

# Eines Informàtiques per a l'Estadística

## L<sup>A</sup>T<sub>E</sub>X

A. Alabert

16 de novembre de 2023

El millor resum de què és el L<sup>A</sup>T<sub>E</sub>X és aquesta pàgina, que es llegeix en un minut:  
<https://www.latex-project.org/about/>

## 1 Instal·lació

### 1.1 Distribucions

El L<sup>A</sup>T<sub>E</sub>X és programari lliure. N'existeixen diverses *distribucions*, totes molt similars. En realitat, parlem de *distribucions de T<sub>E</sub>X* (el T<sub>E</sub>X és el programari base, i podeu pensar el L<sup>A</sup>T<sub>E</sub>X com una ampliació de les funcionalitats del T<sub>E</sub>X).

A la mateixa pàgina web anterior, aneu a la pestanya **Get**. Veureu que hi ha distribucions per a Linux, MacOS i Windows. També hi ha la possibilitat d'usar un servei on-line, com el **Overleaf**, que pot ser útil per fer treball col·laboratiu, però és molt recomanable treballar habitualment en una instal·lació local.

### 1.2 Instal·lació del T<sub>E</sub>X

**Windows** <sup>1</sup> Seguiu el link MikTeX (<https://miktex.org>). Aneu a la pestanya **Download** i premeu el botó **Download**. Executeu el fitxer `.exe` que baixa, i seguiu les instruccions. Quan pregunta “for me” o “for anyone”, escolliu la primera opció.

**MacOS** <sup>2</sup> Seguiu el link MacTeX (<https://www.tug.org/mactex>). Seguiu **MacTeX Download Page** i **MacTeX.pkg**. Fent doble-click en aquest fitxer comença la instal·lació.

**Linux** Useu el gestor de paquets de la vostra distribució. Típicament, fareu  
`sudo apt install texlive-full`

---

<sup>1</sup>Els que sapiguen usar el gestor de paquets `winget` podeu fer  
`winget install -interactive -exact -id=MikTeX.MikTeX`  
des d'una consola i ja està.

<sup>2</sup>Els que tinguen instal·lat el gestor de paquets `homebrew` podeu fer  
`brew install -cask mactex`  
des d'una consola i ja està.

### 1.3 Instal·lació del TeXstudio

El L<sup>A</sup>T<sub>E</sub>X que acabeu d'instal·lar és una col·lecció de programes que processen fitxers de text, però no hi ha una interfície gràfica. Necessitem un editor per escriure aquests fitxers de text que després seran processats o *compilats* pel L<sup>A</sup>T<sub>E</sub>X, per obtenir un PDF final.

Qualsevol editor de fitxers pot anar bé (el Notepad del Windows, el TextEdit del MacOS, etc). Però és molt convenient usar un editor especialment dissenyat per al L<sup>A</sup>T<sub>E</sub>X, que ens permeti usar menús, botons, dreceres de teclat, que ens proporcionin “syntax highlighting” (colors), i que ens permeti veure ràpidament el resultat que es va obtenint. Recomanem el **TeXstudio**, tant en Linux, com en MacOS, com en Windows.

És altament recomanable instal·lar-lo *després* d'instal·lar el L<sup>A</sup>T<sub>E</sub>X, per tal que el TeXstudio el trobi i s'autoconfiguri automàticament. Aneu a <https://texstudio.org/>, secció Download.

**Windows** <sup>3</sup> Seguiu el link **Installer**, i executeu el `.exe` que es descarrega.

**MacOS** <sup>4</sup> Seguiu el link `dmg` (<https://www.tug.org/mactex>). Feu doble-click en el fitxer `.dmg` que es descarrega. Moveu `texstudio.app` a la carpeta **Applications**. Proveu d'obrir la app. Si es queixa que “the developer cannot be verified”, feu-hi click mentre premeu la tecla `<control>`, i seleccioneu **Open** (només caldrà fer-ho el primer cop). Observeu que requereix MacOS 11 (Big Sur) almenys.<sup>5</sup>

**Linux** Useu el gestor de paquets de la vostra distribució. Típicament, fareu `sudo apt install texstudio`.

## 2 Fonaments de L<sup>A</sup>T<sub>E</sub>X

Aquesta és una introducció molt resumida al L<sup>A</sup>T<sub>E</sub>X. El document de referència recomanat és *The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X*[1], que es pot trobar fàcilment en PDF a internet. Hi ha moltíssim material a l'abast a internet: manuals, fòrums, wikis, etc.

