

# Sesión III

## $\text{\LaTeX}$ para Matemáticas

## ① Fórmulas

Distintos tipos de fórmulas

Símbolos

Matrices

Diagramas

## ② Teoremas

# Sección 1

## Fórmulas

# Escribiendo fórmulas

Hay diferentes entornos para escribir fórmulas:

## ① En línea

### Código

Puedo escribir `$e^{i\pi} + 1 = 0$`

Puedo escribir  $e^{i\pi} + 1 = 0$

# Escribiendo fórmulas

Hay diferentes entornos para escribir fórmulas:

- 1 En línea
- 2 **Presentada**  
(con `\[...\]` o  
`$$...$$` )

## Código

Puedo escribir

```
\[ e^{i\pi} + 1 = 0 \]
```

Puedo escribir

$$e^{i\pi} + 1 = 0$$

Añadir `*` al final de estos comandos (`equation*`, `align*`, ...) elimina la numeración.

# Escribiendo fórmulas

Hay diferentes entornos para escribir fórmulas:

- ① En línea
- ② Presentada  
(con `\[...\]` o `$$...$$`)
- ③ `equation`

## Código

Puedo escribir

```
\begin{equation}
    e^{i\pi} + 1 = 0
\end{equation}
```

Puedo escribir

$$e^{i\pi} + 1 = 0 \quad (1)$$

Añadir `*` al final de estos comandos (`equation*`, `align*`, ...) elimina la numeración.

# Escribiendo fórmulas

Hay diferentes entornos para escribir fórmulas:

- 1 En línea
- 2 Presentada  
(con `\[...\]` o `$$...$$`)
- 3 `equation`
- 4 `align`

## Código

Puedo escribir

```
\begin{align}
e^{i\pi} + 1 &= 0 \quad \backslash\backslash
e^{i\pi} &= -1
\end{align}
```

Puedo escribir

$$e^{i\pi} + 1 = 0 \quad (1)$$

$$e^{i\pi} = -1 \quad (2)$$

Añadir `*` al final de estos comandos (`equation*`, `align*`, ...) elimina la numeración.

# Símbolos útiles

$+$	<code>+</code>	$\varepsilon$	<code>\varepsilon</code>	$\frac{a}{b}$	<code>\frac{a}{b}</code>
$-$	<code>-</code>	$\delta$	<code>\delta</code>	$\sqrt{a}$	<code>\sqrt{a}</code>
$\times$	<code>\times</code>	$\partial$	<code>\partial</code>	$a^b$	<code>{a}^{{b}}</code>
$\div$	<code>\div</code>	$\Omega$	<code>\Omega</code>		
$\cdot$	<code>\cdot</code>	$\pi$	<code>\pi</code>		
$\oplus$	<code>\oplus</code>				
$\otimes$	<code>\otimes</code>				

La web Detexify permite buscar símbolos a partir de un dibujo a mano alzada.



# La ecuación más bella del mundo

La ecuación de Euler, popular por contener algunas de las más importantes constantes matemáticas puede escribirse

## Código

```
e^{i\pi} + 1 = 0
```

$$e^{i\pi} + 1 = 0$$

## Código

Puedo escribir la ecuación de Euler `$e^{i \pi} + 1 = 0$` en línea o presentada

```
\[  
    e^{i\pi} + 1 = 0
```

```
\]  
para que quede mejor
```

