

$\text{\LaTeX}$  2<sub>ε</sub>: Història, funcionament, codificació de caràcters, idioma, notes a peu de plana, espais, salts, accents, . . .

Els entorns `itemize` i `enumerate`. El paquet `enumerate`. Taules i fórmules.

Lluís Alsedà, Albert Ruiz

Departament de Matemàtiques



Universitat Autònoma  
de Barcelona

# Continguts

- 1 Introducció del curs
- 2 Història
- 3 Funcionament del  $\text{\LaTeX} 2_{\epsilon}$
- 4 Idioma i codificació de caràcters
- 5 Espais, salts de línia i salts de plana
- 6 Notes al peu de plana
- 7 Accents, cometes, guions i títols
- 8 Els entorns `itemize` i `enumerate`
- 9 El paquet `enumerate`
- 10 Taules
- 11 Fórmules
- 12 Projectes grans

# El curs

## Què suposem

Aquest és un curs d'introducció d'ús del processador de textos  $\text{\LaTeX} 2_{\epsilon}$ .

Suposarem que disposem d'un ordinador amb el  $\text{\LaTeX} 2_{\epsilon}$  instal·lat i amb l'entorn necessari per a utilitzar-lo:

- Un editor (per exemple el `kile`, el `winedt` o el `winshell`).
- Una distribució de  $\text{\LaTeX} 2_{\epsilon}$  (per exemple el `texlive` o el `Miktex`).
- Un visualitzador de `dvi` (per exemple l'`xdvi` o el `yap`).
- Un visualitzador de `pdf` (per exemple l'`xpdf` o el `acroread`).
- Un visualitzador de `ps` (per exemple el `ghostview`).

# El curs

## Què suposem

Aquest és un curs d'introducció d'ús del processador de texts  $\text{\LaTeX} 2_{\epsilon}$ .

Suposarem que disposem d'un ordinador amb el  $\text{\LaTeX} 2_{\epsilon}$  instal·lat i amb l'entorn necessari per a utilitzar-lo:

- Un editor (per exemple el `kile`, el `winedt` o el `winshell`).
- Una distribució de  $\text{\LaTeX} 2_{\epsilon}$  (per exemple el `texlive` o el `Miktex`).
- Un visualitzador de `dvi` (per exemple l'`xdvi` o el `yap`).
- Un visualitzador de `pdf` (per exemple l'`xpdf` o el `acroread`).
- Un visualitzador de `ps` (per exemple el `ghostview`).

## Què no veurem

A aquest curs no veurem com instal·lar el  $\text{\LaTeX} 2_{\epsilon}$  en un ordinador.

# Bibliografia

Si es fa tota la instal·lació del  $\text{\LaTeX} 2_{\epsilon}$  hi ha un document que anomenat `lshort...` que és un bon manual. A més, està en varis idiomes.

# Bibliografia

Si es fa tota la instal·lació del  $\text{\LaTeX} 2_{\epsilon}$  hi ha un document que anomenat `lshort...` que és un bon manual. A més, està en varis idiomes.

## Bibliografia i enllaços

- 1 Donald E. Knuth. *The  $\text{\TeX}$ book*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachussets, second edition, 1984.
- 2 Leslie Lamport,  *$\text{\LaTeX}$ : A Document Preparation System*. Addison-Wesley, Reading, Massachussets, second edition, 1994.
- 3 Tobias Oetiker, Hubert Partl, Irene Hyna and Elisabeth Schlegl. *The Not So Short Introduction to  $\text{\LaTeX} 2_{\epsilon}$ . Or  $\text{\LaTeX} 2_{\epsilon}$  in 157 minutes*.  
<http://tobi.oetiker.ch/lshort/lshort.pdf>

# Història

El  $\text{T}_{\text{E}}\text{X}$  és un programa creat per en Donald E. Knuth (computer scientist) i la primera versió és del 1977.

La versió de  $\text{T}_{\text{E}}\text{X}$  que utilitzem actualment està basada en una del 1982, amb millores. Per exemple, al 1989 es van incorporar les compatibilitats amb les diferents codificacions de caràcters.

Les versions de  $\text{T}_{\text{E}}\text{X}$  s'enumeren afegint un dígit a la versió anterior i tendint a  $\pi$ . Actualment és la 3.1415926.