El L<sup>A</sup>T<sub>E</sub>X és un *typesetting system* que produeix documents de gran qualitat visual i ben estructurats. Es considera gairebé imprescindible per a documents amb contingut tècnic, com ara fórmules matemàtiques, i és l'estàndard en els llibres científics moderns.

Per ser precisos, el nucli de programari que permet fer tot això s'anomena T<sub>E</sub>X. En primera aproximació i sense entrar en detalls, es pot pensar que el L<sup>A</sup>T<sub>E</sub>X és una extensió del T<sub>E</sub>X que proporciona un conjunt addicional d'instruccions per ajudar a concentrar-se en el contingut i no en la forma. Al seu torn, el L<sup>A</sup>T<sub>E</sub>X és ampliable amb *packages* que es poden carregar opcionalment i separatament per a cada document.

<sup>3</sup>Amb el `winget`, feu  
`winget install -interactive -exact -id=TeXstudio.TeXstudio`

<sup>4</sup>Amb el `homebrew`, feu  
`brew install -cask texstudio`

<sup>5</sup>Si teniu una versió 10.8 a 10.15, aneu a <https://www.macupdate.com/app/mac/49122/texstudio/old-versions> i baixeu la versió 2.12.6. Dins el fitxer `.zip` hi ha la carpeta `.app` que heu de moure a **Applications**.

## 2.1 Què ens cal per començar

Un editor de fitxers qualsevol seria suficient, però ens hem instal·lat el `TeXstudio` perquè està pensat específicament per a `LATEX`.

El fitxer contindrà el text que es vol escriure, intercalat amb instruccions per als aspectes de forma del document i per als caràcters i símbols que no es poden entrar amb el teclat (per exemple, `\sum` produeix  $\Sigma$ ).

També es necessita un visor de fitxers PDF extern (com ara l'Adobe Reader, o qualsevol altre) si es vol imprimir o visualitzar el resultat independentment del `TeXstudio`.

## 2.2 Un document minimal

---

```
\documentclass{article}
\begin{document}
Hola, món!      Mireu què he escrit: % No intenteu traduir!

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat ...

\end{document}
```

---

Suposem que hem guardat aquestes línies en un fitxer de nom `myfile.tex`.

Suposem per un moment que no tenim el `TeXstudio`, que com hem dit facilita molt la feina però és un accessori no imprescindible. Obrim una consola, i naveguem al directori on es troba el fitxer<sup>6</sup>. Escrivim

```
pdflatex myfile.tex
```

El programa `pdflatex` *compila* el fitxer, indica si hi ha errors a corregir o avisos a considerar i produeix el fitxer final `myfile.pdf`. Aquest l'obrirem amb el nostre visor de PDF per comprovar com ha quedat.

En el procés es creen una sèrie de fitxers auxiliars que podeu esborrar tranquil·lament quan vulgueu; només cal conservar el fitxer font `.tex` i el `.pdf` final. El `.tex` és de fet l'únic realment important a conservar, perquè amb ell podem tornar a produir el `.pdf`; a l'inrevés no.

Si estem en el `TeXstudio`, podem fer el procés més àgil: En el menú `Tools` tenim l'entrada `Build and view`, que automàticament crida al programa `pdflatex`, ens escriu en un panel els missatges d'error o avisos, i obre el PDF en un altre panel, en el punt en què tenim el cursor de l'editor. Equivalentment, podem usar les entrades `Compile` i `View` per fer-ho en dos passos, que de vegades va bé. Equivalentment, podem usar els botons

---

<sup>6</sup>Vegeu l'apèndix al final sobre la consola.

anàlegs de la barra d'eines superior. Equivalentment, podem usar les dreceres de teclat assignades en els menús, que és el que tots acabem fent per anar encara més ràpid.