Puedo escribir la ecuación de Euler  $e^{i\pi} + 1 = 0$  en línea o presentada

$$e^{i\pi} + 1 = 0$$

para que quede mejor

# Subíndices y superíndices

Los subíndices de sumarios e integrales cambian de formato presentado a en línea

## Código

En línea digo

```
$\sum_{i=1}^n \int_a^b$
```

mientras que presentado

```
\[  
    \sum_{i=1}^n \int_a^b  
\]
```

En línea digo  $\sum_{i=1}^n \int_a^b$  mientras que presentado

$$\sum_{i=1}^n \int_a^b$$

# Subíndices y superíndices

Los subíndices de sumarios e integrales cambian de formato presentado a en línea

## Código

En línea digo

```
$\sum_{i=1}^n \int_a^b$
```

mientras que presentado

```
\[  
  \sum_{i=1}^n \int_a^b  
\]
```

En línea digo  $\sum_{i=1}^n \int_a^b$  mientras que presentado

$$\sum_{i=1}^n \int_a^b$$

Además la función `\substack` es muy útil

## Código

```
\[ \max_{\substack{y \in \Omega \\ |y| > 1}} \]
```

$$\max_{\substack{y \in \Omega \\ |y| > 1}}$$

# Matrices

Las matrices se introducen siempre en entornos matemáticos. Maple y matlab permiten exportar matrices a  $\text{\LaTeX}$ . Hay distintos tipos de matrices predeterminadas en el paquete `amsmath`.

- ① `matrix` Sin bordes
- ② `pmatrix` Entre `()`
- ③ `vmatrix` Entre `| |`
- ④ `bmatrix` Entre `[ ]`

# Matrices

## Ejemplo

### Código

```
\[  
\begin{pmatrix}  
1 & 2 & 3 \\  
4 & 5 & \\  
6 & & 7  
\end{pmatrix}  
\]
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & \\ 6 & & 7 \end{pmatrix}$$

# Matrices

## Ejemplo

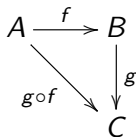
### Código

```
\[  
\begin{bmatrix}  
1 & 2 & 3 \\  
4 & 5 & \\  
6 & & 7  
\end{bmatrix}  
\]
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & \\ 6 & & 7 \end{bmatrix}$$

# El paquete xy-pic

Este paquete se emplea para hacer todo tipo de gráficos, por ejemplo el diagrama



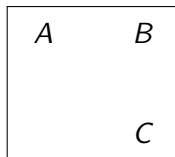
Tiene infinitud de opciones.

# xymatrix

Es la manera más sencilla de introducir diagramas. Los elementos que se conectarán por flechas se introducen en las posiciones de una matriz, de tipo `xymatrix`

## Código

```
\xymatrix{  
A & B \\  
& C  
}
```



Se puede introducir una `xymatrix` dentro o fuera de fórmulas, pero deberemos tener cuidado con el contenido.



# Las flechas

Dentro de una `xymatrix` podemos introducir flechas con el comando `\ar`  
Admite varios modificadores

- 1 **Destino** Colocando la flecha en la casilla de la que parte se coloca un cadena de cuantas casillas a derecha o izquierda y arriba o abajo está el destino.

`\ar[<hop>]`

u	arriba
d	abajo
r	derecha
l	izquierda
	a si misma

# Las flechas

Dentro de una `xymatrix` podemos introducir flechas con el comando `\ar`  
Admite varios modificadores

- ① Destino
- ② Etiqueta Se puede escribir sobre las letras

<code>\ar[r]^{f}</code>	$a \xrightarrow{f} b$
<code>\ar[r]_{f}</code>	$a \xrightarrow[f]{} b$
<code>\ar[r] {f}</code>	$a \xrightarrow{ f} b$

# Las flechas

Dentro de una `xymatrix` podemos introducir flechas con el comando `\ar`  
Admite varios modificadores

- ① Destino
  - ② Etiqueta
  - ③ **Tipo** Hay distintos tipos de base, cuerpos y cabezas de flecha
- `\ar@{<type>}[<hop>]`

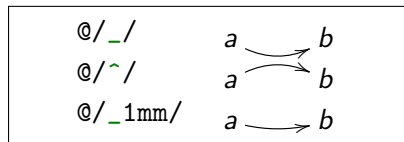
<code>@{=&gt;}</code>	$a \Longrightarrow b$
<code>@{.&gt;}</code>	$a \cdots \rightarrow b$
<code>@{:&gt;}</code>	$a \cdots \rightarrow b$
<code>@{~&gt;}</code>	$a \rightsquigarrow b$
<code>@{--&gt;}</code>	$a \dashrightarrow b$
<code>@{ -&gt;}</code>	$a \mapsto b$

