symbols}{\insertframenumbers/\inserttotalframenumbers}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\textbf{Blah!}
\textbf<2>{Bleh!}
\textbf<3>{Blih!}

\structure{Blah!}
\structure<2>{Bleh!}
\structure<3>{Blih!}

\alert{Blah!}
\alert<2>{Bleh!}
\alert<3>{Blih!}

\end{frame}

\end{document}

```

Exemples des différents types de spécifications :

- <1> : seul le layer 1 du frame est concerné par la commande
- <1,2,4> : liste de layers du frame concernés par la commande
- <1-4> : plage de layers du frame concernés par la commande
- <3-> : les layers du frame concernés par la commande sont le 3^e et tous les suivants

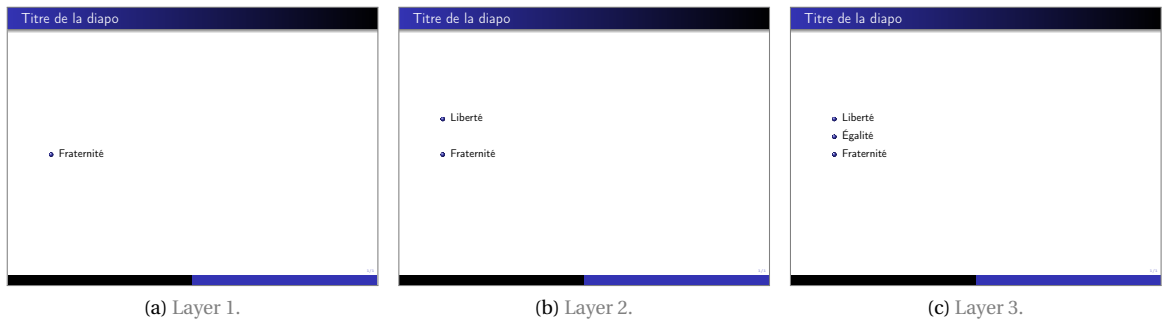


FIGURE 5.11 – Exemple d’un même frame avec trois layers.

- <-3> : les layers du frame concernés par la commande sont tous ceux jusqu’au 3^e inclus
- <-3,5,8-> : on peut combiner les écritures.

Commandes sujettes à spécifications :

- Commandes sur les polices : `\textbf`, `\textit`, `\textsl`, `\textrm`, `\textsf`
- Commande d’itération : `\item`
- Commandes de changement de couleur : `\color`, `\alert`, `\structure`
- Commandes et environnements propres à Beamer : `\alert` et `\structure`

Voici à la figure 5.11 un exemple du même frame avec trois layers obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\usetheme{Warsaw}
\setbeamertemplate[navigation symbols]{\insertframenumbar/\inserttotalframenumbar}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\begin{itemize}
\item<2-> Liberté
\item<3-> Égalité
\item<1-> Fraternité
\end{itemize}

\end{frame}

\end{document}
```

Voici à la figure 5.12 un exemple du même frame avec trois layers obtenu avec le code

```
\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}
```

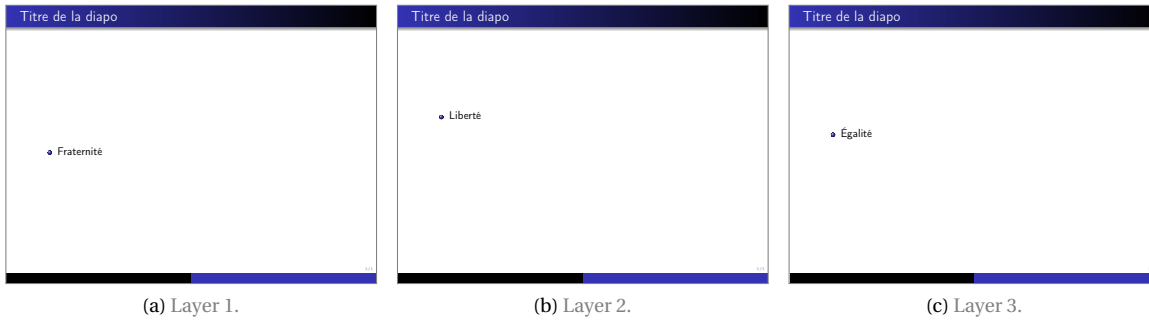


FIGURE 5.12 – Exemple d’un même frame avec trois layers.

```

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\usetheme{Warsaw}
\setbeamertemplate{navigation symbols}{\insertframenumber/\inserttotalframenumber}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\begin{itemize}
\item<2> Liberté
\item<3> Égalité
\item<1> Fraternité
\end{itemize}

\end{frame}

\end{document}

