



EBook Gratis

APRENDIZAJE

latex

Free unaffiliated eBook created from
Stack Overflow contributors.

#latex

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con el l^atex.....	2
Observaciones.....	2
L^atex.....	2
¿Qué es LaTeX?.....	2
Versiones.....	2
Examples.....	2
Instalación y configuración.....	2
Instalación.....	3
Windows (TeXLive).....	3
Windows (MiKTeX).....	3
Mac OS X (TeXLive).....	3
Linux (TeXLive).....	3
Uso de gestores de paquetes.....	3
Instalación desde aguas arriba.....	4
Instalación de prueba.....	4
Editores de LaTeX.....	5
Capítulo 2: Accediendo a la documentación de los paquetes de LaTeX.....	7
Examples.....	7
CTAN.....	7
TeX Live - texdoc.....	10
Capítulo 3: Añadir cita.....	11
Examples.....	11
Añadir cita al documento LaTeX ya existente.....	11
Capítulo 4: Clases de documentos.....	12
Sintaxis.....	12
Observaciones.....	12
Examples.....	12
Artículo.....	12

¿Cuándo usar la clase de artículo?	12
¿Cuáles son las especificidades de esta clase?	12
Ejemplo simple	12
Beamer	13
¿Cuándo usar la clase beamer?	13
¿Cuáles son las especificidades de esta clase?	13
Ejemplo simple	13
Definiendo la clase de documento	13
Capítulo 5: Contadores, si enunciados y bucles con látex.	15
Examples	15
Contador declaración, inicialización e impresión a pdf.	15
Operaciones con contadores	15
Si declaraciones	17
Bucles - repitiendo cosas	18
Usando bucles en Tikz	19
Capítulo 6: Creación de carteles utilizando beamer.	21
Introducción	21
Examples	21
Orientación y tamaño	21
Esquema básico de un póster de proyector	21
Ejemplo completo de cartel de beamer	23
Capítulo 7: Creando una bibliografía	25
Sintaxis	25
Parámetros	25
Examples	25
Bibliografía básica sin paquetes (formateo manual)	25
Bibliografía básica con biber	26
Capítulo 8: Definiendo macros	28
Sintaxis	28
Parámetros	28
Examples	28

Definición básica de macros.....	28
Definir un nuevo comando básico.....	28
Definir un nuevo comando con argumentos.....	28
Redefiniendo un comando existente.....	29
Capítulo 9: Dibujando graficas.....	30
Examples.....	30
TikZ - Diseño manual.....	30
TikZ - Especificaciones gráficas.....	30
TikZ - Dibujo de gráfico algorítmico.....	32
Diagrama de transición de estado de una cadena de Markov.....	33
Capítulo 10: Encabezado y pié de página.....	35
Examples.....	35
Usando los paquetes fancyhdr y titleps.....	35
Número de página como CurrPage / TotalPages en pie de página.....	36
Capítulo 11: Formato de texto.....	38
Examples.....	38
Enfatizar texto.....	38
Huelga a través del texto.....	38
Texto en negrita.....	38
Capítulo 12: Grabado Partituras.....	39
Examples.....	39
LilyPond.....	39
Capítulo 13: Herramientas de construcción.....	42
Examples.....	42
Arara.....	42
Capítulo 14: Mesas.....	44
Examples.....	44
El ambiente tabular.....	44
Mesa para colorear.....	45
Capítulo 15: Páginas de título.....	49
Observaciones.....	49

Examples.....	49
Página de título del informe estándar.....	49
Capítulo 16: Presentación con paquete beamer.....	50
Parámetros.....	50
Observaciones.....	50
Examples.....	50
Diapositiva simple del título de un autor.....	50
Diapositiva de título de autor y afiliación múltiple.....	51
Capítulo 17: Tipografía matemática.....	53
Introducción.....	53
Sintaxis.....	53
Observaciones.....	53
Examples.....	53
Ecuaciones básicas.....	53
Encontrar símbolos.....	54
Paquetes disponibles para su uso.....	55
Buenos comandos para saber.....	56
Creando nuevos símbolos.....	57
Matrices.....	58
Creditos.....	59

Acerca de

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

Capítulo 1: Empezando con el látex

Observaciones

Látex

¿Qué es LaTeX?

LaTeX (pronunciado *lay-tech* o *lah-tekh*) es un lenguaje de marcado para escribir documentos de manera similar a cómo HTML es uno para los sitios web.

LaTeX tiene ventajas sobre los editores de lo que ves que es lo que obtienes (WYSIWYG) como Microsoft Word porque con LaTeX proporcionas el contenido y LaTeX se encarga del diseño. Separación del contenido de los resultados de la composición en documentos con un formato coherente y hermoso. Además, debido a que la marca LaTeX es de formato de texto simple (a diferencia de los formatos de archivos más complejos producidos por editores WYSIWYG, como `.docx`), los archivos LaTeX son ligeros y se pueden mantener fácilmente bajo el [control de versiones](#).

Los documentos LaTeX generalmente se compilan en archivos PDF, de modo que la consistencia en el diseño se mantiene en diferentes visores y para la impresión.

LaTeX es especialmente popular en la escritura académica debido a su amplio soporte para la composición tipográfica de ecuaciones, cuadros y tablas de referencias cruzadas, y citas y bibliografías.

Versiones

Versión	Fecha de lanzamiento
LaTeX 2.09	1985-09-01
LaTeX 2e	1994-06-01

Examples

Instalación y configuración

Puede elegir entre las principales distribuciones de LaTeX:

- [TeX Live](#) (Windows, Linux y OS X), la distribución estándar y multiplataforma.
- [MacTeX](#) (Mac) Una versión empaquetada de TeX Live hecha para OS X con algunas herramientas específicas de Mac

- **MiKTeX** (Windows) Una distribución separada totalmente que

Todas las distribuciones son más o menos equivalentes en un mundo ideal. TeX Live tiene la ventaja de estar disponible en todas las plataformas y, por lo tanto, tiene mucho mejor soporte de la comunidad. MiKTeX puede aprovechar las características específicas de Windows. Por razones de licencia, MiKTeX también distribuirá algunos paquetes que TeX Live no distribuirá.