Està clar que rara vegada usarem la consola. Però convé tenir-la present pels motius següents (llegiu! l'últim és el més important):

- Si alguna cosa no funciona, convé comprovar si és culpa del `TeXstudio` mal configurat o el `TeX` mal instal·lat.
- Amb la consola, la compilació s'aturarà en el primer error i ens dirà quina és la línia que no ha pogut entendre. El `TeXstudio` en canvi no s'atura, i ens pot regalar una llarga llista d'errors que de vegades costen de llegir<sup>7</sup>.
- En situacions en què el resultat no és l'esperat, el `TeXstudio` de vegades oculta informació rellevant.
- Si s'han de compilar molts fitxers a la vegada, pot ser més ràpid fer-ho des de la consola. Es pot fer un *script* que compili tots els fitxers d'un directori amb una sola instrucció de consola.
- Els treballs de `LATEX` es corregiran amb el procediment del punt anterior. Per tant: Abans d'entregar-lo, useu la consola!

Tant si usem el `TeXstudio` com si no, la sortida de l'exemple anterior serà:

Hola, món! Mireu què he escrit:  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat ...

Observeu que:

- La línia en blanc ha provocat un canvi de paràgraf (la instrucció `\par` fa el mateix).
- Els paràgrafs tenen una indentació en la primera línia.
- Espais en blanc entre paraules queden reduïts a un sol espai (i diverses línies seguides en blanc equivalen a una sola).
- El que hi ha darrera del símbol `%` s'ignora. Serveix per deixar comentaris.

Es dedueix del que hem fet que el `LATEX` usa certs caràcters amb fins especials. Aquests són

`% $ ^ & _ { } ~ # \`

---

<sup>7</sup>Tant el funcionament des de la consola com des del `TeXstudio` es poden configurar per fer-ho diferent, però no tocarem aquesta qüestió aquí.

Per escriure realment aquests caràcters, precediu-los per un *backslash* `\`, excepte l'últim, que s'escriu amb la instrucció `\textbackslash`.

El  $\text{T}\text{E}\text{X}$  suposa d'entrada que estem escrivint en un idioma amb alfabet llatí, sense cap mena d'accent (ni tampoc `ñ`, `ç`, ``, `ñ`, ...), juntament amb els símbols tipogràfics més habituals (parèntesis, comes, etc) i els números. En resum, bàsicament el  $\text{T}\text{E}\text{X}$  pressuposa que estem escrivint en anglès.

Aquests símbols bàsics, que no arriben a cent, queden codificats sempre de la mateixa manera a l'interior de tots els ordinadors. És el que s'anomena l'estàndard ASCII<sup>8</sup>. En canvi, per a les lletres accentuades hi ha diversos estàndards, que depenen de l'editor que estiguem utilitzant. En Windows, dins la nostra àrea geogràfica, típicament els editors guarden els caràcters en la codificació coneguda com ISO-8859-1, també anomenada `latin1`; en Linux i en MacOS, és més típica la codificació UTF-8. I són absolutament incompatibles!

Naturalment, un bon editor ens ha de deixar escollir quina codificació volem utilitzar. El `TeXstudio`, a la barra de status, abaix a la dreta, ens diu la codificació que està utilitzant. I ens la deixa canviar.

Com que el  $\text{T}\text{E}\text{X}$  processarà el document des de fora del `TeXstudio`, cal que li diguem quin és el *input encoding*, o sigui en quina codificació està el fitxer que li estem enviant. Això ho farem posant, entre el `\documentclass` i el `\begin{document}`, la instrucció<sup>9</sup>

```
\usepackage[latin1]{inputenc}
```

o bé

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

Es convenient també especificar un *font encoding*, per raons que serien llargues d'explicar aquí. Recomanem usar sempre

```
\usepackage[T1]{fontenc}
```

per obtenir un PDF de qualitat òptima, a qualsevol resolució de pantalla o d'impressora.

Si cal tallar una paraula al final de la línia, com es fa? El  $\text{L}\text{A}\text{T}\text{E}\text{X}$  ho fa automàticament, però cada idioma té regles diferents. Aquesta i altres subtiletes relacionades amb l'idioma fan convenient declarar en quin idioma estem escrivint, amb

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

(o `[spanish]`, `[french]`, etc).<sup>10</sup> Per escriure en anglès, no cal posar res.

