

```

\documentclass[11pt]{article}
\usepackage{amsmath}
\usepackage{bm}
\usepackage{tikz}

%Write an article
\title{An Article}
\author{Me}
\date{Today}

\begin{document}
\maketitle

\section{First Topic}
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla facilisis cursus justo, quis sodales orci tempus vitae. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam vitae ex nec ligula blandit pharetra nec pretium arcu. Nunc metus quam, iaculis vel mi nec, placerat ultricies nunc. Quisque magna sem, sodales eget tempus posuere, finibus ut quam. Integer feugiat nibh lectus, eget vestibulum est pellentesque vel. Vivamus commodo lorem metus, ac faucibus magna commodo sit amet. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam varius tempor odio, id finibus lectus. Suspendisse eget diam diam. Duis sed ex in ex blandit varius vitae id lorem. Duis suscipit leo non accumsan sagittis. Etiam quis auctor ligula.

\[\iint_{\Sigma}\nabla\!\times\!\mathbf{F}\cdot d\mathbf{bm}\{\Sigma\}=\oint_{\partial\Sigma}\mathbf{F}\cdot d\mathbf{bm}\{r\}.\]

\section{Second Topic}

```

LaTeX y Git aplicado a la investigación científica

Módulo 4 – Aspectos Avanzados de LaTeX

del 15 de marzo al 17 de mayo de 2022

AULA VIRTUAL



Manual: *"LaTeX y Git aplicado a la investigación científica"*.

Curso Virtual

Módulo 4 – Aspectos Avanzados de LaTeX

17 páginas.

Abril 2022. Elaborado por Pablo Hinojosa.

Asociación Darwin Eventur

Avanzando en contenido y forma

Acentos y UTF-8	1
Caracteres especiales	2
Tipos de letra	3
Más tipos de letra.....	5
Cambiado el tamaño de la letra	6
Unidades de medida.....	7
Listas.....	8
Primeras tablas.....	11
Introducción a las cajas.....	13
Cambiando los márgenes de la página	15
Indentado y espacios entre párrafos.....	16

Acentos y UTF-8

Por defecto, LaTeX sólo usa ASCII y, si se quieren poner acentos, debe hacerse del siguiente modo (utilizando el carácter "\" como escape):

```
\a \e \i \o \u
```

del mismo modo, por ejemplo, la letra ñ se pondría del siguiente modo:

```
\~n
```

La fórmula es similar para otros caracteres fuera del ASCII, como se puede ver en los siguientes ejemplos:

```
\"o \^u \AE \=e \.o \c c
```

Para poder usar tildes, eñes y otros caracteres no-ASCII directamente, hay que cargar el juego de caracteres correspondiente con *inputenc*. Es recomendable el uso de UTF-8, que se cargaría de este modo (en la cabecera de nuestro documento, claro).

```
\usepackage[utf8]{inputenc}
```



Otro paquete interesante a ese respecto es babel, que permitirá a LaTeX el manejo de las peculiaridades del español, como las reglas de separación de sílabas a final de línea o el manejo de signos de puntuación.

```
\usepackage[spanish]{babel}
```

De este modo, nuestro documento ya empieza a parecerse a esto:

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

```
\usepackage[utf8]{inputenc}
```

```
\begin{document}
```

Por fin ya podemos usar eñes, acentos, diéresis y esas tonterías.

```
\end{document}
```

Existen otros comandos para agregar caracteres especiales como, por ejemplo:

```
\ldots
```

Esto colocará tres puntos suspensivos.

Caracteres especiales

Algunos símbolos tienen un significado especial, y el compilador de LaTeX los interpreta como instrucciones. Este es el caso, por ejemplo, de la barra "\" para iniciar comandos, las llaves "{" y "}" para las opciones de estos o el signo del dólar "\$", que vimos en las fórmulas matemáticas.

Para evitar este significado especial y que se imprima el carácter en sí mismo en lugar de ser interpretado por el compilador (a esto se le llama "escapar" un carácter), lo que se hace normalmente es agregar una barra "\" delante, de este modo:

```
\{ \}, \$, ,, \%
```

Pero ¿Cómo escapamos la propia barra? (Hacer "\\" no sirve, porque sabemos que eso es una nueva línea).

Para ello tenemos el comando `\textbackslash` (en realidad, el código de un carácter especial), que inserta el signo \ en su lugar.



