

 eBook Gratuit

APPRENEZ

latex

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#latex

Table des matières

À propos.....	1
Chapitre 1: Commencer avec le latex.....	2
Remarques.....	2
Latex.....	2
Qu'est-ce que LaTeX?.....	2
Versions.....	2
Exemples.....	2
Installation et configuration.....	2
Installation.....	3
Windows (TeXLive).....	3
Windows (MiKTeX).....	3
Mac OS X (TeXLive).....	3
Linux (TeXLive).....	3
Utiliser les gestionnaires de paquets.....	3
Installation en amont.....	4
Test d'installation.....	4
Editeurs LaTeX.....	5
Chapitre 2: Accéder à la documentation des packages LaTeX.....	7
Exemples.....	7
CTAN.....	7
TeX Live - texdoc.....	10
Chapitre 3: Ajouter une citation.....	11
Exemples.....	11
Ajouter une citation au document LaTeX déjà existant.....	11
Chapitre 4: Classes de document.....	12
Syntaxe.....	12
Remarques.....	12
Exemples.....	12
Article.....	12

Quand utiliser la classe d'article?	12
Quelles sont les spécificités de cette classe?	12
Exemple simple	12
Beamer.....	13
Quand utiliser la classe beamer?	13
Quelles sont les spécificités de cette classe?	13
Exemple simple	13
Définition de la classe de document.....	13
Chapitre 5: Composition mathématique	14
Introduction.....	14
Syntaxe.....	14
Remarques.....	14
Exemples.....	14
Équations de base.....	14
Trouver des symboles.....	15
Packages disponibles pour utilisation.....	16
Bonnes Commandes à Savoir.....	17
Créer de nouveaux symboles.....	18
Matrices.....	19
Chapitre 6: Compteurs, si instructions et boucles avec latex	20
Exemples.....	20
Compteur déclaration, initialisation et impression en pdf.....	20
Opérations avec des compteurs.....	20
Si déclarations.....	22
Boucles - répéter des choses.....	23
Utiliser des boucles dans Tikz.....	24
Chapitre 7: Création d'affiches à l'aide d'un beamer	26
Introduction.....	26
Exemples.....	26
Orientation et taille.....	26
Plan de base d'une affiche de beamer.....	26

Exemple complet d'affiche beamer	28
Chapitre 8: Créer une bibliographie	30
Syntaxe	30
Paramètres	30
Exemples	30
Bibliographie de base sans paquets (formatage manuel)	30
Bibliographie de base avec biber	31
Chapitre 9: Définition de macros	33
Syntaxe	33
Paramètres	33
Exemples	33
Définition de base des macros	33
Définir une nouvelle commande de base	33
Définir une nouvelle commande avec des arguments	33
Redéfinir une commande existante	34
Chapitre 10: Dessiner des graphiques	35
Exemples	35
TikZ - Mise en page manuelle	35
TikZ - Spécifications graphiques	35
TikZ - Dessin graphique algorithmique	37
Diagramme de transition d'état d'une chaîne de Markov	38
Chapitre 11: En-tête et pied de page	40
Exemples	40
Utiliser des paquets fancyhdr et titleps	40
Numéro de page comme CurrPage / TotalPages dans le pied de page	41
Chapitre 12: Formatage du texte	43
Exemples	43
Emphase le texte	43
Frappe à travers le texte	43
Texte en gras	43
Chapitre 13: Gravure de partitions	44

Exemples.....	44
LilyPond.....	44
Chapitre 14: les tables.....	47
Exemples.....	47
L'environnement tabulaire.....	47
Tableau de coloriage.....	48
Chapitre 15: Outils de construction.....	52
Exemples.....	52
Arara.....	52
Chapitre 16: Pages de titres.....	54
Remarques.....	54
Exemples.....	54
Page de titre du rapport standard.....	54
Chapitre 17: Présentation avec package beamer.....	55
Paramètres.....	55
Remarques.....	55
Exemples.....	55
Diapositive de titre simple d'un auteur.....	55
Diapositive de titre d'auteur et d'affiliation multiple.....	56
Crédits.....	58

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [latex](#)

It is an unofficial and free latex ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official latex.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Commencer avec le latex

Remarques

Latex

Qu'est-ce que LaTeX?

LaTeX (prononcé *lay-tech* ou *lah-tekh*) est un langage de balisage pour la composition de documents similaire à celui de HTML pour les sites Web.

LaTeX présente des avantages sur les éditeurs WYSIWYG tels que Microsoft Word, car avec LaTeX, vous fournissez le contenu et LaTeX se charge de la mise en page. La séparation du contenu de la composition aboutit à des documents formatés de manière cohérente et élégante. En outre, le balisage LaTeX étant au format texte brut (contrairement aux formats de fichiers plus complexes produits par les éditeurs WYSIWYG, tels que `.docx`), les fichiers LaTeX sont légers et peuvent être facilement conservés sous le [contrôle des versions](#).

Les documents LaTeX sont généralement compilés dans des fichiers PDF afin que la cohérence de la présentation soit conservée entre les différents lecteurs et pour l'impression.

LaTeX est particulièrement populaire dans la rédaction académique en raison de son support important pour les équations de composition, les chiffres et les tableaux de références croisées et les citations et bibliographies.

Versions

Version	Date de sortie
LaTeX 2.09	1985-09-01
LaTeX 2e	1994-06-01

Exemples

Installation et configuration

Vous pouvez choisir entre les principales distributions de LaTeX:

- [TeX Live](#) (Windows, Linux et OS X), la distribution standard multiplateforme.
- [MacTeX](#) (Mac) Une version packagée de TeX Live conçue pour OS X avec des outils spécifiques à Mac
- [MiKTeX](#) (Windows) Une distribution séparée entièrement

Toutes les distributions sont plus ou moins équivalentes dans un monde idéal. TeX Live a l'avantage d'être disponible sur toutes les plates-formes et bénéficie ainsi d'un meilleur support communautaire. MiKTeX peut tirer parti des fonctionnalités spécifiques à Windows. Pour des raisons de licence, MiKTeX distribuera également quelques paquets qui ne seront pas disponibles pour TeX Live.

Dans tous les cas, l'installation complète est recommandée. Plus précisément, l'utilisation de la fonctionnalité de téléchargement à la demande de MiKTeX suspend / écrase de nombreux éditeurs.

Installation

Windows (TeXLive)

1. Téléchargez le fichier `install-tl-windows.exe` le plus récent depuis leur [site Web](#) .
2. Exécutez `install-tl-windows.exe` et suivez les instructions.

Windows (MiKTeX)

1. Téléchargez le programme d'installation MiKTeX le plus récent depuis leur [site Web](#) .
2. Exécutez le programme d'installation et suivez les instructions.

Mac OS X (TeXLive)

1. Téléchargez le plus récent MacTeX depuis leur [site Web](#) .
2. Exécutez `MacTeX.pkg` et suivez les instructions.

Linux (TeXLive)

Les utilisateurs de Linux ont deux options:

1. Installer via le gestionnaire de paquets de votre distribution (généralement plusieurs versions)
2. Installer depuis l'amont (publié annuellement, mis à jour souvent)

Utiliser les gestionnaires de paquets

- Arch Linux: `pacman -S texlive-most`
- Debian / Ubuntu / Mint: `apt-get install texlive-full`
- Fedora: `yum install texlive`

Notez que l'utilisation de cette méthode signifie que vous serez dépendant du responsable de ce package pour la distribution des mises à jour. Ces paquets seront souvent plusieurs versions de la distribution la plus récente, ce qui signifie souvent que des mises à jour critiques seront

manquantes. Il est presque toujours préférable de l'installer en amont. Notez également que le gestionnaire de paquets de la distribution ne reconnaîtra probablement pas l'installation directe et pourrait essayer de l'installer en installant d'autres packages de support associés.

Installation en amont

1. Téléchargez le plus récent `install-tl-unx.tar.gz` de `install-tl-unx.tar.gz` sur leur [site Web](#) .
2. Extrayez les fichiers de l'archive avec `tar -zxvf install-tl-unx.tar.gz` .
3. Passez dans le dossier téléchargé avec `cd install-tl-unx` .
4. Exécutez `./install-tl` et suivez les instructions.

TeXLive devrait maintenant être installé sous `/usr/local/texlive/YEAR/` , où `YEAR` est l'année à quatre chiffres (par exemple `2016`). De cette façon, il est possible d'avoir plusieurs versions de TeXLive les unes à côté des autres et de basculer entre elles en modifiant votre variable `PATH`.

Ouvrez ce dossier et vérifiez le dossier `bin` . Il devrait contenir un sous-dossier, qui (selon votre plate-forme) sera quelque chose comme `i386-linux` ou `x86_64-linux` .