En todos los casos, se recomienda la instalación completa. Específicamente, el uso de la función de descarga bajo comando de MiKTeX bloqueará / bloqueará a muchos editores.

Instalación

Windows (TeXLive)

1. Descargue la versión más reciente de TeXLive `install-tl-windows.exe` de su [sitio web](#) .
2. Ejecute `install-tl-windows.exe` y siga las instrucciones.

Windows (MiKTeX)

1. Descargue el instalador MiKTeX más reciente de su [sitio web](#) .
2. Ejecuta el instalador y sigue las instrucciones.

Mac OS X (TeXLive)

1. Descarga el MacTeX más reciente de su [sitio web](#) .
2. Ejecute `MacTeX.pkg` y siga las instrucciones.

Linux (TeXLive)

Los usuarios de Linux tienen dos opciones:

1. Instale a través del gestor de paquetes de su distribución (normalmente varias versiones atrás)
2. Instalar desde aguas arriba (publicado anualmente, actualizado a menudo)

Uso de gestores de paquetes

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

Tenga en cuenta que usar este método significa que dependerá del mantenedor de ese paquete para la distribución de las actualizaciones. Estos paquetes a menudo serán varios lanzamientos detrás de la distribución más reciente, lo que significa que faltarán actualizaciones críticas. Casi siempre es mejor instalar desde el principio. También tenga en cuenta que el administrador de

paquetes de la distribución probablemente no reconocerá la instalación directa y podría intentar instalarla cuando se instalen otros paquetes de soporte relacionados.

Instalación desde aguas arriba

1. Descargue la versión más reciente de TeXLive `install-tl-unx.tar.gz` desde su [sitio web](#) .
2. Extraiga los archivos del archivo comprimido con `tar -zxvf install-tl-unx.tar.gz` .
3. Cambie a la carpeta descargada con `cd install-tl-unx` .
4. Ejecute `./install-tl` y siga las instrucciones.

TeXLive ahora debe instalarse en `/usr/local/texlive/YEAR/` , donde `YEAR` es el año de cuatro dígitos (por ejemplo, `2016`). De esta manera, es posible tener varias versiones de TeXLive juntas y cambiar entre ellas cambiando la variable `PATH`.

Abra esta carpeta y compruebe la carpeta `bin` . Debe contener una subcarpeta, que (dependiendo de su plataforma) será algo como `i386-linux` o `x86_64-linux` .

5. Agregue la carpeta binaria TeX Live a su ruta con

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

donde `YEAR` es el año de cuatro dígitos (por ejemplo, `2016`), y `PLATFORM` es su plataforma (por ejemplo, `x86_64-linux`).

Instalación de prueba

La instalación de LaTeX ahora está completa. Para probarlo, cree un nuevo archivo con su editor de texto favorito, `test.tex` nombre `test.tex` y agregue el siguiente contenido:

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

Ahora, abra la consola o el terminal, navegue a la carpeta donde guardó `test.tex` y ejecute

```
pdflatex test
```

(Tenga en cuenta que su editor puede tener facilidades para ejecutar esto por usted).

Esto crea varios archivos nuevos, incluyendo `test.pdf` . Este es el documento de salida, y se ve así:

Hello World!

- El editor [Vim](#) con el plugin [LaTeX-suite](#) .
- [Texmaker](#) - un IDE especializado en LaTeX.
- [TeXstudio](#) - otro IDE de LaTeX.
- [TeXworks](#) - un IDE más de LaTeX.

Si bien los usuarios experimentados de Emacs o Vim pueden querer atenerse a su editor (cuyos complementos proporcionan una gran cantidad de funcionalidades que no están disponibles en otros lugares), un IDE especializado puede ser más fácil de instalar / usar para principiantes. Los últimos tres en la lista tienen una función de vista previa donde se pueden ver los resultados de la compilación del documento.

Adicionalmente, existen herramientas de LaTeX en línea que pueden ser de utilidad para los principiantes o personas que deben colaborar, por ejemplo [ShareLaTeX](#) y [dorso](#) .

Lea [Empezando con el látex en línea](#): <https://riptutorial.com/es/latex/topic/2269/empezando-con-el-latex>

Capítulo 2: Accediendo a la documentación de los paquetes de LaTeX.

Examples

CTAN

La [red completa de archivos TeX](#) (CTAN) es, de hecho, el repositorio completo de los paquetes LaTeX. La mayoría, si no todos los paquetes de calidad (y más) están allí, y todos los buenos incluyen documentación.

1. Ingrese el nombre del paquete en la barra de búsqueda.



2. Seleccione el paquete de la lista.

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. Accede a los documentos documentales.

BibL^AT_EX – Sophisticated Bibliographies in L^AT_EX

BibL^AT_EX is a complete reimplementación 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).

Importante: CTAN tiene las versiones más recientes. ¡Si su instalación está desactualizada, la documentación no coincidirá! En ese caso, consulte los documentos de documentación enviados

con su distribución de LaTeX.

TeX Live - texdoc

Si usa la distribución [TeX Live](#) , puede usar el programa de línea de comandos `texdoc` . Por ejemplo,

```
texdoc biblatex
```

Se abrirá la documentación del paquete `biblatex` .