```

Commandes spécifiques :

- `\visible<plage>{texte}` : texte visible seulement sur plage
- `\uncover<plage>{texte}` : texte visible sur plage et en filigrane sur les autres layers (pour cela, ajoutera par exemple `\setbeamercovered{transparent=30}` dans le **préambule**)
- `\invisible<plage>{texte}` : contraire de `\visible`
- `\only<plage>{texte}` : comme `\visible` mais ne réserve pas d'espace à ce qui est invisible

Voici à la figure 5.13 un exemple du même frame avec trois layers obtenu avec le code

```

\documentclass[hyperref={pdfpagemode=FullScreen,colorlinks=false}]{beamer}

\usepackage{concrete}
\usepackage[latin1]{inputenc} % sous windows
%\usepackage[utf8]{inputenc} % sous linux
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

```

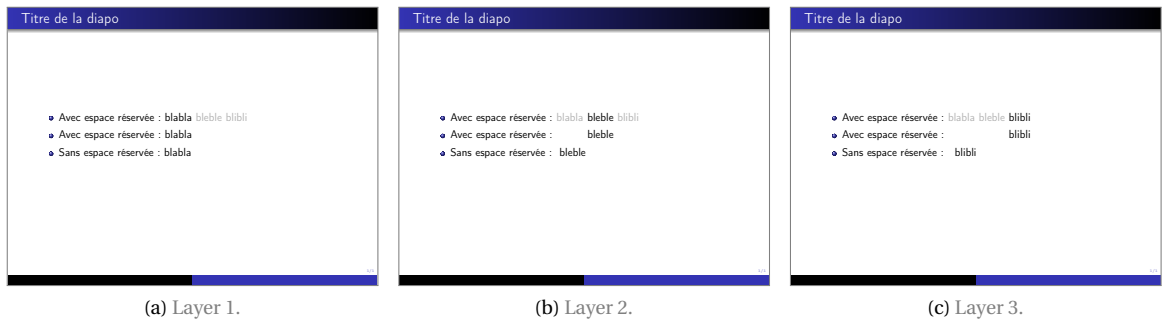


FIGURE 5.13 – Exemple d'un même frame avec trois layers.

```

\usetheme{Warsaw}
\setbeamercovered{transparent=30}
\setbeamertemplate{navigation symbols}{\insertframenum/\inserttotalframenumber}

\begin{document}

\begin{frame}
\frametitle{Titre de la diapo}

\begin{itemize}
\item Avec espace réservée: \uncover<1>{blabla} \uncover<2>{bleble} \uncover<3>{bllibli}
\item Avec espace réservée: \visible<1>{blabla} \visible<2>{bleble} \visible<3>{
bllibli}
\item Sans espace réservée: \only<1>{blabla} \only<2>{bleble} \only<3>{bllibli}
\end{itemize}

\end{frame}

\end{document}

```



5.10 Pour aller plus loin

- La documentation du package beamer disponible à l'adresse <https://www.ctan.org/pkg/beamer>
- Quelques conseils pour ne pas rater sa présentation orale (de N. Seguin) : <http://seguin.perso.math.cnrs.fr/conseils.html>
- et en anglais <https://people.inf.ethz.ch/markusp/teaching/guides/guide-presentations-new.pdf>

6 Compléments

6.1 Dessiner avec TikZ

Dans beaucoup d'ouvrages, les figures et les diagrammes souffrent de défauts rédhibitoires : traits trop épais, flèches dans un style complètement différent de celles du reste du document, polices de caractères différentes de celles du document, pixellisation, etc. Il y a plusieurs façon de faire des figures (mathématiques ou autre) avec \LaTeX , et un des package les plus puissants est TikZ. Voici deux exemples aux figures 6.1 et 6.2.