**Exercici:** Amb el `TeXstudio`, feu un fitxer que contingui les instruccions de què hem parlat, i amb el text que vulgueu. Guardeu-lo donant-li un nom que acabi en `.tex`. Useu el menú **Build and view** per compilar-lo, i comproveu que es veu bé en el panel del visor de pdf. Si teniu un visor extern de PDF, segurament el `TeXstudio` l'haurà trobat al instal·lar-se i un botó que hi ha a la barra sobre el panel us permetrà obrir-lo externament. És el botó que té *tooltip* **External Viewer**.

---

<sup>8</sup>Podeu veure la taula de caràcters ASCII a la Wikipedia.

<sup>9</sup>Des del 2018, UTF-8 és la codificació per ommissió en  $\text{L}\text{A}\text{T}\text{E}\text{X}$  si no es posa res. No obstant, és recomanable posar la instrucció, perquè el fitxer és compili sempre correctament en diferents ordinadors.

<sup>10</sup>En particular, l'opció `[catalan]` defineix la instrucció `\l.1`, que produeix una ela geminada tipogràficament correcta, i ben tallada al final de línia. No useu el punt volat que hi ha al teclat.

### 3 Fórmules matemàtiques

Els paquets `amsmath`, `amssymb` i `amsthm` afegeixen moltes instruccions al  $\text{\LaTeX}$  bàsic per escriure símbols i fórmules matemàtiques. Per no haver de distingir què es pot fer amb ells i què no, els carregarem sempre. Per tant, a partir d'ara suposarem que el nostre fitxer comença sempre amb aquestes línies, almenys, entre el `\documentclass` i el `\begin{document}`:

---

```
\documentclass{article}
\usepackage[utf8]{inputenc} %alternativa: [latin1]
\usepackage[T1]{fontenc}
\usepackage[catalan]{babel} %alternativa: altres idiomes
\usepackage{amsmath, amssymb, amsthm}
\begin{document}
```

---

Una fórmula matemàtica dins d'una línia de text, com ara  $x^2 + y^2 = 1$ , s'escriu així: `\$x^2+y^2=1\$`. Observeu com es posen superíndexs. Els subíndexs es posen usant el blanc subratllat `_`. Les claus `{ i }` s'utilitzen a tot arreu en  $\text{\LaTeX}$  per agrupar. Per exemple,  $W_{[a,b]}^{1,2}$  s'obté amb `\$W^{1,2}_{[a,b]}\$`.

Com heu vist, les “instruccions” de  $\text{\LaTeX}$  (tècnicament anomenades *control sequences*) comencen amb un *backslash* `\`. Hi ha seqüències de control per a tots els símbols matemàtics que pogueu trobar. Per exemple, observeu atentament les seqüències de control següents

```
\$ \lim_{n\to\infty} \sum_{k=1}^n \frac{1}{k^2} \ge
\int_1^{\infty} \frac{1}{x^2} \, dx \$
```

i el resultat que produeixen:  $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} \geq \int_1^{\infty} \frac{1}{x^2} dx$ .

Aquesta fórmula potser ens agradaria més en un *display*, o sigui en una línia per a ella sola, adequadament separada de les línies superior i inferior. Això s'aconsegueix posant-la en un *environment equation*, de la manera següent:

```
\begin{equation}
\lim_{n\to\infty} \sum_{k=1}^n \frac{1}{k^2} \ge
\int_1^{\infty} \frac{1}{x^2} \, dx
\end{equation}
```

que produeix:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} \geq \int_1^{\infty} \frac{1}{x^2} dx \quad (1)$$

Observeu:

- Els espais i canvis de línia en les fórmules no tenen efecte.
- Els símbols `^` i `_` per als superíndexs i subíndexs s'interpreten adequadament quan s'utilitzen rere els operadors `lim`, `sum`, `int`.

- L'aparença de la fórmula és diferent dins una línia de text (*inline*) o en un *display*. Els operadors com ara `\sum` i `\int` tenen una forma diferent, i la posició de llurs sub i superíndexs canvia.
- Els *environments* (entorns) apareixen molt en  $\text{\LaTeX}$ , i sempre es delimiten amb les seqüències `\begin{ }` i `\end{ }`.
- Les fórmules es numeren consecutivament en un document. Si no volem que es numeri una fórmula, usem l'entorn `equation*`.
- `\ge` produeix el símbol de *greater or equal*. Podeu endevinar com es posa el de *less or equal*?
- Què fa la seqüència `\, ?`  
Introdueix un petit espai (vegeu l'Apartat 5). El  $\text{\LaTeX}$  col·loca els espais raonablement bé a tot arreu, però de vegades voldrem controlar-los una mica. En aquesta fórmula concreta és qüestió de gustos.