O si no tiene conocimientos de la línea de comandos, puede encontrar lo mismo en línea en <http://www.texdoc.net/>

Lea [Accediendo a la documentación de los paquetes de LaTeX. en línea:](#)

<https://riptutorial.com/es/latex/topic/5820/accediendo-a-la-documentacion-de-los-paquetes-de-latex->

Capítulo 3: Añadir cita

Examples

Añadir cita al documento LaTeX ya existente

Al final del documento agregue lo siguiente:

`\bibliographystyle { estilo }`

`\bibliography { ubicación del archivo }`

Cree un archivo con la extensión `.bib` y guarde la cita de la siguiente manera:

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

Para citar usa lo siguiente: **`\citet { citation_name }`**

Lea Añadir cita en línea: <https://riptutorial.com/es/latex/topic/8357/anadir-cita>

Capítulo 4: Clases de documentos

Sintaxis

- `\documentclass {...}`

Observaciones

Este tema tiene como objetivo explicar los diferentes tipos de documentos y sus especificidades.

Una buena forma de organizarlo sería 1 ejemplo por tipo.

Examples

Artículo

```
\documentclass{article}
```

¿Cuándo usar la clase de artículo?

Para artículos en revistas científicas, presentaciones, informes breves, documentación del programa, invitaciones, ... ¹

¿Cuáles son las especificidades de esta clase?

Un artículo no contiene capítulos o partes. Se puede dividir en secciones, subsecciones y párrafos, etc.

De forma predeterminada, el título se muestra en la parte superior de la primera página, y no en una página de título separada.

Ejemplo simple

```
\documentclass{article}

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

\begin{document}
```

```
\maketitle  
  
Hello, World!  
\end{document}
```

Beamer

```
\documentclass{beamer}
```

¿Cuándo usar la clase beamer?

Para diapositivas de presentación.

¿Cuáles son las especificidades de esta clase?

La salida está orientada al paisaje. El documento está separado en "marcos" (diapositivas).

Ejemplo simple

El siguiente ejemplo fue adaptado 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}
```

Definiendo la clase de documento.

La primera línea en cada uno de sus programas LaTeX debe hacer esto. Debe seguir el formulario `\documentclass{...}`. Lo que pones dentro de las llaves es muy importante. Algunas

clases de documentos le dan comandos adicionales para usar, otras usan un formato diferente y todas tienen parámetros específicos que puede ingresar (descritos en la sección de parámetros).

Lea Clases de documentos en línea: <https://riptutorial.com/es/latex/topic/7458/clases-de-documentos>

Capítulo 5: Contadores, si enunciados y bucles con látex.

Examples

Contador declaración, inicialización e impresión a pdf.

Es posible utilizar variables enteras con látex. Para crear una nueva variable necesitamos el `\newcounter{name}` , donde `name` es el nombre del nuevo contador. El `name` debe contener sólo letras. Este comando crea uno nuevo con el nombre `\thename` . Con este comando podemos imprimir la variable del `name` en el papel. El valor inicial del `name` es 0. Para dar valor a "nombre" podemos usar `\setcounter{name}{n}` donde `n` es un número entero. `\value{name}` es una función que regresa con el valor 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

‡

Operaciones con contadores.

Este ejemplo muestra cómo usar operaciones matemáticas con contadores. Puede ser útil para los bucles de látex.

Adición: `\addtocounter{num}{n}`

este comando agrega n a num , donde num es un contador y n es un entero positivo.

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

este comando resta n de num , donde num es un contador y n es un entero positivo.

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

este comando multiplica num por n , donde num es un contador y n es un entero.

División `\divide\value{num} by n`

este comando divide num por n y obtiene la parte entera del cociente (num es un contador y n es un entero)

```
\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}` declara contador. `\setcounter{num}{3}` establece el valor `\setcounter{num}{3}` en 3.

`\addtocounter{num}{10}` agrega 10 a num .

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

`\stepcounter{num}` agrega 1 a num

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

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

`\divide\value{num} by 60` divide num por 60 y obtiene la parte entera.

El resultado del código: 13 \ 10 \ 11 \ 121 \ 242 \ 4

(\\ simboliza nueva línea)

El paquete intcalc agrega algunas otras operaciones de enteros, por ejemplo, mod, pow, sng, abs, inv ...

[intcalc_package.pdf](#)

Si declaraciones

En el látex podemos usar comandos incorporados para ejecutar el código si las condiciones son verdaderas o no.

Comparando dos enteros: `\ifnum\value{num}>n {A} \else {B}\fi`

Este código ejecuta A si num. N más B. Podemos sustituir > con < y =.

Si un número es impar: `\ifodd\value{num} {A}\else {B}\fi`

Si num es impar entonces ejecuta A else B.

Si con condición: `\ifthenelse{condition}{A}{B}`

Tenemos que cargar el paquete para usar este comando. Si la condición es verdadera, entonces se ejecuta A else B.

Es posible crear una condición compleja con `\(\)`, `\AND`, `\OR`, `\NOT`.

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

Esta pieza de código escribe "B" en la página. `\NOT 4<2` es verdadero y `4>11` es falso. Si conectamos una declaración falsa y una verdadera con "O", el resultado es verdadero. Entonces `\(\NOT 4<2 \OR 4>11\)` es verdadero. `\isodd{4}` es falso porque 4 es par. Una declaración falsa y verdadera conectada con "AND" es falsa, por lo que la salida es B.