Pour apprendre à utiliser ce très puissante package on pourra consulter :

- un manuel pour débiter <http://math.et.info.free.fr/TikZ/>
- le manuel complet (1165 pages) décrivant toutes les fonctions disponibles
<http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>
- la galerie d'exemples en ligne <http://www.texample.net/tikz/examples/all/>

```

\begin{tikzpicture}[scale=1.7]
\shade[top color=blue,bottom color=gray!50]
(0,0) parabola (1.5,2.25) |- (0,0);
\draw (1.05cm,2pt)
node[above] {$\int_0^{3/2}x^2\mathrm{d}x$};
\draw[help lines] (0,0) grid (2.5,3.5)
[step=0.25cm] (1,2) grid +(1,1);
\draw[->] (-0.2,0) -- (2.5,0) node[below] {$x$};
\draw[->] (0,-0.2) -- (0,3.5) node[left] {$f(x)$};
\foreach \x/\xtext in {1/1, 1.5/\frac{3}{2}, 2/2}
\draw[shift={(\x,0)}] (0pt,2pt)
-- (0pt,-2pt) node[below] {$\xtext$};
\foreach \y/\ytext in {1/1, 2/2, 2.25/\frac{5}{4},
3/3}
\draw[shift={(0,\y)}] (2pt,0pt)
-- (-2pt,0pt) node[left] {$\ytext$};
\draw (-.5,.25) parabola bend (0,0) (2,4)
node[below right] {$x^2$};
\end{tikzpicture}

```

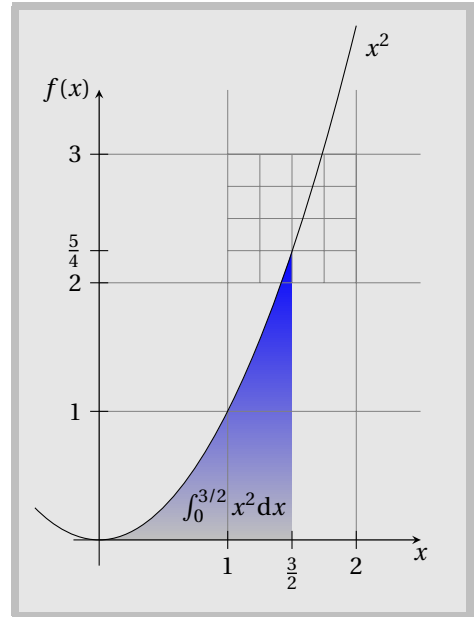


FIGURE 6.1 – Exemple de figures réalisée avec TikZ

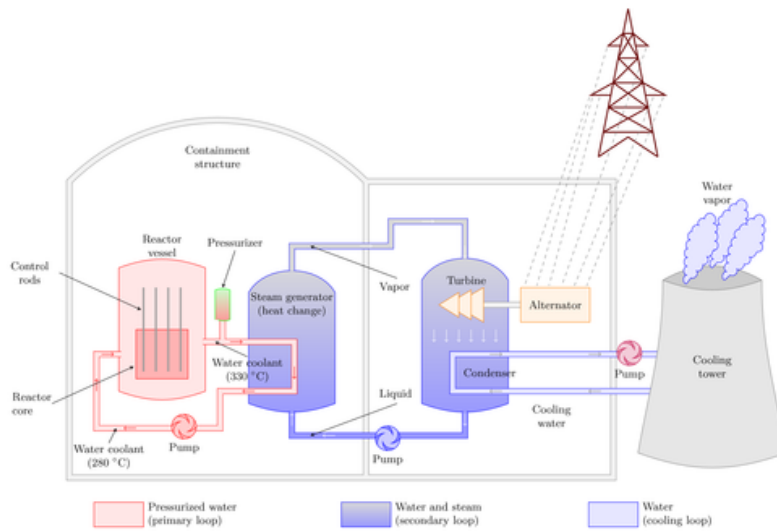


FIGURE 6.2 – <https://texample.net/tikz/examples/pressurized-water-reactor/>