5. Ajoutez le dossier binaire TeX Live à votre chemin avec

```
EXPORT PATH=/usr/local/texlive/YEAR/bin/PLATFORM:$PATH
```

où `YEAR` est l'année à quatre chiffres (par exemple `2016`) et `PLATFORM` est votre plate-forme (par exemple, `x86_64-linux`).

Test d'installation

L'installation de LaTeX est maintenant terminée. Pour le tester, créez un nouveau fichier avec votre éditeur de texte préféré, nommez-le `test.tex` et ajoutez le contenu suivant:

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

Maintenant, ouvrez la console ou le terminal, accédez au dossier où vous avez enregistré `test.tex` et exécutez

```
pdflatex test
```

(Notez que votre éditeur peut avoir des installations pour exécuter cela pour vous.)

Cela crée plusieurs nouveaux fichiers, y compris `test.pdf` . Ceci est le document de sortie et ressemble à ceci:

Hello World!

- L'éditeur [Vim](#) avec le plugin [LaTeX-suite](#) .
- [Texmaker](#) - un IDE LaTeX spécialisé.
- [TeXstudio](#) - un autre IDE LaTeX.
- [TeXworks](#) - un autre IDE LaTeX.

Bien que les utilisateurs expérimentés d'Emacs ou de Vim puissent vouloir s'en tenir à leur éditeur (dont les plug-ins fournissent une multitude de fonctionnalités indisponibles ailleurs), un IDE spécialisé pourrait être plus facile à installer / à utiliser pour les débutants. Les trois derniers sur la liste ont une fonction de prévisualisation où l'on peut voir les résultats de la compilation du document.

En outre, il existe des outils LaTeX en ligne qui peuvent être utiles aux débutants ou aux personnes qui doivent collaborer, par exemple [ShareLaTeX](#) et [Overleaf](#) .

Lire Commencer avec le latex en ligne: <https://riptutorial.com/fr/latex/topic/2269/commencer-avec-le-latex>

Chapitre 2: Accéder à la documentation des packages LaTeX

Exemples

CTAN

Le [Comprehensive TeX Archive Network](#) (CTAN) est en fait celui, le référentiel complet des packages LaTeX. La plupart, sinon tous les paquets de qualité (et plus) sont là, et tous les bons incluent la documentation.

1. Entrez le nom du package dans la barre de recherche.



2. Sélectionnez le package dans la liste.

Search biblatex

The search found 16 of 42 hits in 10ms.

1 2 3 Next

Package biblatex

Sophisticated Bibliographies in L^AT_EX

[/pkg/biblatex](#)

Last modified: 2016-05-15 17:33

Topic biblatex

BibL^AT_EX bibliography support

[/topic/biblatex](#)

Last modified: 2016-07-04 18:34

Package biblatex-jura

BibL^AT_EX stylefiles for German legal literature

[/pkg/biblatex-jura](#)

Last modified: 2015-08-03 06:36

Package biblatex-bwl

BibL^AT_EX citations for FU Berlin

[/pkg/biblatex-bwl](#)

Last modified: 2015-08-03 06:36

Package biblatex-nejm

BibL^AT_EX style for the New England Journal of Medicine (NEJM)

[/pkg/biblatex-nejm](#)

Last modified: 2015-08-03 06:36

3. Accédez aux documents de documentation.

BibL^AT_EX – Sophisticated Bibliographies in L^AT_EX

BibL^AT_EX is a complete reimplementaion of the bibliographic facilities provided by L^AT_EX. Formatting of the bibliography is entirely controlled by L^AT_EX macros, and a working knowledge of L^AT_EX should be sufficient to design new bibliography and citation styles. BibL^AT_EX uses its own data backend program called “[biber](#)” to read and process the bibliographic data. With [biber](#), BibL^AT_EX has many features rivalling or surpassing other bibliography systems. To mention a few:

- Full Unicode support
- Highly customisable sorting using the Unicode Collation Algorithm + CLDR tailoring
- Highly customisable bibliography labels
- Complex macro-based on-the-fly data modification without changing your data sources
- A tool mode for transforming bibliographic data sources
- Multiple bibliographies and lists of bibliographic information in the same document with different sorting
- Highly customisable data source inheritance rules
- Polyglossia and babel support for automatic language switching for bibliographic entries and citations
- Automatic bibliography data recoding (UTF-8 -> latin1, L^AT_EX macros -> UTF-8 etc)
- Remote data sources
- Highly sophisticated automatic name and name list disambiguation system
- Highly customisable data model so users can define their own bibliographic data types
- Validation of bibliographic data against a data model
- Subdivided and/or filtered bibliographies, bibliographies per chapter, section etc.

Apart from the features unique to BibL^AT_EX, the package also incorporates core features of the following packages: [babelbib](#), [bibtopic](#), [bibunits](#), [chapterbib](#), [cite](#), [inlinebib](#), [mcite](#) and [mciteplus](#), [mlbib](#), [multibib](#), [splitbib](#).

Sources	/macros/latex/contrib/biblatex
Documentation	 Readme   Package documentation (English)   Release notes for current version 
Version	3.4
License	The L^AT_EX Project Public License 1.3
Copyright	2012–2016 Philipp Lehman, Joseph Wright, Audrey Boruvka, Philip Kime 2006–2012 Philipp Lehman
Maintainer	Philipp Lehman (inactive) Philip Kime
Contained in	T_EX Live as biblatex MiK_TE_X as biblatex
Topics	BibL^AT_EX bibliography support bibliography processor



[Download](#) the contents of this package in one zip archive (17.2M).

Important: CTAN détient les versions les plus récentes. Si votre installation est obsolète, la documentation ne correspondra pas! Dans ce cas, reportez-vous aux documents de

documentation livrés avec votre distribution LaTeX.

TeX Live - texdoc

Si vous utilisez la distribution [TeX Live](#), vous pouvez utiliser le programme de ligne de commande `texdoc` . Par exemple,

```
texdoc biblatex
```

va ouvrir la documentation du paquet `biblatex` .

Ou, si vous ne maîtrisez pas la ligne de commande, vous pouvez trouver la même chose en ligne à l' [adresse http://www.texdoc.net/](http://www.texdoc.net/)

Lire [Accéder à la documentation des packages LaTeX en ligne:](#)

<https://riptutorial.com/fr/latex/topic/5820/acceder-a-la-documentation-des-packages-latex>

Chapitre 3: Ajouter une citation

Exemples

Ajouter une citation au document LaTeX déjà existant

À la fin du document, ajoutez ce qui suit:

`\bibliographystyle { style }`

`\bibliography { emplacement du fichier }`

Créez un fichier avec l'extension *.bib* et enregistrez la citation comme suit:

```
@inproceedings{citation_name,  
  title={Paper Title},  
  author={List Authors},  
  pages={45--48},  
  year={2013},  
  organization={organization name}  
}
```

Pour citer, utilisez ce qui suit: **`\citet { citation_name }`**

Lire Ajouter une citation en ligne: <https://riptutorial.com/fr/latex/topic/8357/ajouter-une-citation>

Chapitre 4: Classes de document

Syntaxe

- `\documentclass {...}`

Remarques

Ce sujet vise à expliquer les différents types de documents et leurs spécificités.

Une bonne façon de l'organiser serait 1 exemple par type

Exemples

Article

```
\documentclass{article}
```

Quand utiliser la classe d'article?

Pour des articles dans des revues scientifiques, des présentations, des rapports courts, de la documentation de programme, des invitations, ... ¹

Quelles sont les spécificités de cette classe?

Un article ne contient pas de chapitres ou de parties. Il peut être divisé en sections, sous-sections et paragraphes, etc.

Par défaut, le titre est affiché en haut de la première page et non sur une page de titre distincte.

Exemple simple

```
\documentclass{article}

\title{Hello world}
\author{Me    }
\date{\today}

\begin{document}

\maketitle

Hello, World!
```

```
\end{document}
```

Beamer

```
\documentclass{beamer}
```

Quand utiliser la classe beamer?

Pour les diapositives de présentation.

Quelles sont les spécificités de cette classe?

La sortie est orientée paysage. Le document est séparé en "frames" (slides).

Exemple simple

L'exemple suivant a été adapté de: sharelatex.com/learn/Beamer

```
\documentclass{beamer}

\usepackage[utf8]{inputenc}

\title{Sample title}
\author{Me}
\date{\today}

\begin{document}

\frame{\titlepage}

\begin{frame}
\frametitle{Sample frame title}
This is a text in first frame. This is a text in first frame. This is a text in first frame.
\end{frame}

\end{document}
```

Définition de la classe de document

La toute première ligne de chacun de vos programmes LaTeX devrait le faire. Il devrait suivre la forme `\documentclass{...}`. Ce que vous mettez dans les accolades est très important. Certaines classes de documents vous donnent des commandes supplémentaires à utiliser, d'autres utilisent un format différent et toutes peuvent être entrées avec des paramètres spécifiques (décrits dans la section des paramètres).