Un código de ejemplo:

```
\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.

Bucles - repitiendo cosas

Podemos crear bucles en l atex. Son similares pero no tan personalizables como los bucles en otros lenguajes de programaci on. Una alternativa para usar loops es @loops. Si usamos un comando que incluye "@" en su nombre, debemos colocarlo entre \makeatletter y \makeatother . No est a permitido usarlos en una macro que describe una nueva definici on.

Incorrecto:

```

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

```

Derecha:

```

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

```

@for loop: \@for\command:={list}\do{commands}

Ejemplo :

```

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

```

Crea el siguiente texto: El sol est a saliendo. El sol se esta poniendo.

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

Ejemplo :

```

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

```

Este c odigo escribe n umeros impares del 1 al 9.

"repetici on de bucle" bucle: \loop {commands} \ifnum condition \repeat

Ejecuta comandos hasta que la condición sea verdadera.

Ejemplo

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

Este código hace lo mismo que @whilenum loop.

Un código de ejemplo:

```
\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
```

Usando bucles en Tikz

Los bucles son útiles en Tikz.

El siguiente código dibuja un reloj sin números:

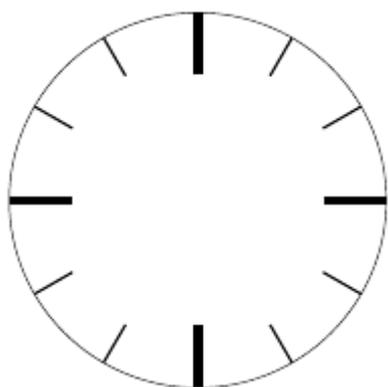
```

\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}

```

El resultado:



Lea Contadores, si enunciados y bucles con látex. en línea:

<https://riptutorial.com/es/latex/topic/9224/contadores--si-enunciados-y-bucles-con-latex->

Capítulo 6: Creación de carteles utilizando beamer.

Introducción

Crear un póster con el paquete beamerposter es muy similar a crear un solo cuadro. Poner el contenido en columnas. Dentro de cada columna, separe el contenido utilizando bloques.

Examples

Orientación y tamaño.

Mientras agrega el paquete de beamerposter, proporcione los parámetros requeridos.

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

También puede personalizar el tamaño del cartel.

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

Las dimensiones de altura y anchura aquí, están en cms. La `scale` se utiliza para el tamaño de fuente.

Esquema básico de un póster de proyector.

En orientación horizontal

```
\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/es/latex/topic/10714/creacion-de-carteles-utilizando-beamer->

Capítulo 7: Creando una bibliografía

Sintaxis

- Para una bibliografía con formato manual, no es necesario tener citas - `\cite` - dentro del documento.

Parámetros

Parámetro	Detalle
<code>thebibliography</code>	Este entorno establece el alcance de la bibliografía real. Define un entorno similar a una lista en el que puede usar <code>\bibitem</code> para establecer un elemento de bibliografía.
<code>{x}</code>	El entorno <code>thebibliography</code> toma un solo argumento que representa el elemento más amplio que se espera en la enumeración de <code>\bibitem</code> s. Para menos de 10 entradas, use un solo carácter / dígito; Para menos de 100 entradas, use dos caracteres / dígitos, ...
<code>\bibitem{<a>}</code>	Establezca el elemento de bibliografía <code></code> y haga que esté disponible para <code>\cite</code> dentro del documento usando la etiqueta <code><a></code> .

Examples

Bibliografía básica sin paquetes (formateo manual).

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{<biblabeled>} <citation>
\end{thebibliography}

\end{document}
```

```

\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}

```

Tenga en cuenta que a menos que sepa realmente *por qué*, probablemente no debería hacer esto. Usar paquetes designados (ver otros ejemplos) es preferible.

Bibliografía básica con biber.

Para comenzar una bibliografía necesitas definir tus fuentes. Cree un [archivo de base de datos](#) (como `sources.bib`) e incluya algo de contenido:

```

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

```

Luego puede incluir su archivo de base de datos en su documento principal y citar la nueva fuente (`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}

```

Para compilar el documento, deberá ejecutar 3 comandos en secuencia:

1. `pdflatex` para crear un archivo auxiliar que le dice a Biber qué fuentes son necesarias
2. `biber` para crear un archivo auxiliar con todas las fuentes que puede utilizar `pdflatex`
3. `pdflatex` para incluir el archivo auxiliar y crear el PDF.

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

References

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

Encuentre muchas más opciones y campos adicionales para archivos de bib en la [documentación del paquete en CTAN](#) .

Lea [Creando una bibliografía en línea](https://riptutorial.com/es/latex/topic/3488/creando-una-bibliografia): <https://riptutorial.com/es/latex/topic/3488/creando-una-bibliografia>

Capítulo 8: Definiendo macros

Sintaxis

- `\newcommand {\macro} {texto de reemplazo}`
- `\newcommand {\macro} [argcount] {texto de reemplazo}`
- `\renewcommand {\macro} {texto de reemplazo}`
- `\renewcommand {\macro} [argcount] {texto de reemplazo}`

Parámetros

Parámetro	Detalles
<code>\macro</code>	La macro a definir.
<code>argcount</code>	El número de argumentos que espera la macro (opcional)
<code>replacement text</code>	El texto de reemplazo para la macro. Dentro de ese texto <code>#1</code> , <code>#2</code> etc. se reemplazan con los argumentos de macro.

Examples

Definición básica de macros

Definir un nuevo comando básico.

Una macro se puede definir usando `\newcommand` . Por ejemplo:

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

define una macro `\foo` que se expande a `Just foo, you see?` . Luego puede usarse como cualquier comando incorporado, por ejemplo, después de esa definición:

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

se expande a

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

Definir un nuevo comando con argumentos.

Las macros también pueden tener argumentos. El número de argumentos se proporciona como

argumento opcional entre el nombre del comando y el texto de reemplazo. En el texto de reemplazo, se accede a los argumentos con #1 , #2 etc. Por ejemplo:

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

Redefiniendo un comando existente

Si ya se ha definido una macro, `\newcommand` da un error. Para dar una nueva definición para un comando existente, se utiliza `\renewcommand` su lugar. Aparte del nombre diferente, la sintaxis es exactamente la misma. Por ejemplo, después de la definición de `\foo` anterior, uno podría usar:

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

Después de esa redefinición, la macro `\foo` ya no se expande a `Just foo, you see?` Pero a `Another foo, please.`

Lea [Definiendo macros en línea](https://riptutorial.com/es/latex/topic/7658/definiendo-macros): <https://riptutorial.com/es/latex/topic/7658/definiendo-macros>

Capítulo 9: Dibujando graficas

Examples

TikZ - Diseño manual

El paquete [TikZ se](#) presta muy bien para dibujar gráficos.