# Las flechas

Dentro de una `xymatrix` podemos introducir flechas con el comando `\ar`  
Admite varios modificadores

- 1 Destino
- 2 Etiqueta
- 3 Tipo
- 4 **Curvatura** Podemos curvar las flechas hacia arriba y hacia abajo, para evitar que se corten, o solo para quede más estiloso

`\ar@/<curve>/` [`<hop>`]

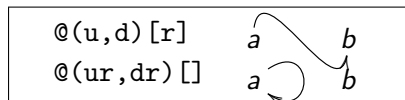


# Las flechas

Dentro de una `xymatrix` podemos introducir flechas con el comando `\ar`  
Admite varios modificadores

- 1 Destino
- 2 Etiqueta
- 3 Tipo
- 4 Curvatura
- 5 **Entrada y salida** Si queremos que la flecha salga desde una parte en concreto de la celda podemos especificarlo

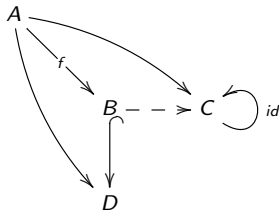
`\ar@(<in>,<out>)[<hop>]`



# Las flechas

## Ejercicio

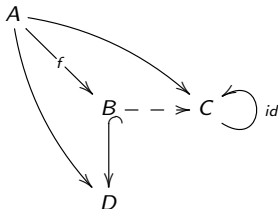
Escriba el siguiente diagrama:



# Las flechas

## Ejercicio

Escriba el siguiente diagrama:



## Código (ejercicio3.tex)

```
\[
\xymatrix{
  A \ar@/_2ex/[ddr] \ar[dr] |f \ar@/^2ex/[drr] \\
    & B \ar@{-->}[r] \ar@{^(->)}[d] & C \ar@(dr,ur)[]_{id} \\
    & D
}
\]
```

# El paquete xy-pic y el paquete babel

El paquete babel entra en conflicto con @ así que si queremos hacer buenos diagramas debemos desactivarlo. Empleando inputenc con utf8 no tendremos problemas con los acentos. Debemos cambiar los nombres de capítulos y secciones. Para ello

```
\renewcommand{<command>}{<new_name>}
```

<code>\abstractname</code>	Abstract
<code>\appendixname</code>	Appendix
<code>\bibname</code>	Bibliography (report,book)
<code>\chaptername</code>	Chapter (report,book)
<code>\contentsname</code>	Contents
<code>\figurename</code>	Figure (for captions)
<code>\indexname</code>	Index
<code>\listfigurename</code>	List of Figures
<code>\listtablename</code>	List of Tables
<code>\tablename</code>	Table (for caption)



# Teoremas

El paquete `amsthm`

A la hora definir un teorema debemos tener en cuenta tres cosas

- 1 **El estilo:** Los teoremas se escriben en cursiva, mientras que las definiciones se escriben con fuente normal.

## Código

```
\documentclass
(...)
\theoremstyle{<style>}
\newtheorem{<env>}{<name>}
(...)
\begin{document}
```

Hay tres estilos predefinidos:

<code>plain</code>	<b>Theorem 1.</b> <i>Theorem text.</i>
<code>definition</code>	<b>Definition 1.</b> Definition text.
<code>remark</code>	<i>Remark 1.</i> Remark text.