La cantidad de caracteres posibles es ilimitada porque, de hecho, se pueden definir nuevos. Por ejemplo: Instalando y cargando el paquete "*hieroglf*" se puede escribir en jeroglíficos egipcios, o con "*ccllicenses*" se podrían insertar los distintos símbolos de las licencias Creative Commons.

En este enlace puedes ver una [lista de caracteres para LaTeX \(PDF\)](#).

Tipos de letra

LaTeX soporta una serie de tipografías (tipos de letra) que pueden asignarse a un fragmento de texto. Para ello provee de los siguientes comandos:

`\rm` Roman (redonda serif, la tipografía normal)

`\em` Italic (cursiva)

`\bf` Boldface (negrita)

`\sl` Slanted (inclinada)

`\sf` Sans Serif (sin serif, de "palo seco")

`\sc` Small Caps (todas mayúsculas, solo cambia el tamaño)

`\tt` Typewriter (De "paso fijo" o monotype)

Cuando introducimos en el texto uno de estos comandos, modificará todo el texto subsiguiente, hasta que llegue a otro comando que vuelva a cambiarlo. Este no es un comportamiento demasiado eficiente, por lo que el modo adecuado para cambiar la tipografía un pequeño trozo de texto es encerrarlo entre llaves junto con el comando del siguiente modo:

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

```
\usepackage[utf8]{inputenc}
```

```
\usepackage[spanish]{babel}
```

```
\begin{document}
```

```
{\rm Roman} (redonda serif, la tipografía normal).
```

```
{\em Italic} (cursiva).
```

```
{\bf Boldface} (negrita).
```

`{\sl Slanted}` (inclinada).

`{\sf Sans Serif}` (sin serif, de "palo seco").

`{\sc Small Caps}` (todas mayúsculas, solo cambia el tamaño).

`{\tt Typewriter}` (De "paso fijo" o monotype).

`\end{document}`

Esto es válido si el texto a modificar es relativamente corto. Para partes mayores o más complejas es mejor usar (como no), un entorno:

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

```
\usepackage[utf8]{inputenc}
```

```
\usepackage[spanish]{babel}
```

```
\begin{document}
```

```
\begin{rm}
```

Roman (redonda serif, la tipografía normal). Usando un entorno, que es más cómodo en estas circunstancias.

```
\end{rm}
```

```
\begin{em}
```

Italic (cursiva). Usando un entorno, que es más cómodo en estas circunstancias.

```
\end{em}
```

```
\begin{bf}
```

Boldface (negrita). Usando un entorno, que es más cómodo en estas circunstancias.

```
\end{bf}
```

```
\begin{sl}
```

Slanted (inclinada). Usando un entorno, que es más cómodo en estas circunstancias.

```
\end{sl}
```

```
\begin{sf}
```

Sans Serif (sin serif, de "palo seco"). Usando un entorno, que es más cómodo en estas circunstancias.

```
\end{sf}
```

`\begin{sc}`

Small Caps (todas mayúsculas, solo cambia el tamaño). Usando un entorno, que es más cómodo en estas circunstancias.

`\end{sc}`

`\begin{tt}`

Typewriter (De "paso fijo" o monotype). Usando un entorno, que es más cómodo en estas circunstancias.

`\end{tt}`

`\end{document}`

Más tipos de letra

Otra opción para cambiar la tipografía es con la combinación de Series, Formas y Familias:

* Series:

`\mdseries` Medium (peso normal)

`\bfseries` Boldface (negrita)

* Formas:

`\upshape` Upright (Recta normal)

`\itshape` Italic (cursiva)

`\slshape` Slanted (inclinada)

`\scshape` Small Caps (todas mayúsculas, solo cambia el tamaño)

* Familias:

`\rmfamily` Roman (redonda serif, la tipografía normal)

`\sffamily` Sans Serif (sin serif, de "palo seco")

`\ttfamily` Typewriter (De "paso fijo" o monotype)

Aunque los nombres son algo más engorrosos, el uso es exactamente el mismo que hemos visto antes, con la salvedad de que estos comandos nos permiten combinar varias propiedades para, por ejemplo, escribir un texto en Sans Serif cursiva:

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

```
\usepackage[utf8]{inputenc}

\usepackage[spanish]{babel}

\begin{document}

{\sffamily \itshape Familia Sans Serif en cursiva.}

\begin{rmfamily}

\begin{bfseries}

Familia Roman en negrita

\end{bfseries}

\end{rmfamily}

\end{document}
```

Nota: Algunas combinaciones pueden no representarse adecuadamente en algunos equipos.