# Història

El  $\text{T}_{\text{E}}\text{X}$  és un programa creat per en Donald E. Knuth (computer scientist) i la primera versió és del 1977.

La versió de  $\text{T}_{\text{E}}\text{X}$  que utilitzem actualment està basada en una del 1982, amb millores. Per exemple, al 1989 es van incorporar les compatibilitats amb les diferents codificacions de caràcters.

Les versions de  $\text{T}_{\text{E}}\text{X}$  s'enumeren afegint un dígit a la versió anterior i tendint a  $\pi$ . Actualment és la 3.1415926.

El  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  és un macro-paquet que utilitza el  $\text{T}_{\text{E}}\text{X}$  com a motor. Va ser creat per Leslie Lamport i té moltes funcions predefinides que faciliten el processament de textos.

El  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  és una evolució del  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ .

A aquest curs farem servir el  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ .



# Funcionament

## Com funciona?

El  $\text{T}_{\text{E}}\text{X}$  no és un processador de texts WYSIWYG (*what you see is what you get*) sinó un processador WYTIWYG (*what you type is what you get*) i això té avantatges (i algun inconvenient). Això vol dir que mentre escrivim el text no veiem com quedarà, i és el mateix  $\text{T}_{\text{E}}\text{X}$  qui, a partir del text que hem entrat i de les instruccions que li donem, decideix com processar el document i com quedarà al final.

# Avantatges

## Avantatges

- Es poden incloure fórmules matemàtiques de forma senzilla.
- Tant sols cal aprendre instruccions per a definir l'estructura del document. Quasi mai s'ha de decidir sobre el format de la pàgina.
- Estructura automàtica de l'índex, les notes a peu de plana, la bibliografia, les figures, les taules, ...
- Paquets addicionals per a tasques específiques: presentacions, altres alfabetes, partitures, ...
- El  $\text{\LaTeX} 2_{\epsilon}$  està disponible per a molts sistemes operatius.

# Desavantatges

## Desavantatges

- Cal estructurar el document: és molt difícil escriure documents de forma desestructurada i desorganitzada.
- Modificar algun paràmetre de la pàgina és fàcil, però definir tota una nova estructura és difícil.
- Una instal·lació completa en alguns sistemes operatius pot ser complicada.

# Fitxers

## Fitxers

Nosaltres escriurem els documents en un fitxer de text pla. Això vol dir que aquest fitxer conté, a més del text que volem generar, les instruccions necessàries per a processar-lo. En general aquest fitxer té l'extensió `.tex`.

# Fitxers

## Fitxers

Nosaltres escriurem els documents en un fitxer de text pla. Això vol dir que aquest fitxer conté, a més del text que volem generar, les instruccions necessàries per a processar-lo. En general aquest fitxer té l'extensió `.tex`.

Quan processem aquest fitxer amb el  $\text{\LaTeX}$  obtindrem un fitxer `dvi` (si tot va bé).

# Fitxers

## Fitxers

Nosaltres escriurem els documents en un fitxer de text pla. Això vol dir que aquest fitxer conté, a més del text que volem generar, les instruccions necessàries per a processar-lo. En general aquest fitxer té l'extensió `.tex`.

Quan processem aquest fitxer amb el  $\text{\LaTeX}$  obtindrem un fitxer `dvi` (si tot va bé).

Quan processem aquest fitxer amb el `pdflatex` obtindrem un fitxer `pdf` (si tot va bé).

# Fitxers

## Fitxers

Nosaltres escriurem els documents en un fitxer de text pla. Això vol dir que aquest fitxer conté, a més del text que volem generar, les instruccions necessàries per a processar-lo. En general aquest fitxer té l'extensió `.tex`.

Quan processem aquest fitxer amb el  $\text{\LaTeX}$  obtindrem un fitxer `dvi` (si tot va bé).

Quan processem aquest fitxer amb el `pdflatex` obtindrem un fitxer `pdf` (si tot va bé).

Cada cop que processem fitxers es creen fitxers auxiliars que s'utilitzen en la compilació següent (per exemple per generar l'índex). Per tant un fitxer s'ha de processar diverses vegades.

# Fitxers

## Fitxers

Nosaltres escriurem els documents en un fitxer de text `pla`. Això vol dir que aquest fitxer conté, a més del text que volem generar, les instruccions necessàries per a processar-lo. En general aquest fitxer té l'extensió `.tex`.