# Teoremas

El paquete `amsthm`

A la hora definir un teorema debemos tener en cuenta tres cosas

- 1 El estilo
- 2 **El nombre:** Debemos poner un nombre de entorno `<env>`, ya sea `teorema` (por ejemplo `<env>=teorema`) un nombre para mostrar en el documento (por ejemplo `<name>=Teorema`)

## Código

```
\documentclass
(...)
\theoremstyle{<style>}
\newtheorem{<env>}{<name>}
(...)
\begin{document}
```

# Teoremas

El paquete `amsthm`

A la hora definir un teorema debemos tener en cuenta tres cosas

- 1 El estilo
- 2 El nombre
- 3 **La numeración**: Podemos numerar los teoremas de diferentes maneras

## Código

```
\documentclass  
(...)  
\theoremstyle{<style>}  
\newtheorem{<env>}{<name>}  
(...)  
\begin{document}
```

# Teoremas

El paquete `amsthm`

A la hora definir un teorema debemos tener en cuenta tres cosas

- 1 El estilo
- 2 El nombre
- 3 La numeración
  - a) Con su propio contador: El contador se crea por defecto si no decimos nada más, y se nombra automáticamente como `<env>`

## Código

```
\documentclass  
(...)  
\theoremstyle{<style>}  
\newtheorem{<env>}{<name>}  
(...)  
\begin{document}
```

# Teoremas

El paquete `amsthm`

A la hora definir un teorema debemos tener en cuenta tres cosas

- ① El estilo
- ② El nombre
- ③ La numeración
  - a) Con su propio contador:
  - b) Siguiendo la numeración de otro teorema ya definido

## Código

```
\documentclass
(...)
\theoremstyle{<style>}
\newtheorem{<env>}
[<counter>]{<name>}
(...)
\begin{document}
```

# Teoremas

El paquete `amsthm`

A la hora definir un teorema debemos tener en cuenta tres cosas

- ① El estilo
- ② El nombre
- ③ La numeración
  - a) Con su propio contador:
  - b) Siguiendo la numeración de otro teorema ya definido
  - c) Supeditada a otro contador, por ejemplo la sección. En este caso el contador de tipo a) lleva como predecesor el otro contador, y se resetea al cambiar el contador al que supedita

## Código

```
\documentclass
(...)
\theoremstyle{<style>}
\newtheorem{<env>}{<name>}[<counte
(...)
\begin{document}
```

# Teoremas

## Ejemplo

Este es el aspecto de un teorema normal definido con el paquete `amsthm`.

### Código (`thm1.tex`)

```
\documentclass{article}
\usepackage{amsthm}

\theoremstyle{plain}
\newtheorem{teorema}{Teorema}

\begin{document}

\begin{teorema}[Euclides]
No existe un primo mayor que el resto.
\end{teorema}

\end{document}
```

**Teorema 1** (Euclides) . *No existe un primo mayor que el resto.*

# Operadores matemáticos

Con el paquete `amsmath` se pueden definir operadores matemáticos, como `div` o `rot`:

```
\DeclareMathOperator{\rot}{rot}
```

Si queremos que el operador tenga sub-índice debajo entonces

```
\DeclareMathOperator*{\argmin}{arg\,min}
```



# Ejercicio

Escribir el siguiente documento  $\text{\LaTeX}$

## El teorema de la divergencia

Un estudiante

10 de abril de 2013

### 1. El teorema

El teorema de la divergencia de Gauss se enuncia de la siguiente manera

**Teorema 1.1.** *Dado ....*

*Demostración.* La prueba...

□

### 2. Ejercicios

**Ejercicio 1.** Este ejercicio

# Solución

## Código (ejercicio1.tex)

```
\documentclass{article}

\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}
\usepackage{amsthm}

\theoremstyle{plain}
\newtheorem{thm}{Teorema}[section]
\theoremstyle{definition}
\newtheorem{ex}{Ejercicio}

\title{El teorema de la divergencia}
\author{Un estudiante}
\date{10 de abril de 2013}

\begin{document}
\maketitle
```

## Código

```
\section{El teorema}

El teorema de la divergencia de Gauss
se enuncia de la siguiente manera

\begin{thm}
  Dado ....
\end{thm}
\begin{proof}
  La prueba...
\end{proof}

\section{Ejercicios}

\begin{ex}
  Este ejercicio
\end{ex}

\end{document}
```