A l'apartat 3.10. *List of Mathematical Symbols* de [1] trobareu una bona col·lecció de símbols per utilitzar en fórmules matemàtiques. El `TeXstudio` també ens n'ofereix un catàleg per introduir-los amb un click. Encara més exhaustiva, la col·lecció del document de 348 pàgines *The Comprehensive \LaTeX Symbol List* [2].

Anem a veure unes quantes construccions de *displays* útils per a fórmules llargues.

El *environment* `multline` permet escriure una fórmula en més d'una línia, la primera moguda cap a l'esquerra, l'última cap a la dreta, i les del mig centrades:

```
\begin{multline}
(a + b)^4
= (a + b)^2 (a + b)^2
= (a^2 + 2ab + b^2)(a^2 + 2ab + b^2) \\
= a^4 + 2a^3b + a^2b^2 + 2a^3b + 4a^2b^2 + 2ab^3 + a^2b^2 + 2ab^3 + b^4 \\
= a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4
\end{multline}
```

$$\begin{aligned}
 (a + b)^4 &= (a + b)^2(a + b)^2 = (a^2 + 2ab + b^2)(a^2 + 2ab + b^2) \\
 &= a^4 + 2a^3b + a^2b^2 + 2a^3b + 4a^2b^2 + 2ab^3 + a^2b^2 + 2ab^3 + b^4 \\
 &= a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4 \quad (2)
 \end{aligned}$$

És un estil que es troba sobretot en llibres antics, i que està passat de moda. És més habitual alinear les fórmules en algun punt que ajudi a transmetre el significat. Podem usar l'entorn `split` dins l'entorn `equation`:

```
\begin{equation}
\begin{split}
(a + b)^3 &= (a + b)(a + b)^2 && \\
&= (a + b)(a^2 + 2ab + b^2) && \\
&= a^3 + 3a^2b + 3ab^2 + b^3
\end{split}
\end{equation}
```

$$\begin{aligned}
 (a + b)^3 &= (a + b)(a + b)^2 \\
 &= (a + b)(a^2 + 2ab + b^2) \\
 &= a^3 + 3a^2b + 3ab^2 + b^3
 \end{aligned}
 \tag{3}$$

o bé l'entorn independent `align`:

```

\begin{align}
(a + b)^3 &= (a + b)(a + b)^2 & \\\
&= (a + b)(a^2 + 2ab + b^2) & \\\
&= a^3 + 3a^2b + 3ab^2 + b^3
\end{align}

```

$$(a + b)^3 = (a + b)(a + b)^2 \tag{4}$$

$$= (a + b)(a^2 + 2ab + b^2) \tag{5}$$

$$= a^3 + 3a^2b + 3ab^2 + b^3 \tag{6}$$

Veieu la petita diferència que hi ha entre els resultats de `split` i `align`?

Si volem posar diverses fórmules centrades una sota l'altre, l'entorn `gather` és l'adequat:

```

\begin{gather}
(a + b)^2 = a^2 + 2ab + b^2 & \\\
(a + b) \cdot (a - b) = a^2 - b^2
\end{gather}

```

$$(a + b)^2 = a^2 + 2ab + b^2 \tag{7}$$

$$(a + b) \cdot (a - b) = a^2 - b^2 \tag{8}$$

Quan es vol alinear en diferents punts, es pot usar `alignat`. Aquí volem alinear en tres punts, i ho diem en el paràmetre addicional que especifiquem. Però de símbols `&` n'hi ha el doble menys un a cada línia.

```

\begin{alignat}{3}
x_1&= y_1-2&&y_2+y_3&&-5y_4+y_5 & \\\
x_2&= y_1+&\}&y_2 & &&+7y_4
\end{alignat}