Lire Classes de document en ligne: <https://riptutorial.com/fr/latex/topic/7458/classes-de-document>

Chapitre 5: Composition mathématique

Introduction

L'un des principaux avantages de LaTeX est sa capacité à composer des équations. Ici, les principes fondamentaux des équations de composition, certains des différents paquets qui peuvent être utilisés, ainsi que des symboles communs, sont décrits.

Syntaxe

- `\begin {equation} ... \end {equation}`
- `text $... $ text`
- `\usepackage {amsmath} ... \begin {equation *} ... \end {equation *}`

Remarques

Voici quelques idées de base pour vous assurer que votre code ne vous brise pas et que vos équations sont meilleures:

1. Assurez-vous que tous les crochets, accolades, signes dollar et commandes `\begin{}` `\end{}` correspondent. C'est une chose où une petite erreur peut perturber complètement votre code.
2. Si vous obtenez des erreurs, assurez-vous que le package approprié est chargé (par exemple, n'utilisez pas la commande `\begin{equation*}` sans le package `amsmath`).
3. Jamais, jamais, *jamais* utiliser des signes double dollar (`$$an equation here$$`) au lieu de `\begin{equation}` .
4. N'utilisez jamais le mode mathématique pour rendre votre texte en italique.
5. Complètement coincé? Essayez [TeX.SX](https://www.tex.sx) , un site pour répondre aux questions sur TeX, LaTeX et les langages associés.

Bonne chance!

Exemples

Équations de base

Equations simples en ligne

Vous pouvez faire une simple équation en ligne en utilisant `$an equation here$` .

Par exemple, vous pourriez faire

```
 $\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar{z}$
```

qui, si nous mettons un petit texte faux autour de lui, donne

Foo $\lim_{n \rightarrow \infty} \frac{1}{2^n} i \bar{z}$ quux

Équations numérotées et centrées

Lorsque vous rédigez des articles ou d'autres documents, il est parfois préférable que vos équations soient centrées et numérotées, et non en ligne. Ensuite, utilisez les commandes

`\begin{equation}` **et** `\end{equation}` .

Par exemple, si nous utilisons le code

```
\begin{equation}
\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar{z}
\end{equation}
```

Et ajoutez un petit texte autour, nous obtenons

Foo quux bla lipsum lipsum $\lim_{n \rightarrow \infty} \frac{1}{2^n} i \bar{z}$ foo quux bla.

Vous pouvez supprimer la numérotation de l'équation en utilisant `\begin{equation*}` **et** `\end{equation*}` .

Par exemple, si nous utilisons le code

```
\begin{equation*}
\lim\limits_{n \to \infty} \frac{1}{2^n} i\bar{z}
\end{equation*}
```

et ajouter un petit texte autour, nous obtenons

Foo quux bla bla lipsum $\lim_{n \rightarrow \infty} \frac{1}{2^n} i \bar{z}$ quux.

(mais il faut noter que vous devez utiliser le paquet `amsmath` pour cela).

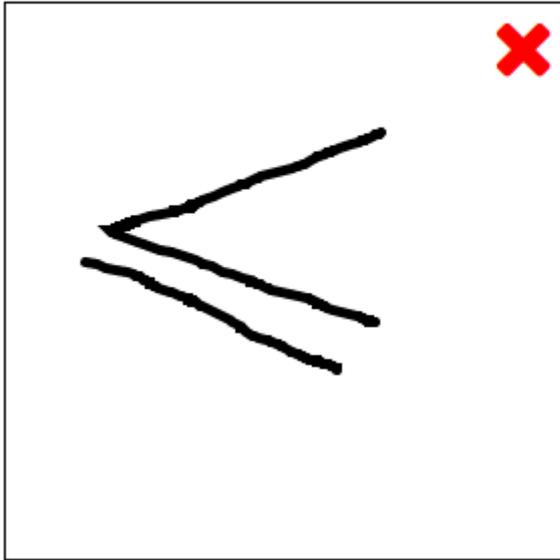
Trouver des symboles

Parfois, il peut être difficile de trouver le symbole mathématique dont vous avez besoin. Il y a plusieurs options ici. Le premier (et le plus rapide) est d'utiliser [Detexify](#) , où vous dessinez le symbole que vous souhaitez, et il essaie de trouver ce que vous voulez, comme indiqué ci-dessous:

Detexify

classify

symbols



Score: 0.076721263121454
`\usepackage{ amssymb }`
`\leqslant`
mathmode



Score: 0.11561454413754882
`\usepackage{ amssymb }`
`\leq`
mathmode



Score: 0.11892004279045627
`\usepackage{ tipa }`
`\texttrptr`
textmode

Une autre option consiste à utiliser la liste complète des symboles LaTeX, disponible [ici](#) . Si vous utilisez le package `unicode-math` [cette liste](#) de tous les symboles pris en charge peut être utile. Une autre option est [ce site Web](#) , qui a des symboles mathématiques communs.

Packages disponibles pour utilisation

Alors que LaTeX standard est tout ce qui est nécessaire pour la plupart des formules et des équations mathématiques simples, il faut parfois plus de symboles et d'outils. Il existe plusieurs forfaits qui amélioreront vos équations et vous fourniront plus de travail. Trois des principaux packages sont décrits ci-dessous. N'oubliez pas, pour charger un package, tapez

`\usepackage{package}` dans votre préambule de document.

`amsmath`

Le package `amsmath` est un package incroyablement utile. Il est utilisé pour permettre à vos équations d'être centrées mais non numérotées, comme dans `\begin{equation*}` , il est utilisé pour créer des matrices (comme décrit ci-dessous) et introduit de nombreuses autres commandes utiles, telles que `\overset` et `\underset` , décrit ci-dessous. La documentation du paquetage `amsmath` est disponible [ici](#) .

`mathtools`

Le package `mathtools` s'appuie sur le package `amsmath` , en ajoutant d'autres symboles et outils utiles. Il charge automatiquement le package `amsmath` , vous n'avez donc pas besoin de charger les deux dans le préambule de votre document. La documentation `mathtools` se trouve [ici](#) .

`amssymb`

Le package `amssymb` fournit de nombreux symboles supplémentaires qui peuvent être très utiles pour les équations plus complexes. La documentation `amssymb` peut être trouvée [ici](#) .

Paquets de polices

Il existe également diverses polices que vous pouvez utiliser pour vos équations, comme décrit sur [cette question](#) sur l'échange de piles TeX, pour TeX, LaTeX et les amis.

[Ce document](#) est une explication concise des différentes fonctionnalités fournies par certains packages ainsi que par la norme LaTeX; c'est très utile.

Bonnes Commandes à Savoir

Les commandes les plus courantes sont les suivantes:

- **Fractions et racines carrées:** Pour les fractions, utilisez `\frac {numerator}{denominator}` .
Pour les racines carrées, utilisez `\sqrt [root]{number}` .
- **Lettres grecques:** utilisez les commandes données dans le tableau ci-dessous:

αA	<code>\alpha A</code>	νN	<code>\nu N</code>
βB	<code>\beta B</code>	$\xi \Xi$	<code>\xi \Xi</code>
$\gamma \Gamma$	<code>\gamma \Gamma</code>	$o O$	<code>o O</code>
$\delta \Delta$	<code>\delta \Delta</code>	$\pi \Pi$	<code>\pi \Pi</code>
$\epsilon \varepsilon E$	<code>\epsilon \varepsilon E</code>	$\rho \varrho P$	<code>\rho \varrho P</code>
ζZ	<code>\zeta Z</code>	$\sigma \Sigma$	<code>\sigma \Sigma</code>
ηH	<code>\eta H</code>	τT	<code>\tau T</code>
$\theta \vartheta \Theta$	<code>\theta \vartheta \Theta</code>	$\upsilon \Upsilon$	<code>\upsilon \Upsilon</code>
ιI	<code>\iota I</code>	$\phi \varphi \Phi$	<code>\phi \varphi \Phi</code>
κK	<code>\kappa K</code>	χX	<code>\chi X</code>
$\lambda \Lambda$	<code>\lambda \Lambda</code>	$\psi \Psi$	<code>\psi \Psi</code>
μM	<code>\mu M</code>	$\omega \Omega$	<code>\omega \Omega</code>

- **Opérateurs:** `\leq` donne le symbole inférieur ou égal à, `\geq` donne le symbole supérieur ou égal à, `\neq` donne le symbole non égal, `\sum` donne le symbole de sommation, `\partial` donne le symbole dérivé partiel, `\nabla` donne l'opérateur Laplacien, `\times` donne le produit croisé ou le symbole de multiplication, `\cdot` donne le produit scalaire ou le symbole de multiplication et `\int` donne le symbole intégral.
- **Les flèches:** `\rightarrow` et `\leftarrow` donnent respectivement des flèches droite et gauche.
- **Pourcentages:** si vous tapez% dans LaTeX, il est important d'inclure une barre oblique

inverse, \% car le symbole de pourcentage est normalement utilisé pour les commentaires.