Quan processem aquest fitxer amb el  $\text{\LaTeX}$  obtindrem un fitxer `dvi` (si tot va bé).

Quan processem aquest fitxer amb el `pdflatex` obtindrem un fitxer `pdf` (si tot va bé).

Cada cop que processem fitxers es creen fitxers auxiliars que s'utilitzen en la compilació següent (per exemple per generar l'índex). Per tant un fitxer s'ha de processar diverses vegades.

En general, podem passar de `dvi` a `pdf` o `ps` amb programes externs.



# Primer exemple

## Exemple sessio1a.tex

```
\documentclass[a4paper,11pt]{article}
% define the title
\title{Minimalism}
\author{H.~Author}
\begin{document}
% insert the title
\maketitle
% insert a table of contents
\tableofcontents
\section{Some interesting Words}
Well, and here begins my article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}
```

# Observacions

## Capçalera i cos

Els fitxers tenen dues parts ben diferenciades:

La capçalera comença amb una línia on es declara quin tipus de document és. La instrucció és `\documentclass` i els documents més habituals són `article`, `book`, `letter`, `report` i `beamer`.

A continuació es carreguen paquets, es declaren instruccions i variables. La capçalera acaba a la línia `\begin{document}`.

# Observacions

## Capçalera i cos

Els fitxers tenen dues parts ben diferenciades:

La capçalera comença amb una línia on es declara quin tipus de document és. La instrucció és `\documentclass` i els documents més habituals són `article`, `book`, `letter`, `report` i `beamer`.

A continuació es carreguen paquets, es declaren instruccions i variables. La capçalera acaba a la línia `\begin{document}`.

El cos del document: és el que hi ha entre el `\begin{document}` i el `\end{document}`. Aquí és on escrivim la gran part del text que volem processar.

# Observacions (cont)

## Caràcters especials

Les instruccions en  $\text{\LaTeX}$  comencen amb el caràcter  $\backslash$ . Per tant aquest és un caràcter especial que no s'imprimeix. Els altres caràcters especials són:  $\%$ ,  $\#$ ,  $\$$ ,  $\^$ ,  $\_$ ,  $\{$ ,  $\}$  i  $\sim$ .

Si ens interessa imprimir s'ha d'escriure un  $\backslash$  al davant, excepte el  $\backslash$ . Per exemple  $\%$  imprimeix un  $\%$ .

El  $\backslash$  es pot imprimir amb la comanda  $\$\\backslash\$$ .

# Observacions (cont)

## Caràcters especials

Les instruccions en  $\text{\LaTeX}$  comencen amb el caràcter  $\backslash$ . Per tant aquest és un caràcter especial que no s'imprimeix. Els altres caràcters especials són:  $\%$ ,  $\#$ ,  $\$$ ,  $\^$ ,  $\_$ ,  $\{$ ,  $\}$  i  $\sim$ .

Si ens interessa imprimir s'ha d'escriure un  $\backslash$  al davant, excepte el  $\backslash$ . Per exemple  $\%$  imprimeix un  $\%$ .

El  $\backslash$  es pot imprimir amb la comanda  $\$\\backslashslash\$$ .

## Instruccions

Algunes instruccions necessiten paràmetres obligatoris, que s'entren entre claus ( $\{\dots\}$ ).

En canvi, els paràmetres opcionals s'entren entre claudàtors ( $[\dots]$ ).

# Algunes observacions més

## Parts, capítols, seccions, subseccions ...

Un document es compon de parts (el tipus llibre), cada part per capítols (llibre o report), llavors seccions, subseccions, subsubseccions, paràgrafs i subparàgrafs. Les comandes són:

```
\part
\chapter
\section
\subsection
\subsubsection
\paragraph
\subparagraph
```

Totes tenen un argument obligatori (el títol) i un d'opcional (el títol curt). Per exemple

```
\chapter[Títol curt]{Títol massa llarg per la capçalera}
```

# Un altre exemple

## Exemple sessio1b.tex

```

\documentclass[a4paper,12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage[catalan]{babel}
\title{Segon document de la sessió 1}
\author{Curs de \LaTeXe}
\begin{document}
\maketitle
\section{Primera secció}
Per a fer una fórmula hem d'utilitzar el \$,
i no es veu igual

$$\sum_{i=1}^{\infty} i$$

que

$$\sum_{i=1}^{\infty} i$$

\end{document}

```

# Observacions de l'exemple sessio1b.tex

## El mode matemàtic

Veiem que el  $\LaTeX$  tracta de manera diferent el text de les fórmules.