Cambiado el tamaño de la letra

Como recordarás, al principio de cada documento, en la declaración `\documentclass`, tenemos opción a indicar el tamaño de la fuente por defecto. Los valores permitidos son 10pt (que es también el valor por defecto), 11pt y 12 pt.

Pero también, del mismo modo que hemos visto con la familia o el peso de la letra, LaTeX permite cambiar el tamaño de los caracteres. Los tamaños disponibles son, de menor a mayor: `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge` y `\Huge`.

Nota: Si hemos elegido en `\documentclass` un tamaño de 12pt, los tamaños `\huge` y `\Huge` se verán igual.

El tamaño por defecto es, como su propio nombre indica, `\normalsize`. y será el que hayamos puesto en `\documentclass` o, en su defecto, de 10pt.

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

\usepackage[utf8]{inputenc}

\usepackage[spanish]{babel}
```




```
\begin{document}
```

```
{\scriptsize \itshape texto pequeño en cursiva.}
```

Texto normal, con la medida por defecto (11 pt en nuestro caso, porque lo hemos declarado en el documentclass)

```
\begin{Large}
```

Texto en largue y, por tanto, grandote

```
\end{Large}
```

```
\end{document}
```

Como puedes comprobar, se usa del mismo modo que ya hemos visto, y se puede combinar con las opciones de tipos y familias de letras.

Unidades de medida

A lo largo de este curso se han ido usando una serie de unidades para medir tamaños. Concretamente, se ha usado el punto (pt) que es una popular medida tipográfica y suele usarse para medir tamaños de letras y distancias o espacios en el texto, y el centímetro (cm), que forma parte del sistema métrico y no necesita ser explicado.

LaTeX admite el uso de una gran variedad de unidades, algunas de las cuales pueden ser más convenientes en según qué circunstancias:

mm (Milímetro) 1/1000 de metro en el sistema métrico decimal.

cm (Centímetro) 1/100 de metro en el sistema métrico decimal.

in (Pulgada) Equivalente a 2.54 cm. Muy popular en el ámbito anglosajón.

pt (Punto) Equivale a 0.351 mm y es una de las más clásicas unidades de medida tipográficas.

pc (Pica) 12 pt

bp (Big point) a 0.353 mm o 1/72 pulgadas (se le conoce como Punto Postscript)

dd (Didôt Point) Antigua medida tipográfica francesa de 0.376 mm

cc (Cicero) 12 dd

sp (Scaled point) 1/65536 de pt ¡No es una errata!

Además de todas las anteriores, que son absolutas y su valor no cambia nunca, también hay un juego de unidades relativas, que dependen del tamaño de la tipografía que se esté usando:

ex (Equis) Altura de la letra "x" de la letra que se esté usando en un momento dado.

em (Eme) Anchura de la letra "M" de la letra que se esté usando en un momento dado.

mu (μ) 1/18 de em.

Listas

Una lista es, básicamente una sucesión de elementos colocados unos después de otro. Las listas pueden ser "ordenadas" cuando tienen una numeración que indica un orden o "desordenadas" cuando no existe tal numeración.

Para hacer una lista ordenada de elementos se usa el entorno *enumerate* de este modo:

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

```
\usepackage[utf8]{inputenc}
```

```
\usepackage[spanish]{babel}
```

```
\begin{document}
```

```
\begin{enumerate}
```

```
\item Primer elemento de la lista
```

```
\item Segundo elemento de la lista
```

```
\item Tercer elemento de la lista
```

```
\end{enumerate}
```

```
\end{document}
```

Como se puede ver, cada uno de los elementos está precedido del comando `\item` (Recordemos que LaTeX ignora los espacios, la indentación en el ejemplo de arriba es solo por claridad en el ejemplo y no es necesaria).



Como ya nos tiene acostumbrados, LaTeX asigna por sí mismo los números (o letras, luego veremos) de orden, de modo que se reconstruirán automáticamente cada vez que haya algún cambio.

Naturalmente, las listas pueden anidarse unas dentro de otras para profundizar niveles. Para ello sólo hay que crear otro entorno *enumerate* dentro del anterior:

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

\usepackage[utf8]{inputenc}

\usepackage[spanish]{babel}

\begin{document}

\begin{enumerate}

\item Primer elemento de la lista

\item Segundo elemento de la lista

\begin{enumerate}

\item Primer elemento en el segundo nivel

\begin{enumerate}

\item Primer elemento en el tercer nivel

\item Segundo elemento en el tercer nivel

\end{enumerate}

\end{enumerate}

\item Segundo elemento en el segundo nivel

\end{enumerate}

\item Tercer elemento de la lista

\end{enumerate}

\end{document}
```

Como puedes ver en el resultado compilado, LaTeX identifica los distintos niveles asignándoles por sí mismo una forma diferente de numeración (este entorno admite hasta cuatro niveles de anidamiento).