- **Index supérieurs et indices:** Pour **créer** un exposant, vous pouvez taper x^2 ou, pour les exposants plus longs, x^{2x} . Pour faire un indice, vous pouvez taper x_a ou, pour des indices plus longs, x_{ab} .
- **Gras:** Utilisez `\boldmath{...}` pour rendre vos symboles mathématiques en gras. D'autres options sont données à [cette question TeX.SX](#). Les symboles mathématiques sont en italique Si vous ne voulez pas que cela soit vrai, rédigez votre équation comme décrit ci-dessous.
- **Infinity:** Pour écrire l'infini, utilisez la commande `\infty`.
- **Déplacement d'éléments sur ou sous un autre:** Tout d'abord, pour les opérateurs mathématiques uniquement, il existe une autre méthode. Vous pouvez taper l'opérateur mathématique, disons `\int`, puis utiliser la commande `\limits`. Un exemple est `\int\limits_{\infty}` ou `\int\limits^{\infty}`. Ensuite, pour les cas normaux, vous pouvez faire `\overset{top}{normal}` ou `\underset{bottom}{normal}`. Cela peut être très utile pour faire des vecteurs. Par exemple, vous pourriez faire `\vec{x}`. Le `amsmath` package est nécessaire pour `overset` et `underset`.
- **Accolades:** Comme les accolades sont utilisées dans les commandes, il est nécessaire de taper `\{` ou `\}` pour obtenir des accolades.
- **Texte:** Pour inclure du texte dans les équations, tapez `\usepackage{amsmath}` dans le préambule, puis tapez `\text{...}`.
- **Space:** Pour ajouter de l'espace dans vos équations, tapez `\quad` entre les deux éléments que vous souhaitez séparer (par exemple, vous pourriez avoir $2x \quad \cos$).

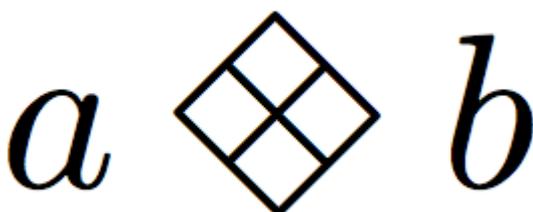
Créer de nouveaux symboles

Disons que vous ne pouvez trouver le symbole dont vous avez besoin nulle part. Vous pouvez créer un symbole personnalisé. Par exemple, le code

```
\documentclass{article}
\usepackage{graphicx,amsmath,amssymb}
\DeclareRobustCommand{\diamondtimes}{%
  \mathbin{\text{\rotatebox[origin=c]{45}{\boxplus}}}%
}

\begin{document}
$a\diamondtimes b$
\end{document}
```

crée et appelle un symbole, donnant



Ceci est un exemple plus simple; il suffit simplement de faire pivoter un symbole existant. Cependant, vous pouvez créer des symboles plus complexes.

Cette section est en cours d'élargissement.

Matrices

Matrices

Vous devez toujours utiliser le package `amsmath` si vous souhaitez utiliser les commandes suivantes. Il existe quatre types principaux de matrice, comme indiqué dans le code ci-dessous:

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

Ce code produit

$$\begin{matrix} a & b \\ c & d \end{matrix} \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

Il y a quelques points importants à noter à ce sujet:

1. Il est important de placer votre matrice dans l' `equation` , l' `equation*` ou l'environnement `$...$` - la commande `bmatrix` n'est pas un environnement mathématique en soi.
2. La construction de la matrice est en fait assez simple. Pour chaque ligne, vous créez chaque élément (disons `x_{11}`), puis placez un `&` , puis écrivez l'élément suivant. Pour plusieurs lignes, à la fin de chaque ligne, placez `\\` (il n'est pas nécessaire de le faire pour la dernière ligne). Il est assez semblable à un tableau dans ce document.

Lire **Composition mathématique en ligne**: <https://riptutorial.com/fr/latex/topic/5950/composition-mathematique>

Chapitre 6: Compteurs, si instructions et boucles avec latex

Exemples

Compteur déclaration, initialisation et impression en pdf

Il est possible d'utiliser des variables entières avec latex. Pour créer une nouvelle variable, nous avons besoin de la commande `\newcounter{name}` , où `name` est le nom du nouveau compteur. Le `name` doit contenir que des lettres. Cette commande en crée une nouvelle avec le nom `\thename` . Avec cette commande, nous pouvons imprimer une variable de `name` sur le papier. La valeur initiale de `name` est 0. Pour donner une valeur à "name", nous pouvons utiliser `\setcounter{name}{n}` où `n` est un entier. `\value{name}` est une fonction qui renvoie la valeur de `name` .

```
\documentclass{article}
\begin{document}
\newcounter{num}           %new counter, initial value is 0
\thenum                   %print 0
\setcounter{num}{3}       %set num to 3
\thenum                   %print 3
\newcounter{number}
\setcounter{number}{\value{num}} %set number to value of num
\thenumber                %print 3
```

Latex provides some other formats to print a number.

Other types of printing:

```
\arabic{num} \\
\Roman{num} \\ %→ I, II, III, IV, . . . (num = 1, 2, 3, . . . )
\roman{num} \\ %→ i, ii, iii, iv, . . . (num = 1, 2, 3, . . . )
\Alph{num} \\ %→ A, B, C, D, . . . (num = 1, 2, 3, . . . , 26)
\alph{num} \\ %→ a, b, c, d, . . . (num = 1, 2, 3, . . . , 26)
\fnsymbol{num} \\ %→ *, †, ‡, §, ¶, k, **, ††, ‡‡ (num = 1, 2, 3, . . . , 9)
\end{document}
```

0

3

3

Latex provides some other formats to print a number.

Other types of printing:

3

III

iii

C

c

‡

Opérations avec des compteurs

Cet exemple montre comment utiliser des opérations mathématiques avec des compteurs. Cela peut être utile pour les boucles en latex.

Ajout: `\addtocounter{num}{n}`

cette commande ajoute n à num , où num est un compteur et n est un entier positif.

Soustraction: `\addtocounter{num}{-n}`

cette commande soustrait n de num , où num est un compteur et n est un entier positif.

Multiplication: `\multiply\value{num} by n`

cette commande multiplie num par n , où num est un compteur et n est un entier.

Division `\divide\value{num} by n`

cette commande divise num par n et obtient la partie entière du quotient (num est un compteur et n est un entier)

```
\documentclass{article}
\begin{document}
\newcounter{num}
\setcounter{num}{3}
\addtocounter{num}{10}
\thenum\%prints 13
\addtocounter{num}{-3}
\thenum\%prints 10
\stepcounter{num}
\thenum\%prints 11
\multiply\value{num} by \value{num}
\thenum\%prints 121
\multiply\value{num} by 2
\thenum\%prints 242
\divide\value{num} by 60
\thenum%prints 4
\end{document}
```

`\newcommand{num}` déclare un compteur. `\setcounter{num}{3}` définit la valeur de num sur 3.

`\addtocounter{num}{10}` ajoute 10 à num .

`\addtocounter{num}{-3}` soustrayez 3 de num .

`\stepcounter{num}` ajoute 1 à num

`\multiply\value{num} by \value{num}` carrés num .

`\multiply\value{num} by 2` doubles num .

`\divide\value{num} by 60` divise num par 60 et obtient la partie entière.

Le résultat du code: 13 \ 10 \ 11 \ 121 \ 242 \ 4

(\\ symbolise la nouvelle ligne)

Le package intcalc ajoute d'autres opérations sur les nombres entiers, par exemple mod, pow, sng, abs, inv ...

[intcalc_package.pdf](#)

Si déclarations

Dans latex, nous pouvons utiliser des commandes intégrées pour exécuter du code, que les conditions soient vraies ou non.

Comparaison de deux entiers: `\ifnum\value{num}>n {A} \else {B}\fi`

Ce code exécute A si $\text{num} > n$ else B. On peut substituer $>$ par $<$ et $=$.