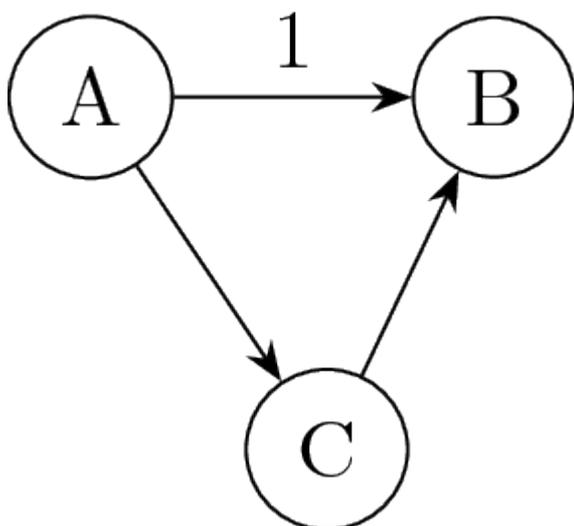
Este es un pequeño ejemplo (requiere TikZ 3.0+):

```
\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}
```



Puedes crear gráficas arbitrariamente complejas; Tenga cuidado con el código largo, sin embargo. Recuerde que hay `\foreach` y tome nota de todas las opciones de posicionamiento y estilo (consulte el manual de TikZ, secciones 13 a 17).

TikZ - Especificaciones gráficas

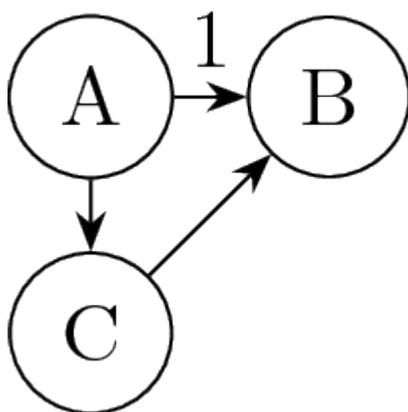
TikZ proporciona una sintaxis similar a [DOT](#) que puede utilizar para ajustar considerablemente el

código de dibujo de su gráfico.

```
\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}
```

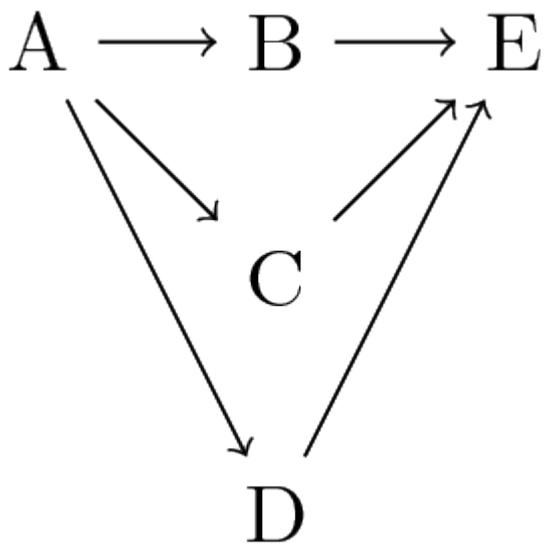


Como puede ver, cambia el control de grano fino para una sintaxis más fácil. La biblioteca de `graphs` realmente brilla cuando se especifican gráficos más complicados:

```
\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}
```



Hay muchas más opciones y gráficos predefinidos; Consulte la sección 19 del manual de TikZ.

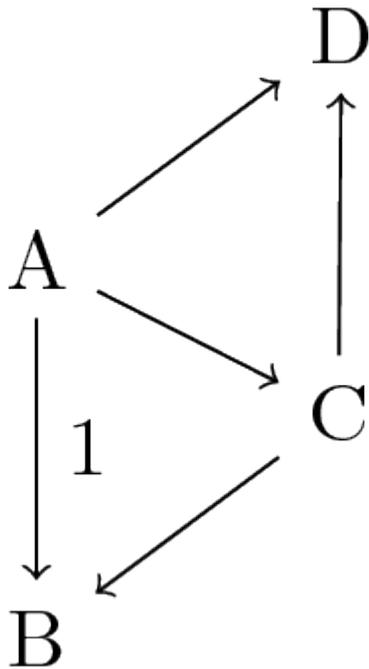
TikZ - Dibujo de gráfico algorítmico

TikZ implementa varios algoritmos para diseños de gráficos *automáticos* (requiere 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}
```



Hay varios algoritmos y muchas opciones para influir en ellos. Vea la parte IV del manual de TikZ para más detalles.

Diagrama de transición de estado de una cadena de Markov

Supongamos que la siguiente matriz es la matriz de probabilidad de transición asociada con una cadena de Markov.

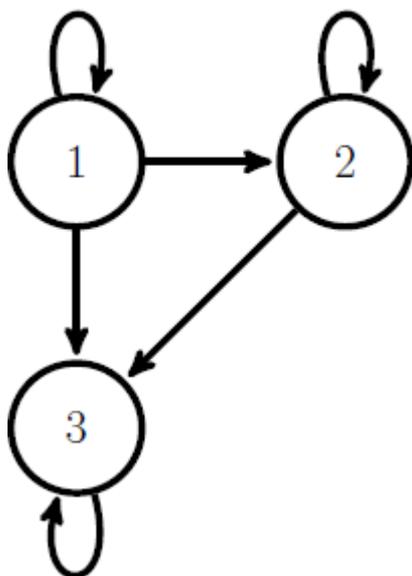
$$P = \begin{pmatrix} 0.5 & 0.2 & 0.3 \\ 0.0 & 0.1 & 0.9 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

Para estudiar la naturaleza de los estados de una cadena de Markov, se dibuja un diagrama de transición de estado de la cadena de Markov.