Para listas sin ordenar (esto es: no numeradas) se utiliza el entorno *itemize* exactamente del mismo modo que hemos visto el anterior:

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

\usepackage[utf8]{inputenc}

\usepackage[spanish]{babel}

\begin{document}

\begin{itemize}

\item Primer elemento de la lista

\item Segundo elemento de la lista

\begin{itemize}

\item Primer elemento en el segundo nivel

\begin{itemize}

\item Primer elemento en el tercer nivel

\item Segundo elemento en el tercer nivel

\end{itemize}

\item Segundo elemento en el segundo nivel

\end{itemize}

\item Tercer elemento de la lista

\end{itemize}

\end{document}
```

Por supuesto, es posible combinar ambos entornos anidando unos dentro de otros:

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

\usepackage[utf8]{inputenc}

\usepackage[spanish]{babel}

\begin{document}

\begin{itemize}

\item Primer elemento de la lista
```

```
\item Segundo elemento de la lista

\begin{enumerate}

\item Primer elemento en el segundo nivel

\begin{itemize}

\item Primer elemento en el tercer nivel

\item Segundo elemento en el tercer nivel

\end{itemize}

\item Segundo elemento en el segundo nivel

\end{enumerate}

\item Tercer elemento de la lista

\end{itemize}

\end{document}
```

Primeras tablas

De modo similar a otros elementos, una tabla se inicia creando un entorno (tabular) con la orden `\begin{tabular}` y finaliza con `\end{tabular}`.

En `\begin{tabular}` hemos de definir también el estilo de la tabla, del siguiente modo:

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

En esta instrucción estamos definiendo dos columnas separadas por líneas verticales dobles (indicadas con el `"|"`) y rodeadas por líneas verticales simples (`"|"`). Si se hubiese puesto.

La letra `"c"` indica que el contenido de cada celda se centrará dentro de ella. otras opciones serían `"r"` para alinear a la derecha, y `"l"` para la izquierda.

Dentro de la tabla, cada fila se separa de las demás por medio de los signos `//` y las celdas dentro de una fila se separan mediante el signo `&`

Si queremos insertar líneas horizontales, lo haremos mediante la orden `\hline`.



De este modo, una tabla simple sería como sigue:

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

\begin{document}

\begin{tabular}{|c|c|}

\hline

Primera celda & Segunda celda \\

\hline

Abajo a la izquierda & Abajo a la derecha \\

\hline

\end{tabular}

\end{document}
```

Una tabla así creada se colocará en el lugar donde hayamos insertado el código. Pero LaTeX es mucho más potente y permite colocarla como un elemento flotante, de modo que sea él mismo el que ajuste dónde ubicarla en función del espacio o la distribución del resto de elementos. Para ello tenemos la orden `\begin{table}`.

Si envolvemos nuestra tabla anterior en este comando del siguiente modo:

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

\begin{document}

\begin{table}

\begin{tabular}{|c|c|}

\hline

Primera celda & Segunda celda \\

\hline

Abajo a la izquierda & Abajo a la derecha \\

\hline

\end{tabular}

\end{table}

\end{document}
```

`\end{document}`

La tabla será manejada como un elemento flotante, y ubicada en función del resto de elementos.

Introducción a las cajas

A la hora de distribuir los elementos en la página, LaTeX trabaja internamente con un mecanismo de "cajas", en la que los elementos se consideran cajas rectangulares, anidadas unas dentro de otras, que se ubican donde sea necesario. Cada letra está contenida en una caja que se agrupa con otras en la caja de la palabra, la cual a su vez está dentro de la caja de línea...

También es posible trabajar con estas cajas directamente.