Si un nombre est impair: `\ifodd\value{num} {A}\else {B}\fi`

Si num est impair, alors il exécute A else B.

Si avec condition: `\ifthenelse{condition}{A}{B}`

Nous devons charger si le paquet pour utiliser cette commande. Si la condition est vraie, elle exécute A else B.

Il est possible de créer une condition complexe avec `\(\)`, `\AND`, `\OR`, `\NOT`.

Par exemple: `\ifthenelse{\(\NOT 4<2 \OR 4>11\)\AND\isodd{4}}{A}{B}`

Ce morceau de code écrit "B" sur la page. `\NOT 4<2` est vrai et `4>11` est faux. Si nous connectons une instruction false et une instruction true avec "OR", le résultat est vrai. Donc `\(\NOT 4<2 \OR 4>11\)` est vrai. `\isodd{4}` est faux car 4 est pair. Une instruction false et une instruction true connectées à "AND" sont fausses, la sortie est donc B.

Un exemple de code:

```
\documentclass{article}
\usepackage{ifthen}
\begin{document}
  \newcounter{num}
  \setcounter{num}{10}

  If num$>$100 then the next sentence will be "Num is large." else "Num is small."

  Num is \ifnum \value{num}>100 {large} \else {small}.

  If num is odd then the next sentence will begin with "Odd" if not then with "Even"

  \ifodd \value{num} {Odd} \else {Even} numbers are cool.

  If (num$>$3 and (1$<$0 or num$=$10)) is true then the next sentence will be "True." else
  "False."
```

```

\ifthenelse{\value{num}>3\AND\ (1<0 \OR \value{num}=10\)}{True.}{False.}

\end{document}

```

If num>100 then the next sentence will be "Num is large." else "Num is small."
 Num is small.
 If num is odd then the next sentence will begin with "Odd", if not then with "Even"
 Even numbers are cool.
 If (num>3 and (1<0 or num=10)) is true then the next sentence will be "True." else "False."
 True.

Boucles - répéter des choses

Nous pouvons créer des boucles en latex. Ils sont similaires mais pas aussi personnalisables que les boucles dans d'autres langages de programmation. Une alternative à l'utilisation de boucles est @loops. Si nous utilisons une commande qui inclut "@" dans son nom, il faut le placer entre \makeatletter et \makeatother . Il n'est pas autorisé de les utiliser dans une macro qui décrit une nouvelle définition.

Faux:

```

\def\is#1#2{\makeatletter\@ifstar{#1}{#2}\makeatother

```

Droite:

```

\makeatletter\def\is#1#2{\@ifstar{#1}{#2}}\makeatother

```

@pour la boucle: \@for\command:={list}\do{commands}

Exemple :

```

\makeatletter
\@for\sun:={rising,setting}\do{The sun is \sun.}
\makeatother

```

Il crée le texte suivant: Le soleil se lève. Le soleil se couche.

@whilenum loop: \@whilenum condition\do{commands}

Exemple :

```

\makeatletter
\newcounter{int}
\@whilenum\value{int}<10\do
{\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{\theint}{}}
\makeatother

```

Ce code écrit les nombres impairs de 1 à 9.

"loop repeat" loop: `\loop {commands} \ifnum condition \repeat`

Exécute les commandes jusqu'à ce que la condition soit vraie.

Exemple

```
\setcounter{int}{1}
\loop
\theint
\addtocounter{int}{2}
\ifnum \value{int}<10
\repeat
```

Ce code fait la même chose que la boucle `@whilenum`.

Un exemple de code:

```
\documentclass{article}
\usepackage{ifthen}
\usepackage{amsmath} %\text{} command needs this package
\begin{document}
  Demonstration of @for loop:

  \makeatletter
  \@for\sun:={rising,setting}\do{The sun is \sun. }
  \makeatother

  \newcounter{int}

  @whilenum loop:

  \setcounter{int}{0}
  \makeatletter
  \@whilenum\value{int}<20\do
  {\stepcounter{int}\ifthenelse{\isodd{\value{int}}}{\theint\text{ }}{}}
  \makeatother

  "loop repeat" loop:

  \setcounter{int}{1}
  \loop
  \theint
  \text{ }\addtocounter{int}{2}\ifnum\value{int}<20
  \repeat
\end{document}
```

```
Demonstration of @for loop:
The sun is rising. The sun is setting.
@whilenum loop:
1 3 5 7 9 11 13 15 17 19
"loop repeat" loop:
1 3 5 7 9 11 13 15 17 19
```

Utiliser des boucles dans Tikz

Les boucles sont utiles dans Tikz.

Le code suivant dessine une horloge sans chiffres:

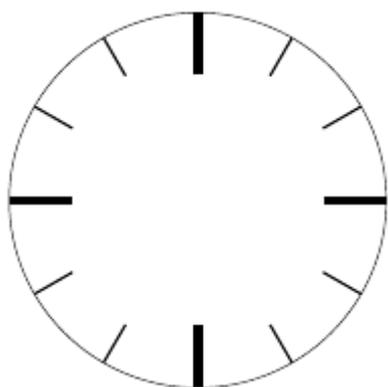
```

\documentclass{article}
\usepackage{ifthen}
\usepackage{intcalc}
\usepackage{tikz}
\newcounter{num}

\begin{document}
\begin{tikzpicture}
  \makeatletter
  \setcounter{num}{1}
  \newcounter{angle}
  \draw (0,0) circle (3cm);
  \@whilenum\value{num}<13\do{
    \setcounter{angle}{360}
    \multiply\value{angle} by \value{num}
    \divide\value{angle} by 12
    \ifnum \intcalcMod{\value{num}}{3}=0{
      \draw[line width=4pt] (\theangle:2cm) -- (\theangle:3cm);    }else
    {
      \draw[line width=1pt] (\theangle:2.3cm) -- (\theangle:3cm);
    }\fi
    \addtocounter{num}{1}
  }
  \makeatother
\end{tikzpicture}
\end{document}

```

Le résultat:



Lire Compteurs, si instructions et boucles avec latex en ligne:

<https://riptutorial.com/fr/latex/topic/9224/compteurs--si-instructions-et-boucles-avec-latex>

Chapitre 7: Création d'affiches à l'aide d'un beamer

Introduction

La création d'une affiche à l'aide du package `beamerposter` est très similaire à la création d'une image unique. Placez le contenu dans des colonnes. Dans chaque colonne, séparez le contenu en utilisant des blocs.

Exemples

Orientation et taille

Lors de l'ajout du package `beamerposter`, fournissez les paramètres requis.

```
\usepackage[orientation=landscape, size=a1]{beamerposter}
```

Vous pouvez également personnaliser la taille de l'affiche.

```
\usepackage[orientation=portrait, size=custom, height=110, width=80, scale=1.4]{beamerposter}
```

Les dimensions en hauteur et largeur sont ici en cm. L' `scale` est utilisée pour la taille de la police.

Plan de base d'une affiche de beamer

En orientation paysage

```
\documentclass[final,t]{beamer}
\mode<presentation>
{
  \usetheme{Berlin}
}

\usepackage[orientation=landscape, size=a1, scale=1, debug]{beamerposter}
\usepackage{lipsum} % for dummy text

\title[]{\huge Awesome title}
\author[]{\large \textbf{Author Name1} \and Author Name2 \and Author Name3}
\institute[]{\Large Dept of XYZ, ABC Institute}
\date{}

\begin{document}

\begin{frame}
\maketitle
\begin{columns}[t]
  \begin{column}{.32\linewidth}

  \begin{block}{Some heading}
```

```
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\end{column}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\end{column}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\begin{column}{.32\linewidth}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\begin{block}{Some heading}
\lipsum[1]
\end{block}

\end{column}
\end{columns}

\end{frame}

\end{document}
```

Some heading

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.

Some heading

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.

Some heading

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.

<https://riptutorial.com/fr/latex/topic/10714/creation-d-affiches-a-l-aide-d-un-beamer>

Chapitre 8: Créer une bibliographie

Syntaxe

- Pour une bibliographie formatée manuellement, il n'est pas nécessaire d'avoir des citations - `\cite` - dans le document.

Paramètres

Paramètre	Détail
<code>thebibliography</code>	Cet environnement définit la portée de la bibliographie proprement dite. Il définit un environnement de type liste dans lequel vous pouvez utiliser <code>\bibitem</code> pour définir un élément de bibliographie.
<code>{x}</code>	L'environnement <code>thebibliography</code> prend un seul argument qui représente l'élément le plus large attendu dans l'énumération des <code>\bibitem</code> s. Pour moins de 10 entrées, utilisez un seul caractère / chiffre; pour moins de 100 entrées, utilisez deux caractères / chiffres, ...
<code>\bibitem{<a>}</code> <code></code>	Définissez l'élément de bibliographie <code></code> et rendez-le disponible pour <code>\cite</code> dans le document en utilisant l'étiquette <code><a></code> .