Utilitzem la comanda  $\$$  per a obrir i tancar una fórmula que ha d'aparèixer enmig d'una línia de text.

Utilitzem la comanda  $\$\$$  per a obrir i tancar una fórmula que ha de situar-se entre dos paràgrafs.



# Observacions de l'exemple sessio1b.tex

## El mode matemàtic

Veiem que el  $\text{\LaTeX}$  tracta de manera diferent el text de les fórmules.

Utilitzem la comanda  $\$$  per a obrir i tancar una fórmula que ha d'aparèixer enmig d'una línia de text.

Utilitzem la comanda  $\$\$$  per a obrir i tancar una fórmula que ha de situar-se entre dos paràgrafs.

## Idioma del document

Per a treballar còmodament en idiomes amb caràcters especials (accents, ç, ...) cal configurar correctament l'editor de text i carregar dos paquets a la capçalera del document: `inputenc` i `babel`.

# Idioma i codificació de caràcters

## L'idioma

Si no fem res, el  $\text{\LaTeX}2_{\epsilon}$  treballa en anglès i no admet caràcters amb accents, ni ç, ...

Si volem escriure un document en català carregarem dos paquets:

- `inputenc`: Encara que no ho vegem, el document `.tex` està guardat en una codificació concreta que afecta principalment als caràcters amb accents, ç, ... Normalment, també s'ha d'especificar a la configuració de l'editor de text que utilitzem. El paquet `inputenc`, amb el paràmetre opcional corresponent, permet dir-li al  $\text{\LaTeX}2_{\epsilon}$  quina és la codificació que utilitza. Si volem treballar en català és recomanable utilitzar les codificacions `utf8` o bé `latin1`.
- `babel`: Fa que la compilació del document sigui la particular de l'opció d'idioma que hi entrem (`catalan`): espaiat entre paraules i com es trenquen les síl·labes al final de línia.

# Espais i salts (I)

## Paràgrafs

En principi el  $\text{\LaTeX} 2_{\epsilon}$  “passa” dels espais que deixem entre les paraules i només té en compte si n’hi ha o no i on són.

Per exemple, si creu que és un punt de final de frase, deixa una mica més d’espai que entre dues paraules qualssevol. Podem evitar que faci espais més grans entre paraules concretes amb el símbol  $\sim$  (per exemple, Dep. $\sim$ Matemàtiques).

A més, interpreta un salt de línia com una separació entre paraules (sempre que no acabi amb el caràcter  $\%$ ).

D’aquesta manera obtenim un paràgraf.

Per a forçar un canvi de paràgraf cal fer dos (o més) salts de línia seguits, deixant una línia entremig en blanc.

# Espais i salts (II)

## Longitud de les línies

El processador ajusta la longitud de les línies a les mides que té predefinides. Això també inclou que, si ho creu necessari, talla les paraules segons l'idioma que té definit.

Tot i això a vegades no talla bé una paraula, o bé no la talla i queda una línia que surt del marge predefinit. Per a arreglar això podem dir-li com es talla la paraula amb la instrucció `\-`. Per exemple `e\ -xem\ -ple`.

Si volem forçar que una paraula no la talli mai, podem utilitzar la comanda `\mbox{paraula}`.

També podem forçar salts de línia amb la comanda `\newline` o bé la comanda `\\` (però això no ens canvia de paràgraf).

# Espais i salts (III)

## Salts de plana

La comanda `\newpage` o bé la comanda `\pagebreak`.

La diferència és que `\pagebreak` allarga el cos de la pàgina fins el marge inferior (aconsellat per a evitar línees orfes).

## Espais i salts (III)

### Salts de plana

La comanda `\newpage` o bé la comanda `\pagebreak`.

La diferència és que `\pagebreak` allarga el cos de la pàgina fins el marge inferior (aconsellat per a evitar línies orfes).

### Mode matemàtic

A l'exemple següent veiem com varia l'espai entre símbols depenent del que siguin.

\$\$

`\int (f-g)\mathrm{d} x-\int h \mathrm{d}x \, , .`

\$\$

$$\int (f - g)dx - \int hdx .$$

# Notes al peu de plana

## Notes al peu de plana (sessio1c.tex)

Per a posar un peu de plana utilitzem la comanda `\footnote`. El text que hi posem ha d'anar entre claus. Numera els peus de plana de manera automàtica.