```

\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}
```



Lea Dibujando graficas en línea: <https://riptutorial.com/es/latex/topic/5955/dibujando-graficas>

Capítulo 10: Encabezado y pié de página

Examples

Usando los paquetes fancyhdr y 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}
```

1 F

1.1 C

Lorem
vestibul
mauris.
magna.
tique se
Cras vi
ultrices
est, iacu
ces bibo
at, mol
risus. D
eget ord

Nam
auctor l
et, tellu
magna,
Suspend
natoque
Aliquan
cursus l

Null
at, tinc
ummy p
massa a
leo. Ma
cipit a,
lorem. S
Integer
in tellus
eu enim

Capítulo 11: Formato de texto

Examples

Enfatizar texto

Para enfatizar el texto, se puede usar el comando `\emph` que generalmente muestra el texto en cursiva:

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

Huelga a través del texto

El comando `\sout` del paquete `ulem` golpea a través de un texto:

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

El paquete `ulem` redefine el comando `\emph`. Cuando no quiera tener este comportamiento, puede usar el paquete `ulem` con la opción `normalem`:

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

Texto en negrita

Para escribir texto en negrita, use `\textbf`:

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

Lea Formato de texto en línea: <https://riptutorial.com/es/latex/topic/7245/formato-de-texto>

Capítulo 12: Grabado Partituras

Examples

LilyPond

El grabador de notación LilyPond se puede utilizar con LaTeX a través del comando `lilypond-book`. Primero, creamos un documento LaTeX (con la extensión de archivo `.lytex`) para incrustar nuestra música en:

```
\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}
```

Luego creamos nuestro archivo LilyPond (`.ly`), incluido el archivo `lilypond-book-preamble.ly` (que LilyPond sabrá cómo encontrar):

```
\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)
  }
}
}
```

para construir, luego ejecutamos el comando `lilypond-book` :

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

que generará un PDF que contiene su música grabada con LilyPond:

con affetto



As I

4



I heard

Capítulo 13: Herramientas de construcción

Examples

Arara

Arara es una herramienta de automatización multiplataforma especialmente diseñada para TeX. Se incluye en una distribución estándar, por lo que no es necesario instalar nada adicional. Se entiende mejor como un medio para grabar las instrucciones de compilación en el propio archivo TeX:

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

Estos pueden ser mucho más complicados, sin embargo:

```
% 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}
```

Lea Herramientas de construcción en línea:

<https://riptutorial.com/es/latex/topic/5015/herramientas-de-construccion>

Capítulo 14: Mesas

Examples

El ambiente tabular.

El entorno `tabular` es la forma más básica de crear una tabla en LaTeX y no requiere ningún otro paquete.

```
\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

El parámetro (`|lcr||` en el ejemplo) se llama la **especificación de la tabla** y le dice a LaTeX cuántas columnas hay y cómo deben formatearse. Cada letra representa una sola columna. Los valores posibles son:

Personaje	Sentido
l	columna alineada a la izquierda
do	columna centrada
r	columna alineada a la derecha
p {'ancho'} por ejemplo p{5cm}	columna de párrafo con ancho definido
(personaje de la pipa)	línea vertical
(2 tubos)	2 líneas verticales

Las células están separadas por el carácter `&`. Una fila termina con 2 barras diagonales `\\`.

Las líneas horizontales se pueden insertar usando el comando `\hline`.

Las tablas siempre están formateadas para ser lo suficientemente amplias como para incluir todo el contenido. Si una tabla es `overfull hbox` grande, LaTeX imprimirá advertencias de `overfull hbox`. Las posibles soluciones incluyen el uso del especificador `p{'width'}` u otros paquetes como `tabularx`.

Se puede crear una tabla con encabezados de columna que abarcan varias columnas usando el comando `\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

Tenga en cuenta que el comando `\multicolumn` tiene tres argumentos obligatorios: el primer argumento especifica el número de columnas sobre las que se extiende el encabezado; el segundo argumento especifica la posición del encabezado (l, c, r) ; y el tercer argumento es el texto para el encabezado. El comando `\cline{2-4}` especifica la columna de inicio (aquí, 2) y la columna de finalización (aquí, 4) sobre la cual se dibujará una línea.

Mesa para colorear

Para hacer que la tabla sea más legible, las siguientes son las formas de colorearla:

1. Filas
2. Columnas
3. Líneas
4. Células

Filas de colorear

Use `\rowcolor` (proporcionado por `colortbl` ; también cargado por `xcolor` en la opción de paquete `[table]`). Ejemplo:

```
\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

Columnas para colorear

Las columnas se pueden colorear de las siguientes maneras:

- Definiendo la propiedad de color de la columna fuera de la etiqueta de la tabla usando

`\newcolumntype` :

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

- Definir la propiedad de color de la columna dentro de los parámetros de la tabla.

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

Ejemplo:

```
\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

Líneas para colorear

Utilice `\arrayrulecolor` . Ejemplo:

```
\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

Células para colorear

Utilice `\cellcolor` . Ejemplo:

```
\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

Podemos definir nuestros propios colores también usando el paquete `colortbl` . Los siguientes son los ejemplos de etiquetas:

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

Lea Mesas en línea: <https://riptutorial.com/es/latex/topic/4956/mesas>

Capítulo 15: Páginas de título

Observaciones

`\title{<title>}` , `\author{<author>}` y `\date{<date>}` almacenan el contenido internamente.

`\maketitle` produce una página de título estándar con los valores definidos previamente.

Examples

Página de título del informe estándar

```
\documentclass{report}

\begin{document}

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

\end{document}
```

Esto creará una página de título sin otro contenido:

I want to be a Wombat

Carl Capybara

July 26, 2016

Lea Páginas de título en línea: <https://riptutorial.com/es/latex/topic/3010/paginas-de-titulo>

Capítulo 16: Presentación con paquete beamer.

Parámetros

tema	Ann Arbor
tema de color	seahoarse

Observaciones

Para otros temas y colores puedes visitar [aquí](#).

Examples

Diapositiva simple del título de un autor

```
\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,

<https://riptutorial.com/es/latex/topic/10677/presentacion-con-paquete-beamer->

Capítulo 17: Tipografía matemática

Introducción

Una de las mayores ventajas de LaTeX es su habilidad para componer ecuaciones. Aquí se describen los fundamentos de las ecuaciones de composición tipográfica, algunos de los diversos paquetes que se pueden usar, así como los símbolos comunes.