Exemples

Bibliographie de base sans paquets (formatage manuel)

See [1] or [2] or [1, 2].

References

[1] AUTHOR, A, *A title*, Journal of So-and-So, 2000.

[2] SOMEONE, B, *Another title*, Book of books, 1900.

```
\documentclass{article}% or book, report, ...

\begin{document}

See \cite{citeA} or \cite{citeB} or \cite{citeA, citeB}.

\begin{thebibliography}{x}
% \bibitem{<biblabel>} <citation>

```

```

\bibitem{citeA}
  {\scshape Author, A}, {\itshape A title}, Journal of So-and-So, 2000.
\bibitem{citeB}
  {\scshape Someone, B}, {\itshape Another title}, Book of books, 1900.
\end{thebibliography}

\end{document}

```

Notez que si vous ne savez pas vraiment *pourquoi*, vous ne devriez probablement pas le faire. L'utilisation de paquets désignés (voir d'autres exemples) est préférable.

Bibliographie de base avec biber

Pour démarrer une bibliographie, vous devez définir vos sources. Créez un [fichier de base de données](#) (comme `sources.bib`) et incluez du contenu:

```

@book{Doe1993,
  Author = {John Doe},
  Publisher = {Earth University},
  Title = {Creating a bibliography with biber},
  Year = {1993}}

```

Vous pouvez ensuite inclure votre fichier de base de données dans votre document principal et citer la nouvelle source (`Doe1993`).

```

\documentclass{article}

% Include the biblatex package and tell it to use biber as a backend.
% Without specifying the backend, it assumes biber.
\usepackage[backend=biber]{biblatex}

% Define where biber can find your sources
\addbibresource{sources.bib}

\begin{document}
"Biber isn't that difficult." \cite{Doe1993}
% Use \cite{source-ID} to generate a citation

% Print the bibliography
\printbibliography

\end{document}

```

Pour compiler le document, vous devrez exécuter 3 commandes en séquence:

1. `pdflatex` pour créer un fichier auxiliaire qui indique à biber quelles sources sont nécessaires
2. `biber` pour créer un fichier auxiliaire avec toutes les sources utilisables par `pdflatex`
3. `pdflatex` pour inclure le fichier auxiliaire et créer le PDF

"Biber isn't that difficult." [1]

References

- [1] John Doe. *Creating a bibliography with biber*. Earth University, 1993.

Trouvez beaucoup plus d'options et de champs supplémentaires pour les fichiers Bib dans la [documentation](#) du [package](#) sur [CTAN](#) .

Lire [Créer une bibliographie en ligne](#): <https://riptutorial.com/fr/latex/topic/3488/creer-une-bibliographie>

Chapitre 9: Définition de macros

Syntaxe

- `\newcommand {\macro} {texte de remplacement}`
- `\newcommand {\macro} [argcount] {texte de remplacement}`
- `\renewcommand {\macro} {texte de remplacement}`
- `\renewcommand {\macro} [argcount] {texte de remplacement}`

Paramètres

Paramètre	Détails
<code>\macro</code>	La macro à définir
<code>argcount</code>	Le nombre d'arguments attendus par la macro (facultatif)
<code>replacement text</code>	Le texte de remplacement pour la macro. A l'intérieur de ce texte, <code>#1</code> , <code>#2</code> etc. sont remplacés par les arguments de macro.

Exemples

Définition de base des macros

Définir une nouvelle commande de base

Une macro peut être définie en utilisant `\newcommand` . Par exemple:

```
\newcommand{\foo}{Just foo, you see?}
```

définit une macro `\foo` qui s'étend à `Just foo, you see?` . Il peut alors être utilisé comme toute commande intégrée, par exemple après cette définition:

```
He said: ``\foo''
```

se développe à

```
He said: ``Just foo, you see?''
```

Définir une nouvelle commande avec des arguments

Les macros peuvent également avoir des arguments. Le nombre d'arguments est donné comme

argument facultatif entre le nom de la commande et le texte de remplacement. Dans le texte de remplacement, les arguments sont accessibles avec #1 , #2 etc. Par exemple:

```
\newcommand{\better}[2]{A #1 is better than a #2.}
\better{solution}{problem} % gives: A solution is better than a problem
```

Redéfinir une commande existante

Si une macro a déjà été définie, `\newcommand` une erreur. Pour donner une nouvelle définition à une commande existante, `\renewcommand` est utilisé à la place. Outre le nom différent, la syntaxe est exactement la même. Par exemple, après la définition de `\foo` ci-dessus, on pourrait utiliser:

```
\renewcommand{\foo}{Another foo, please.}
```

Après cette redéfinition, la macro `\foo` ne se développe plus en `Just foo, you see?` mais à un `Another foo, please.`

Lire Définition de macros en ligne: <https://riptutorial.com/fr/latex/topic/7658/definition-de-macros>

Chapitre 10: Dessiner des graphiques

Exemples

TikZ - Mise en page manuelle

Package [TikZ](#) se prête très bien au dessin de graphiques.

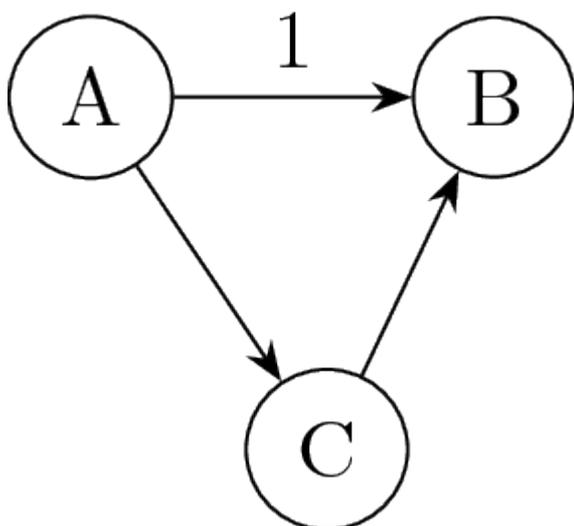
Ceci est un petit exemple (nécessite TikZ 3.0+):

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{positioning,arrows.meta}

\begin{document}
  \begin{tikzpicture}[auto,vertex/.style={draw,circle}]
    \node[vertex] (a) {A};
    \node[vertex,right=1cm of a] (b) {B};
    \node[vertex,below right=1cm and 0.5cm of a] (c) {C};

    \path[-{Stealth[]}]
      (a) edge node {1} (b)
      (a) edge (c)
      (c) edge (b);
  \end{tikzpicture}
\end{document}
```



Vous pouvez créer des graphiques arbitrairement complexes. attention au code long, cependant. Rappelez-vous qu'il existe `\foreach` et prenez note de toutes les options de positionnement et de style (cf. manuel TikZ, sections 13 à 17).

TikZ - Spécifications graphiques

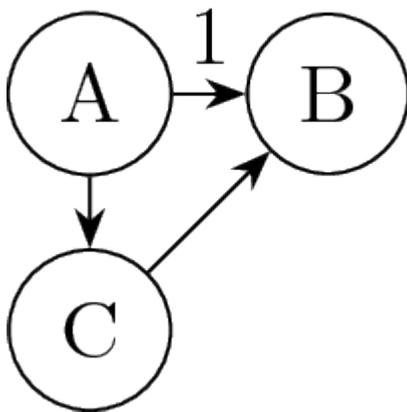
TikZ fournit une syntaxe similaire à [DOT](#) que vous pouvez utiliser pour resserrer considérablement

voire code de dessin graphique.

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,quotes,arrows.meta}

\begin{document}
  \begin{tikzpicture}
    \graph[nodes={draw,circle},edges={->{Stealth[]}}] {
      A -> ["1"] B,
      A -> C,
      C -> B
    };
  \end{tikzpicture}
\end{document}
```

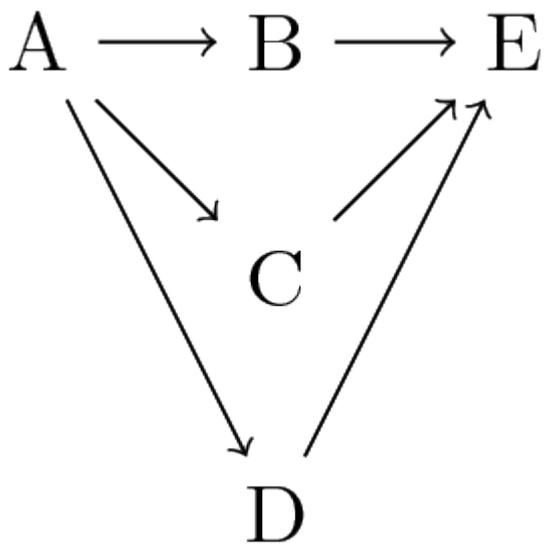


Comme vous pouvez le voir, vous échangez un contrôle plus fin pour une syntaxe simplifiée. La bibliothèque de `graphs` brille vraiment lorsque vous spécifiez des graphiques plus compliqués:

```
\documentclass{standalone}

\usepackage{tikz}
\usetikzlibrary{graphs,graphs.standard}

\begin{document}
  \begin{tikzpicture}
    \graph {
      A -> { subgraph I_n [V= {B,C,D}] } -> E
    };
  \end{tikzpicture}
\end{document}
```



Il y a beaucoup plus d'options et de graphiques prédéfinis. voir la section 19 du manuel TikZ.