# Accents i cometes

## Accents

Si cal es poden escriure els accents i els caràcters particulars de cada idioma com a comandes en T<sub>E</sub>X (és independent de la codificació de caràcters). Per exemple:

```
Cont\'\'i nua, continu\"\'i tat,
\'area, \~n, \c c, \o, \ldots
```

Produeix: Contínua, continuïtat, àrea, ñ, ç, ø, ...

Veiem que hem d'escriure `\i` per a obtenir una *i* sense el punt (una *ı*) i posar-li l'accent a sobre.



# Accents i cometes

## Accents

Si cal es poden escriure els accents i els caràcters particulars de cada idioma com a comandes en  $\text{T}_{\text{E}}\text{X}$  (és independent de la codificació de caràcters). Per exemple:

```
Cont\'\'i nua, continu\"\"i tat,
\'area, \~n, \c c, \o, \ldots
```

Produeix: Contínua, continuïtat, àrea, ñ, ç, ø, ...

Veiem que hem d'escriure `\i` per a obtenir una *i* sense el punt (una *ı*) i posar-li l'accent a sobre.

## Cometes

Pel que fa a les cometes, per obrir les cometes hem d'escriure dos accents oberts seguits (‘‘) i per tancar-la dos apòstrofs seguits (’’). El resultat és “acceptable”.

# Guions

## Guions

El  $\text{\LaTeX} 2_{\epsilon}$  distingeix la longitud de guions depenent de:

- si estem en mode text i volem trencar una paraula (sumar-se) posem un sol guió `-`,
- si estem en mode text i volem delimitar un rang de planes (5–10) podem dos guions `--`,
- encara es pot fer més llarg amb 3 guions `---` i queda (sí—no)  
o
- en mode matemàtic per al signe o la operació (un sol guió):  
 $a - b$ .

# Titles

## Titles

Actualment s'utilitza la titlla a l'adreça de moltes planes web.  
Podem veure la diferència entre les dues opcions que tenim:

`http://mat.uab.cat/~{}seccio`

produeix

`http://mat.uab.cat/~seccio`

mentre que

`http://mat.uab.cat/$\sim$seccio`

produeix

`http://mat.uab.cat/~seccio`

# L'entorn itemize

## itemize

Els `itemize` s'utilitzen per a fer llistes. Comencen amb la comanda `\begin{itemize}`, cada cop que volem afegir un element utilitzem la comanda `\item` i acabem amb `\end{itemize}`.

# L'entorn itemize

## itemize

Els `itemize` s'utilitzen per a fer llistes. Comencen amb la comanda `\begin{itemize}`, cada cop que volem afegir un element utilitzem la comanda `\item` i acabem amb `\end{itemize}`.

## Exemple (sessio1d.tex)

```
\begin{itemize}
  \item Aquest és el primer element de la llista.
  \item Aquest és el segon i en fem un altre:
  \begin{itemize}
    \item Ara hi ha un \texttt{itemize} dins d'un altre.
  \end{itemize}
\end{itemize}
```

# L'entorn enumerate

## enumerate

Si volem que cada element de la llista estigui enumerat utilitzem l'`enumerate`. La sintaxi és la mateixa que la de l'`itemize`.

# L'entorn enumerate

## enumerate

Si volem que cada element de la llista estigui enumerat utilitzem l'enumerate. La sintaxi és la mateixa que la de l'itemize.

## Exemple (sessio1d.tex)

```
\begin{enumerate}
  \item Aquest és el primer element de la llista.
  \item Aquest és el segon i en fem un altre:
  \begin{enumerate}
    \item Ara hi ha un enumerate dins d'un altre.
  \end{enumerate}
\end{enumerate}
```

# El paquet enumerate

El paquet enumerate permet modificar com surten les enumeracions amb una sintaxi senzilla:

## Exemple (sessiold.tex)

```
\begin{enumerate}[1.-]
  \item Ara fa l'enumerate amb 1.-\ .
  \item El segon punt.
\end{enumerate}
\begin{enumerate}[A)]
  \item Ara fa l'enumerate amb A)\ .
  \item El segon punt.
\end{enumerate}
\begin{enumerate}[\bf 1.-]
  \item Ara fa l'enumerate amb \textbf{1.-}\ .
  \item El segon punt.
\end{enumerate}
```