El comando `\mbox` simplemente creará una caja con el texto que se le indique entre las llaves, y `\fbox` hará lo mismo, pero con un marco alrededor de la caja:

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

```
\usepackage[utf8]{inputenc}
```

```
\begin{document}
```

```
\mbox{este texto NO está enmarcado}
```

```
\fbox{pero este texto está enmarcado}
```

```
\end{document}
```

Para tener más control sobre estas cajas, se pueden usar los comandos `\makebox` y `\framebox`, que funcionan igual que los anteriores, pero permiten definir el tamaño de la caja y la alineación del texto en ella:

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

```
\usepackage[utf8]{inputenc}
```

```
\begin{document}
```

```
\makebox[15cm][r]{Caja de 15 centímetros con texto a la derecha}
```

```
\framebox[15cm][s]{Caja de 15 centímetros con texto justificado}
```

```
\end{document}
```



El primer parámetro opcional es el ancho de la caja (si es menor que el texto, este se saldrá fuera), y el segundo es la alineación del texto (r para derecha, l para izquierda, c para centrado y s para justificado).

Un par de comandos muy interesantes para este contexto son `\width`, que nos devuelve el ancho del elemento y `\height`, que nos da el alto. Ambos pueden ser usados como parámetros. Mucho más útil es `\linewidth`, que nos da el ancho total de la línea, y que se podría usar del siguiente modo:

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

\usepackage[utf8]{inputenc}

\begin{document}

\framebox[0.5\linewidth][c]{Caja de 1/2 del ancho total}

\framebox[0.5\linewidth][c]{Caja de 1/2 del ancho total}

\end{document}
```

Nota que, al multiplicar `\linewidth` por 0.5, obtenemos la mitad del ancho total de la línea.

Pero esto sólo nos permite hacer cajas de una línea. Para hacer cajas con párrafos completos necesitamos el entorno *minipage*.

El entorno *minipage* nos crea, como su propio nombre indica, una caja que actúa como una página en miniatura:

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

\usepackage[utf8]{inputenc}

\begin{document}

\begin{minipage}[b]{0.5\linewidth}

Una caja con minipage

\end{minipage}

\begin{minipage}[b]{0.5\linewidth}
```

Otra caja con minipage


```
\end{minipage}
```

```
\end{document}
```

Este entorno acepta muchas más opciones y modificadores. En capítulos sucesivos veremos más herramientas para ampliar y mejorar su uso, y el de las cajas en general.

Cambiando los márgenes de la página

Ya vimos cómo, al declarar el tipo de documento, podíamos asignar el formato del papel que usaríamos. Usando ese dato y el tipo del documento, y con la autonomía a la que ya nos tiene acostumbrados, LaTeX calculará automáticamente los márgenes necesarios en la página.

Pero, si por alguna razón queremos forzar otra medida, el paquete *anysize* puede solucionarlo de un modo simple y cómodo. Para ello debe llamarse al paquete en el preámbulo como ya hemos visto en otros casos:

```
\usepackage{anysize}
```

Después, sólo tenemos que llamar (también en el preámbulo) a la orden `\marginsize` del siguiente modo:

```
\marginsize{izquierda}{derecha}{arriba}{abajo}
```

Veámoslo con un ejemplo:

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

```
\usepackage[utf8]{inputenc}
```

```
\usepackage[spanish]{babel}
```

```
\usepackage{anysize}
```

```
\marginsize{2cm}{2cm}{2cm}{2cm}
```

```
\begin{document}
```

```
\tableofcontents
```

```
\section{Una sección}
```



```
\usepackage[spanish]{babel}
```

```
\parindent 2cm
```

```
\parskip=2cm
```

```
\begin{document}
```

Párrafo perfectamente normal, que colocamos sólo para poder ver el efecto que tienen los cambios en los márgenes y espacios. Naturalmente, harán falta dos o tres como este.

Párrafo perfectamente normal, que colocamos sólo para poder ver el efecto que tienen los cambios en los márgenes y espacios. Naturalmente, harán falta dos o tres como este.

Párrafo perfectamente normal, que colocamos sólo para poder ver el efecto que tienen los cambios en los márgenes y espacios. Naturalmente, harán falta dos o tres como este.

```
\end{document}
```

Nota: Probablemente habrás notado que en `\parskip` hemos colocado un signo "=", pero en `\parindent` no. Ambas notaciones son equivalentes y no hay diferencias entre ellas.

Prueba a cambiar los valores y compilar un par de veces, para que veas el efecto que produce.