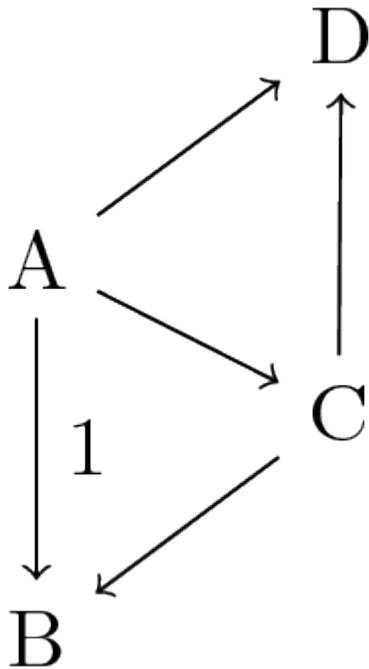
TikZ - Dessin graphique algorithmique

TikZ implémente plusieurs algorithmes pour la *mise en page automatique des graphiques* (nécessite LuaLaTeX).

```
\documentclass{article}

\usepackage{tikz}
\usetikzlibrary{graphs,graphdrawing,quotes}
\usegdlibrary{force}

\begin{document}
  \begin{tikzpicture}
    \graph[spring layout] {
      A -> ["1"] B,
      A -> {C, D},
      C -> {B, D},
    };
  \end{tikzpicture}
\end{document}
```



Il existe plusieurs algorithmes et de nombreuses options pour les influencer. Voir la partie IV du manuel TikZ pour plus de détails.

Diagramme de transition d'état d'une chaîne de Markov

Supposons que la matrice suivante soit la matrice de probabilité de transition associée à une chaîne de Markov.

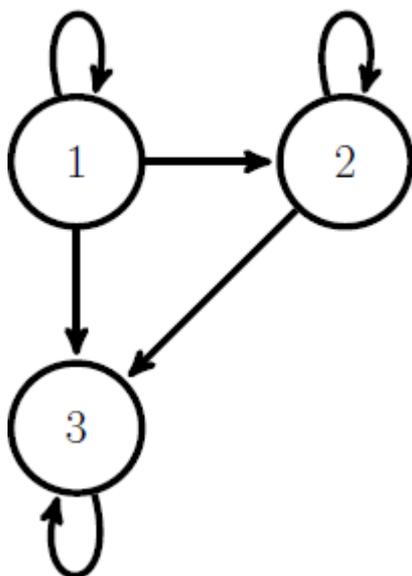
$$P = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.0 & 0.1 & 0.9 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

Afin d'étudier la nature des états d'une chaîne de Markov, un diagramme de transition d'état de la chaîne de Markov est tracé.

```

\documentclass[12pt,a4paper]{article}
\usepackage{tikz}
\usetikzlibrary{shapes,arrows,positioning}
\begin{tikzpicture}[->,>=stealth',shorten >=2pt, line width=3pt,
                    node distance=2cm, style ={minimum size=20mm}]
\tikzstyle{every node}=[font=\huge]
\node [circle, draw] (a) {1};
\path (a) edge [loop above] (a);
\node [circle, draw] (b) [right=of a] {2};
\path (b) edge [loop above] (b);
\draw[->] (a) -- (b);
\node [circle, draw] (c) [below=of a] {3};
\path (c) edge [loop below] (c);
\draw[->] (a) -- (c);
\draw[->] (b) -- (c);
  
```

```
\end{tikzpicture}
```



Lire Dessiner des graphiques en ligne: <https://riptutorial.com/fr/latex/topic/5955/dessiner-des-graphiques>

Chapitre 11: En-tête et pied de page

Exemples

Utiliser des paquets fancyhdr et titleps

```
\documentclass[12pt]{article}

\usepackage{titleps}
\usepackage{fancyhdr}
\usepackage{graphicx}
\usepackage{lipsum} % for dummy text

\pagestyle{myheadings}
\pagestyle{fancy}
\fancyhf{}

\setlength{\headheight}{30pt}

\renewcommand{\headrulewidth}{4pt}
\renewcommand{\footrulewidth}{2pt}

\fancyhead[L]{\includegraphics[width=1cm]{example-image-a}}
\fancyhead[C]{}
\fancyhead[R]{\rightmark}
\fancyfoot[L]{ABC}
\fancyfoot[C]{\textcopyright xyz}
\fancyfoot[R]{\thepage}

\begin{document}

\section{First section}
\subsection{One}
\lipsum[1-3]
\subsection{Two}
\lipsum[4-6]

\end{document}
```

Chapitre 12: Formatage du texte

Exemples

Emphase le texte

Pour accentuer le texte, `\emph` pouvez utiliser la commande `\emph` qui affiche généralement le texte en italique:

```
This is some text with \emph{emphasized words}.
```

Frappe à travers le texte

La commande `\sout` du paquet `ulem` traverse un texte:

```
\sout{This text is striked through}
```

Le paquet `ulem` redéfinit la commande `\emph`. Lorsque vous ne voulez pas avoir ce comportement, vous pouvez utiliser le package `ulem` avec l'option `normalem`:

```
\usepackage[normalem]{ulem}
```

Texte en gras

Pour composer du texte en gras, utilisez `\textbf`:

```
\textbf{This text is typeset in bold.}
```

Lire Formatage du texte en ligne: <https://riptutorial.com/fr/latex/topic/7245/formatage-du-texte>

Chapitre 13: Gravure de partitions

Exemples

LilyPond

Le graveur de notation LilyPond peut être utilisé avec LaTeX via la commande `lilypond-book`. Commençons par créer un document LaTeX (avec l'extension de fichier `.lytex`) pour intégrer notre musique dans:

```
\documentclass[letterpaper,12pt]{article}

\begin{document}

\begin{center}
  {\fontsize{24pt}{24pt}\textbf{Twa Corbies}}\
\end{center}

\begin{flushright}
  \textsc{Your Name}
\end{flushright}

% We don't need to require anything for this because lilypond-book will process it.
\lilypondfile{TwaCorbies.ly}
\end{document}
```

Ensuite, nous créons notre fichier LilyPond (`.ly`), y compris le fichier `lilypond-book-preamble.ly` (que LilyPond saura trouver):

```
\version "2.16.2"

\include "lilypond-book-preamble.ly"

voice = <<
  \relative c' {
    \tempo "con affetto"
    \clef bass
    \key e \minor
    \time 3/4

    a a b | c a a | g a2 |
    a4 a b | c2 ~ c8 a8 | a8 g a2 |
    \bar "|."
  }
  \addlyrics{
    As I was wal -- king all a -- lane
    I heard twa cor -- bies make a mane.
  }
>>

\score {
  <<
    \new Staff = "voice" {
      \voice
```

```
    }
  >>
  \layout { }
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 90 4)
    }
  }
}
```

pour construire, nous `lilypond-book` ensuite la commande `lilypond-book` :

```
lilypond-book --include=mymusicsourcedirectory/ --pdf TwaCorbies.lytex
```

qui affichera un fichier PDF contenant votre musique gravée LilyPond:

con affetto



As I

4



I heard

Chapitre 14: les tables

Exemples

L'environnement tabulaire

L'environnement `tabular` est le moyen le plus simple de créer une table dans LaTeX et ne nécessite aucun autre package.

```
\begin{tabular}{|lcr|}  
  left aligned column & center column & right column \\  
  \hline  
  text & text & text \\  
  text & text & text \\  
\end{tabular}
```

left aligned column	center column	right column
text	text	text
text	text	text

Le paramètre (`|lcr|` dans l'exemple) s'appelle la **spécification de la table** et indique à LaTeX le nombre de colonnes et leur format. Chaque lettre représente une seule colonne. Les valeurs possibles sont:

Personnage	Sens
l	colonne alignée à gauche
c	colonne centrée
r	colonne alignée à droite
p {width} p{5cm} . ex. p{5cm}	colonne de paragraphe avec largeur définie
(personnage de pipe)	ligne verticale
(2 tuyaux)	2 lignes verticales

Les cellules sont séparées par le caractère `&`. Une ligne est terminée par 2 barres obliques inversées `\\`.