## El paquet enumerate (II)

## Exemple (cont. sessio1d.tex)

```

\begin{enumerate}[(i)]
  \item Ara fa l'enumerate amb números romans entre
        parèntesis.
  \item El segon punt.
\end{enumerate}
\begin{enumerate}[(I)]
  \item Ara fa l'enumerate amb números romans.
  \item El segon punt.
\end{enumerate}
\begin{enumerate}[{\Apartat} a]
  \item Ara fa l'enumerate amb Apartat a-\ .
  \item El segon punt.
\end{enumerate}

```

# Taules

Per a fer taules s'utilitza el la instrucció tabular:

## Exemple (sessio1d.tex)

```
\begin{tabular}{|l|c|r|}
\hline Primera col. & Segona col. & Tercera col. \\
\hline Left & Center & Right \\
\hline
\end{tabular}
```

I obtenim:

Primera col.	Segona col.	Tercera col.
Left	Center	Right

## Taules (II)

Si volem que una columna tingui una mida forçada es pot fer amb:

### Exemple (sessio1d.tex)

```
\begin{tabular}{|p{25mm}|c|}
```

```
\hline
```

Aquesta columna està forçada a 25 mm, escrigui  
el que escrigui.&

Aquesta agafa l'amplada del que escrigui.\\

```
\hline
```

```
\end{tabular}
```

<p>Aquesta columna està forçada a 25 mm, escrigui el que escrigui.</p>	<p>Aquesta agafa l'amplada del que escrigui.</p>
--	--

# Taules (III)

Si la nostra taula té agrupacions de columnes i línies es poden utilitzar les comandes `\multicolumn` i `\multirow` (aquest últim necessita carregar el paquet `multirow`).

## Exemple (sessio1d.tex)

```
\begin{tabular}{|c||c|c|c|c|c|c|c|}
\hline
 $W$  &  $E_0$  &  $E_1$  &  $E_{-1}$  &  $V_0$  &  $V_1$  &  $V_{-1}$  & Exotic? \\
\hline \hline
& &  $\bigstar$  & & & {Yes} \\
&  $\angle$  &  $\omega$  &  $\angle$  & & &  $\bigstar$  &  $\bigstar$  & \\
{Yes} & & & & & & & & \\
& & &  $\bigstar$  &  $\bigstar$  &  $\bigstar$  & & No \\
\\ \hline  $\angle$  &  $\eta$  &  $\angle$  & &  $\bigstar$  & & \\
\multicolumn{2}{|c|}{ } & & \multicolumn{2}{|c|}{ } & No \\
\multirow{5}{*}{ $\angle$  &  $\omega$  &  $\angle$  & & \multicolumn{2}{|c|}{ } & & & \\
\multicolumn{2}{|c|}{ } &  $\bigstar$  & & {Yes} & & & & \\
& & \multicolumn{2}{|c|}{ } & & \multirow{2}{*}{ $\bigstar$ } & & & \\
\multicolumn{2}{|c|}{ } & & {Yes} & & & & & \\
& & \multicolumn{2}{|c|}{ } & & & & & \\
\multicolumn{2}{|c|}{ } &  $\bigstar$  & & No & & & & \\
& \multirow{2}{*}{ $\bigstar$ } &  $\bigstar$  & & \multicolumn{2}{|c|}{ } & & & \\
\multicolumn{2}{|c|}{ } & No & & & & & & \\
& & \multicolumn{2}{|c|}{ } & & & \multicolumn{2}{|c|}{ $\bigstar$ } & \\
No \\
\hline
\end{tabular}
```

# Taules (III')

## Exemple (sessio1d.tex)

$W$	$E_0$	$E_1$	$E_{-1}$	$V_0$	$V_1$	$V_{-1}$	Exotic?
$\langle \omega \rangle$				★			Yes
					★	★	Yes
				★	★	★	No
$\langle \eta \rangle$	★						No
$\langle \eta, \omega \rangle$					★		Yes
				★			Yes
					★		No
							No
	★				★		No

## Taules (IV)

Si el problema de la taula és que ocupa més d'una pàgina, i necessitem que el  $\text{T}\text{E}\text{X}$  decideixi on fer els salts de plana podem utilitzar el paquet `longtable` i hem de tenir en compte (veure `sessio1d.tex`):

- L'entorn es diu `longtable`.
- S'ha de definir la primera capçalera i la que apareix a cada plana.
- El mateix pels peus d'entremig i per l'últim peu.