Sintaxis

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

Observaciones

Aquí hay algunas ideas básicas para asegurarse de que su código no se rompa en usted y que sus ecuaciones se vean mejor:

1. Asegúrese de que todos los corchetes, llaves, signos de dólar y los comandos `\begin{}` `\end{}` coincidan. Esto es algo en el que un pequeño error puede desordenar todo tu código de manera considerable.
2. Si recibe errores, asegúrese de tener el paquete correcto cargado (por ejemplo, no use el comando `\begin{equation*}` sin el paquete `amsmath`).
3. Nunca, nunca, *nunca* use signos de dólar doble (`$$an equation here$$`) en lugar de `\begin{equation}`.
4. Nunca use el modo matemático como una forma de hacer que su texto esté en cursiva.
5. Completamente atascado? Pruebe [TeX.SX](#), un sitio para responder preguntas sobre TeX, LaTeX y otros idiomas relacionados.

¡Buena suerte!

Examples

Ecuaciones básicas

Ecuaciones simples, en línea

Puedes hacer una ecuación en línea simple usando `$an equation here$`.

Por ejemplo, podrías hacer

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

el cual, si ponemos un pequeño texto falso a su alrededor, da

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

Numeradas, Ecuaciones Centradas

Al escribir documentos u otros documentos, a veces es preferible que sus ecuaciones estén centradas y numeradas, en lugar de en línea. Luego, use los comandos `\begin{equation}` y `\end{equation}`.

Por ejemplo, si usamos el código.

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

Y añada un pequeño texto a su alrededor, obtenemos

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

Puede eliminar la numeración de la ecuación utilizando `\begin{equation*}` y `\end{equation*}`.

Por ejemplo, si usamos el código.

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

y añadir un poco de texto a su alrededor, obtenemos

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

(aunque debe tenerse en cuenta que tiene que usar el paquete `amsmath` para esto).

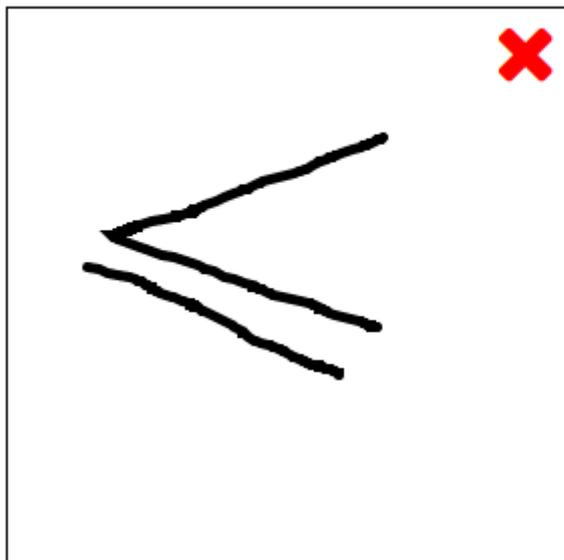
Encontrar símbolos

A veces, puede ser difícil encontrar el símbolo matemático que necesita. Hay varias opciones aquí. La primera (y la más rápida) es usar [Detexify](#), donde dibuja el símbolo que desea, y trata de encontrar lo que desea, como se muestra a continuación:

Detexify

classify

symbols



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



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



Score: 0.11892004279045627
`\usepackage{ tipa }`
`\text{rptr}`
textmode

Otra opción es usar la lista completa de símbolos LaTeX, que se puede encontrar [aquí](#) . Si está utilizando el paquete `unicode-math` [esta lista](#) de todos los símbolos compatibles puede ser útil. Otra opción es [este sitio web](#) , que tiene símbolos matemáticos comunes.

Paquetes disponibles para su uso

Si bien LaTeX estándar es todo lo que se necesita para las fórmulas y ecuaciones matemáticas más simples, a veces se necesitan más símbolos y herramientas. Hay varios paquetes disponibles que mejorarán sus ecuaciones y le brindarán más para trabajar. Tres de los paquetes principales se describen a continuación. Recuerde, para cargar un paquete, escriba `\usepackage{package}` en el preámbulo de su documento.

`amsmath`

El paquete `amsmath` es un paquete increíblemente útil. Se usa para permitir que sus ecuaciones estén centradas pero no numeradas, como en `\begin{equation*}` , se usa para crear matrices (como se describe a continuación) e introduce muchos otros comandos útiles, como `\overset` y `\underset` , descrito abajo. La documentación del paquete `amsmath` se puede encontrar [aquí](#) .

`mathtools`

El paquete `mathtools` basa en el paquete `amsmath` , y agrega más símbolos y herramientas útiles. Carga automáticamente el paquete `amsmath` , por lo que no necesita cargar ambos en el preámbulo de su documento. La documentación de `mathtools` se puede encontrar [aquí](#) .

`amssymb`

El paquete `amssymb` proporciona muchos símbolos adicionales que pueden ser muy útiles para ecuaciones más complejas. La documentación de `amssymb` se puede encontrar [aquí](#) .

Paquetes de fuentes

También hay varias fuentes que puede usar para sus ecuaciones, como se describe en [esta pregunta](#) en el intercambio de pila TeX, para TeX, LaTeX y amigos.