Les lignes horizontales peuvent être insérées à l'aide de la commande `\hline`.

Les tableaux sont toujours formatés pour être suffisamment larges pour inclure tout le contenu. Si une table est trop grande, LaTeX imprimera des `overflow hbox` excessifs. Les solutions possibles incluent l'utilisation du spécificateur `p{'width'}` ou d'autres packages tels que `tabularx`.

Un tableau avec des en-têtes de colonne couvrant plusieurs colonnes peut être créé à l'aide de la commande `\multicolumn{cols}{pos}{text}`.

```
\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
&\multicolumn{3}{|c|}{Income Groups}\\
\cline{2-4}
City&Lower&Middle&Higher\\
\hline
City-1& 11 & 21 & 13\\
City-2& 21 & 31 & 41\\
\hline
\end{tabular}
\end{center}
```

City	Income Groups		
	Lower	Middle	Higher
City-1	11	21	13
City-2	21	31	41

Notez que la commande `\multicolumn` a trois arguments obligatoires: le premier argument spécifie le nombre de colonnes sur lesquelles l'en-tête s'étend; le second argument spécifie la position de l'en-tête (*l, c, r*); et le troisième argument est le texte de l'en-tête. La commande `\cline{2-4}` spécifie la colonne de début (ici 2) et la colonne de fin (ici 4) sur lesquelles une ligne doit être dessinée.

Tableau de coloriage

Pour rendre le tableau plus lisible, voici les moyens de le colorer:

1. Des rangées
2. Colonnes
3. Lignes
4. Cellules

Rangées de coloriage

Utilisez `\rowcolor` (fourni par `colortbl`; également chargé par `xcolor` sous l'option de package `[table]`). Exemple:

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{|l|l|l|l|}
\rowcolor{green}
A & B & C & \\
\rowcolor{red}
D & E & F & \\
\end{tabular}
```

```
G & H & I \\
\rowcolor{blue}
J & K & L
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I
J	K	L

Colonnes De Coloration

Les colonnes peuvent être colorées comme suit:

- Définition d'une propriété de couleur de colonne en dehors de la balise de table à l'aide de

`\newcolumntype` :

```
\newcolumntype{a}{ >{\columncolor{yellow}} c }
```

- Définition d'une propriété de couleur de colonne dans les paramètres de la table

```
\begin{tabular}{ | >{\columncolor{red}} c | l | l }
```

Exemple:

```
\documentclass{article}
\usepackage[table]{xcolor}

\newcolumntype{a}{>{\columncolor{yellow}}c}
\newcolumntype{b}{>{\columncolor{green}}c}

\begin{document}

\begin{tabular}{ a | >{\columncolor{red}}c | l | b }
\hline
A & B & C & D \\
E & F & G & H \\
\hline
\end{tabular}

\end{document}
```

A	B	C	D
E	F	G	H

Lignes de coloriage

Utilisez `\arrayrulecolor` . Exemple:

```
\documentclass{article}
\usepackage[table]{xcolor}

\arrayrulecolor{blue}

\begin{document}

\begin{tabular}{| l | l | l | }
\hline
A & B & C \\
\hline
D & E & F \\
\hline
G & H & I \\
\hline
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I

Cellules colorantes

Utilisez `\cellcolor` . Exemple:

```
\documentclass{article}
\usepackage[table]{xcolor}

\begin{document}

\begin{tabular}{| l | l | l | }
\hline
A & B & C \\
\hline
D & E & \cellcolor{green}F \\
\hline
\end{tabular}
```

```
G & H & I \\
\hline
\end{tabular}

\end{document}
```

A	B	C
D	E	F
G	H	I

Nous pouvons aussi définir nos propres couleurs en utilisant le package `colortbl`. Voici les exemples de balises:

```
\definecolor{Gray}{gray}{0.85}
\columncolor[RGB]{230, 242, 255}
\columncolor[HTML]{AAACED}
```

Lire les tables en ligne: <https://riptutorial.com/fr/latex/topic/4956/les-tables>

Chapitre 15: Outils de construction

Exemples

Arara

Arara est un outil d'automatisation multi-plateforme spécialement conçu pour TeX. Il est inclus dans une distribution standard, il n'est donc pas nécessaire d'installer quoi que ce soit d'autre. C'est le moyen le plus efficace pour enregistrer les instructions de compilation dans le fichier TeX lui-même:

```
% arara: pdflatex
\documentclass{article}
\begin{document}
  Hello, world
\end{document}
```

Celles-ci peuvent être beaucoup plus compliquées, cependant:

```
% arara: pdflatex
% arara: biber
% arara: pdflatex

% To support a self-contained example, this builds a BibTeX file on-the-fly
\begin{filecontents}{references.bib}
@article{dijkstra,
  author = {Dijkstra, Edsger W.},
  title = {Self-stabilizing Systems in Spite of Distributed Control},
  journal = {Commun. ACM},
  issue_date = {Nov. 1974},
  volume = {17},
  number = {11},
  month = nov,
  year = {1974},
  issn = {0001-0782},
  pages = {643--644},
  numpages = {2},
  url = {http://doi.acm.org/10.1145/361179.361202},
  doi = {10.1145/361179.361202},
  acmid = {361202},
  publisher = {ACM},
  address = {New York, NY, USA},
  keywords = {distributed control, error recovery, harmonious cooperation, multiprocessing,
mutual exclusion, networks, robustness, self-repair, self-stabilization, sharing,
synchronization},
}
\end{filecontents}

\documentclass{article}
\usepackage[backend=biber]{biblatex}
\addbibresource{references.bib}

\begin{document}
```

```
Hello, World! \cite{dijkstra}.  
\printbibliography  
\end{document}
```

Lire Outils de construction en ligne: <https://riptutorial.com/fr/latex/topic/5015/outils-de-construction>

Chapitre 16: Pages de titres

Remarques

`\title{<title>}` , `\author{<author>}` et `\date{<date>}` stockent le contenu en interne.

`\maketitle` produit une page de titre standard avec les valeurs précédemment définies.

Exemples

Page de titre du rapport standard

```
\documentclass{report}

\begin{document}

\title{I want to be a Wombat}
\author{Carl Capybara}
\maketitle

\end{document}
```

Cela créera une page de titre sans autre contenu:

I want to be a Wombat

Carl Capybara

July 26, 2016

Lire Pages de titres en ligne: <https://riptutorial.com/fr/latex/topic/3010/pages-de-titres>

Chapitre 17: Présentation avec package beamer

Paramètres

thème	Ann Arbor
thème de couleur	Seahorse

Remarques

Pour d'autres thèmes et couleurs, vous pouvez visiter [ici](#)

Exemples

Diapositive de titre simple d'un auteur

```
\documentclass{beamer}

\mode<presentation>

\usetheme{AnnArbor}

\usecolortheme{seahorse}

\title[Short topic]{Awesome long topic}

\author[Name]{Full name}

\institute[Institute short form]{Full name of institute}

\date{\today}

\begin{document}

\maketitle

\end{document}
```

Awesome loc

Full na

Full name of

July 1,

Crédits

S. No	Chapitres	Contributeurs
1	Commencer avec le latex	Community , eyqs , Harry , hbaderts , jani , Louis , Nijin22 , Sean Allred , SnoringFrog , Spacedman , Sumner Evans , TuringTux , tversteeg
2	Accéder à la documentation des packages LaTeX	Raphael , samcarter
3	Ajouter une citation	Hamzawey
4	Classes de document	Community , heather
5	Composition mathématique	heather
6	Compteurs, si instructions et boucles avec latex	csekri
7	Création d'affiches à l'aide d'un beamer	Sukanya B
8	Créer une bibliographie	Nijin22 , Raphael , Werner
9	Définition de macros	celtschk , strpeter
10	Dessiner des graphiques	L.V.Rao , Raphael
11	En-tête et pied de page	Sukanya B
12	Formatage du texte	celtschk , Stephan Kulla
13	Gravure de partitions	Sam Whited
14	les tables	L.V.Rao , Nijin22 , Nikita Jain , Werner
15	Outils de construction	Sean Allred

16	Pages de titres	adn , Johannes_B , Nijin22 , Sam Whited
17	Présentation avec package beamer	Sukanya B