# Fórmules dins i fora d'un paràgraf

Per a escriure fórmules matemàtiques dins el mateix paràgraf l'entrem entre `$` i `$`, o bé entre `\(` i `\)`, o bé entre `\begin{math}` i `\end{math}`.

# Fórmules dins i fora d'un paràgraf

Per a escriure fórmules matemàtiques dins el mateix paràgraf l'entrem entre `$ i $`, o bé entre `\( i \)`, o bé entre `\begin{math}` i `\end{math}`.

Per exemple `$a^2+b^2$` produeix  $a^2 + b^2$ . O bé `$$\sum_{i=1}^n i^2$` produeix  $\sum_{i=1}^n i^2$ .



# Fórmules dins i fora d'un paràgraf

Per a escriure fórmules matemàtiques dins el mateix paràgraf l'entrem entre `$ i $`, o bé entre `\( i \)`, o bé entre `\begin{math} i \end{math}`.

Per exemple `$a^2+b^2$` produeix  $a^2 + b^2$ . O bé `$$\sum_{i=1}^n i^2$` produeix  $\sum_{i=1}^n i^2$ .

Si volem escriure fórmules que per si mateixes formin un paràgraf les entrem entre `$$ i $$`, o bé entre `\[ i \]`, o bé entre `\begin{displaymath} i \end{displaymath}`.

Per exemple

$$a^2 + b^2$$

$$\sum_{i=1}^n i^2$$

## Fórmules dins i fora d'un paràgraf (II)

Si volem que els sumatoris i límits surtin a les fórmules dins el paràgraf com surten a les que formen un paràgraf independent

podem utilitzar la comanda `\displaystyle`, per exemple  $\sum_{i=1}^n i^2$ .

## Fórmules dins i fora d'un paràgraf (II)

Si volem que els sumatoris i límits surtin a les fórmules dins el paràgraf com surten a les que formen un paràgraf independent podem utilitzar la comanda `\displaystyle`, per exemple  $\sum_{i=1}^n i^2$ .

Els espais a les fórmules no apareixen al compilar. Si volem que apareguin espais utilitzem les comandes `\,`, `\quad` o bé `\qquad`. Si el que volem és un espai negatiu utilitzem `\!`. L'espai negatiu s'utilitza, per exemple, quan fem una integral doble posem dos espais negatius (`\!\!`) entre les dues integrals:

$$\iint\!\!_D dx dy \text{ en lloc de } \int \int_D dx dy$$

# Fórmules: observacions

- Si volem incloure un text dins d'una fórmula podem utilitzar la comanda `\mbox{text}`.
- Les funcions que tenen nom (per exemple sinus, cosinus, ...) s'entren mitjançant una comanda (per exemple `\sin`, `\cos`, ...).

# Fòrmules: observacions

- Si volem incloure un text dins d'una fórmula podem utilitzar la comanda `\mbox{text}`.
- Les funcions que tenen nom (per exemple sinus, cosinus, ...) s'entren mitjançant una comanda (per exemple `\sin`, `\cos`, ...).

## Exemple (sessio1e.tex)

```
$$
\sin(x) \mbox{ no és el mateix que } \sin(x) \text{ ,.}
$$
```

$\sin(x)$  no és el mateix que  $\sin(x)$ .

# Delimitadors

Un delimitador s'adapta a la mida vertical de la fórmula a que conté. Utilitzem les comandes `\left` i `\right` per a posar-los.

# Delimitadors

Un delimitador s'adapta a la mida vertical de la fórmula a que conté. Utilitzem les comandes `\left` i `\right` per a posar-los.

## Exemple (sessio1e.tex)

\$\$

```
A_m=\left\{ (x_1, \dots, x_n) \mid \sum_{i=1}^n x_i=m \right\}
```

\$\$

$$A_m = \left\{ (x_1, \dots, x_n) \mid \sum_{i=1}^n x_i = m \right\}$$

## Delimitadors (II)

Si volem evitar el delimitador d'un costat utilitzem el . (punt).



## Delimitadors (II)

Si volem evitar el delimitador d'un costat utilitzem el . (punt).