[Este documento](#) es una explicación concisa de las diferentes características proporcionadas por algunos paquetes, así como el estándar LaTeX; es de mucha ayuda.

Buenos comandos para saber

Algunos de los comandos más comunes incluyen:

- **Fracciones y raíces cuadradas:** Para fracciones, use `\frac {numerator}{denominator}` . Para las raíces cuadradas, use `\sqrt [root] {number}` .
- **Letras griegas:** use los comandos dados en la tabla de abajo:

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>

- **Operadores:** `\leq` da el símbolo menor o igual que, `\geq` da el símbolo mayor o igual que `\neq` el símbolo no es igual, `\sum` da el símbolo de suma, `\partial` da el símbolo de derivada parcial, `\nabla` da el operador laplaciano, `\times` da el producto cruzado o símbolo de multiplicación, `\cdot` da el producto punto o símbolo de multiplicación, y `\int` da el símbolo integral.
- **Flechas:** `\rightarrow` y `\leftarrow` dan flechas derecha e izquierda, respectivamente.
- **Porcentajes:** si escribe% en LaTeX, es importante incluir una barra invertida, `\%` ya que el símbolo de porcentaje se usa normalmente para los comentarios.
- **Superíndices y subíndices:** para hacer un superíndice, puede escribir `x^2` , o, para

superíndices más largos, x^{2x} . Para hacer un subíndice, puede escribir x_a , o, para subíndices más largos, x_{ab} .

- **Negrita:** use `\boldmath{...}` para hacer que sus símbolos matemáticos `\boldmath{...}` negrita. Otras opciones se dan en [esta pregunta TeX.SX](#). Los símbolos matemáticos son automáticamente en cursiva; Si no quieres que esto sea cierto, haz el texto de tu ecuación como se describe a continuación.
- **Infinito:** Para escribir infinito, usa el comando `\infty` .
- **Mover elementos sobre o debajo de otro:** primero, solo para operadores matemáticos, hay un método alternativo. Puede escribir el operador matemático, decir `\int` , y luego usar el comando `\limits` . Un ejemplo es `\int\limits_{\infty}` o `\int\limits^{\infty}` . Luego, para los casos normales, puede hacer `\overset{top}{normal}` o `\underset{bottom}{normal}` . Esto puede ser muy útil para hacer vectores. Por ejemplo, es posible hacer `\vec{x}` El `amsmath` paquete se necesita para `overset` y `underset` .
- **Aparatos ortopédicos:** debido a que los paréntesis se utilizan en los comandos, es necesario escribir `\{ o \}` para obtener los tirantes.
- **Texto:** para incluir texto en ecuaciones, escriba `\usepackage{amsmath}` en el preámbulo y luego escriba `\text{...}` .
- **Espacio:** para agregar espacio en sus ecuaciones, escriba `\quad` entre los dos elementos que desea separar (por ejemplo, podría tener `$2x \quad \cos`).

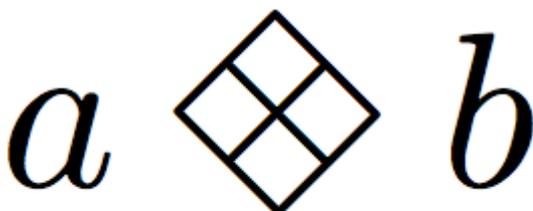
Creando nuevos símbolos

Digamos que no puedes encontrar el símbolo que necesitas en ninguna parte. Puede crear un símbolo personalizado. Por ejemplo, el código

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

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

crea y llama a un símbolo, dando



Este es un ejemplo más simple; simplemente tiene que rotar un símbolo ya existente. Sin embargo, puede crear símbolos más complejos.

Esta sección está en proceso de ser ampliada.

Matrices

Matrices

Siempre debe usar el paquete `amsmath` si va a usar los siguientes comandos. Hay cuatro tipos principales de matriz, como se muestra en el siguiente código:

```
\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}
```

Este código produce

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

Hay un par de cosas importantes a tener en cuenta sobre esto:

1. Es importante que coloque su matriz dentro de la `equation`, la `equation*` o el entorno `$....$`: el comando `bmatrix` no es un entorno matemático por sí solo.
2. La construcción de la matriz es en realidad bastante simple. Para cada fila, crea cada elemento (por ejemplo, `x_{11}`), luego coloca un `&` y luego escribe el siguiente elemento. Para varias filas, al final de cada fila ponga `\\` (no tiene que hacer esto para la última fila). Es bastante similar a una tabla en esto.

Lea Tipografía matemática en línea: <https://riptutorial.com/es/latex/topic/5950/tipografia-matematica>

Creditos

S. No	Capítulos	Contributors
1	Empezando con el látex	Community , eyqs , Harry , hbaderts , jani , Louis , Nijin22 , Sean Allred , SnoringFrog , Spacedman , Sumner Evans , TuringTux , tversteeg
2	Accediendo a la documentación de los paquetes de LaTeX.	Raphael , samcarter
3	Añadir cita	Hamzawey
4	Clases de documentos	Community , heather
5	Contadores, si enunciados y bucles con látex.	csekri
6	Creación de carteles utilizando beamer.	Sukanya B
7	Creando una bibliografía	Nijin22 , Raphael , Werner
8	Definiendo macros	celtschk , strpeter
9	Dibujando graficas	L.V.Rao , Raphael
10	Encabezado y pié de página	Sukanya B
11	Formato de texto	celtschk , Stephan Kulla
12	Grabado Partituras	Sam Whited
13	Herramientas de construcción	Sean Allred
14	Mesas	L.V.Rao , Nijin22 , Nikita Jain , Werner
15	Páginas de título	adn , Johannes_B , Nijin22 , Sam Whited
16	Presentación con	Sukanya B

paquete beamer.		
17	Tipografia matematica	heather