### Exemple (sessio1e.tex)

```


$$|x| = \left\{ \begin{array}{ll} -x & \text{si } x < 0, \\ x & \text{si } x \geq 0. \end{array} \right.$$


```

$$|x| = \begin{cases} -x & \text{si } x < 0, \\ x & \text{si } x \geq 0. \end{cases}$$



# Delimitadors (III)

Els possibles delimitadors són (alguns necessiten l'amssymb):

```
\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|}
\hline [ & ] & { & } & | & \| & \langle & \rangle & ( & ) \\
\hline \downarrow & \Downarrow & \uparrow & \Uparrow & \updownarrow & \Updownarrow & \lceil & \rceil & \lfloor & \rfloor \\
\hline / & \backslash & \curvearrowleft & \curvearrowright & \ulcorner & \urcorner & \llcorner & \lrcorner & \bracevert & \cline{1-7} \\
\hline \end{array}
```

[	]	{	}			<	>	(	)
↓	⇓	↑	⇑	↕	⇕	⌈	⌋	⌊	⌋
/	\	↵	↶	⌞	⌟				
		⌈	⌋	⌊	⌋	,			

## Delimitadors (IV)

A vegades la mida que calcula el  $\text{T}_{\text{E}}\text{X}$  per al delimitador no ens agrada, o bé el delimitador ha d'agafar més d'una línia.

En aquests casos les instruccions `\left` i `\right` no ens serveixen, i ens pot interessar canviar la mida d'alguns d'aquests delimitadors. Per això utilitzem comandes com la `\big`.

## Delimitadors (IV)

A vegades la mida que calcula el T<sub>E</sub>X per al delimitador no ens agrada, o bé el delimitador ha d'agafar més d'una línia.

En aquests casos les instruccions `\left` i `\right` no ens serveixen, i ens pot interessar canviar la mida d'alguns d'aquests delimitadors. Per això utilitzem comandes com la `\big`.

```


$$\Big((x+1)(x-1)\Big)^2$$


$$\big(\Big(\big(\Big(\quad$$


$$\big)\Big)\big)\Big)$$


```

$$((x+1)(x-1))^2$$

$$(((\left( \right)\right)\right)$$

# Matrius

L'entorn `array` té una estructura molt semblant al `tabular`, però l'hem de posar dins de mode matemàtic.

# Matrius

L'entorn array té una estructura molt semblant al tabular, però l'hem de posar dins de mode matemàtic.

## Exemple (sessio1e.tex)

```
$$  
\left( \begin{array}{cc}  
a & b \\ c & d  
\end{array}  
\right)  
$$
```

$$\left( \begin{array}{cc} a & b \\ c & d \end{array} \right)$$

# Espais fantasmes

La comanda `\phantom` produeix un espai en blanc que ocupa el mateix espai que el que hem escrit com argument.



# Espais fantasmes

La comanda `\phantom` produeix un espai en blanc que ocupa el mateix espai que el que hem escrit com argument.

## Exemple (sessio1e.tex)

```

$$
{}^{\{12\}}_{\{6\}}C \ \mbox{el podem comparar amb }
{}^{\{12\}}_{\{\phantom{\{1\}}6\}}C
$$

```

${}^{12}_6 C$  el podem comparar amb  ${}^{12}_6 C$

# Projectes grans

Enlloc de treballar amb un document molt llarg, ens pot interessar dividir-lo en documents petits. Per a això tenim les comandes `\input` i `\include` (juntament amb `\includeonly`):

La comanda `\input{fitxer}` llegeix el text del fitxer com si l'escrivíssim a la posició on hi ha la comanda.

# Projectes grans

Enlloc de treballar amb un document molt llarg, ens pot interessar dividir-lo en documents petits. Per a això tenim les comandes `\input` i `\include` (juntament amb `\includeonly`):

La comanda `\input{fitxer}` llegeix el text del fitxer com si l'escrivíssim a la posició on hi ha la comanda.

La comanda `\include{fitxer}` fa un salt de plana i inclou el fitxer a la posició on hi ha la comanda.

Si a la capçalera hi ha la comanda `\include{fitxers}`, on `fitxers` és una llista de fitxers separats per coma, forcem que tant sols respongui a les comandes `\include` que criden a aquests fitxers.

Aquesta comanda genera el seu propi `.aux` i s'utilitza per a incloure els diferents capítols d'un llibre (veure `sessio1f.tex`).