```

$$x_1 = y_1 - 2y_2 + y_3 - 5y_4 + y_5 \tag{9}$$

$$x_2 = y_1 + y_2 + 7y_4 \tag{10}$$

Amb `alignedat` es pot fer el mateix, però, com `split`, s'ha de posar dins de l'entorn `equation` (es produirà un sol número de fórmula):

```

\begin{equation}
\begin{alignedat}{3}
x_1&= y_1-2&&y_2+y_3&&-5y_4+y_5 & \\\
x_2&= y_1+&\}&y_2 & &&+7y_4
\end{alignedat}
\end{equation}

```



$$\begin{aligned}x_1 &= y_1 - 2y_2 + y_3 - 5y_4 + y_5 \\x_2 &= y_1 + y_2 + 7y_4\end{aligned}\tag{11}$$

Hi ha alineacions més sofisticades, però no les farem aquí<sup>11</sup>. En tot cas, prohibit usar l'entorn `eqnarray` citat (i desaconsellat) a [1]. Hi ha gent que l'utilitza, però el resultat és tipogràficament nefast.

**Exercici:** Proveu de reproduir

$$\begin{aligned}P\{b_{j-1} \leq X < b_j\} &= \int_{b_{j-1}}^{b_j} f(x) dx \\ &= \Delta b \cdot f(y)\end{aligned}$$

## 4 Estructura del document

La idea central del  $\text{\LaTeX}$  és que l'autor es pugui concentrar en el contingut i no en la forma. Com a autors, li direm que posi un títol d'apartat o que faci una llista numerada, però no li direm *com* ho ha de fer. Això no vol dir que no es pugui canviar res; al contrari, tot és controlable fins al més mínim detall, però requereix pràctica, coneixements i bon gust. Mentre els anem adquirint, deixem que el  $\text{\LaTeX}$  faci la feina.

A l'inici de cada fitxer font hi ha d'haver la instrucció `\documentclass{...}`, que li diu al  $\text{\LaTeX}$  quin tipus de document estem creant. Els més importants són `article`, `report`, i `book`. La majoria de treballs que fem són de tipus `article`.

Aquesta instrucció admet *opcions* que es posen entre claudàtors [ ], com per exemple el tipus de paper i la mida de la lletra: `\documentclass[a4paper, 11pt]{article}`.

El *contingut* del document es trobarà entre les instruccions `\begin{document}` i `\end{document}`.

Les instruccions que hi ha entre `\documentclass{...}` i `\begin{document}` formen el *preàmbul*. Aquí posem instruccions `\usepackage{...}` per carregar *paquets* que estenen les capacitats del  $\text{\LaTeX}$  (inclourem els que necessitem per a cada document concret), i potser altres instruccions que estaran en vigor en tot el document.

Disposem de diverses instruccions per donar estructura al document. Cal usar-les per tal que la tipografia sigui uniforme, i es pugui, opcionalment, generar una *table of contents* (índex), i també referenciar-les dins del propi document. Tenim, de nivell més alt a més baix<sup>12</sup>,

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

<sup>11</sup>Podem veure més possibilitats a [https://en.wikibooks.org/wiki/LaTeX/Advanced\\_Mathematics#align\\_and\\_align\\*](https://en.wikibooks.org/wiki/LaTeX/Advanced_Mathematics#align_and_align*).

<sup>12</sup>Dependent del `documentclass`, algunes d'aquestes no estaran disponibles; per exemple, `part` i `chapter` no es poden usar en la classe `article`, però sí en la `book`.

Per exemple, la instrucció

```
\section{Fórmules matemàtiques}\label{sec:formules}
```

produeix una nova secció, numerada correctament, amb el títol que s'indica, que a més és guarda per generar l'índex després. En qualsevol lloc del document es pot escriure

```
A l'apartat \ref{sec:formules}, que comença a la
pàgina \pageref{sec:formules}, hi ha moltes fórmules.
```

i el resultat serà (i podeu comprovar que és correcte)

A l'apartat 3, que comença a la pàgina 6, hi ha moltes fórmules.

El mecanisme `label/ref` o `label/pageref` s'utilitza també per referenciar fórmules, figures, etc. Si posteriorment inserim o esborrem text, les referències s'actualitzaran correctament (pot ser que calgui compilar el fitxer dues vegades seguides, però el `TeXstudio` ja ho fa automàticament si ho veu necessari).

Freqüentment hem de fer enumeracions i llistes.

```
\begin{itemize}
\item SageMath
\item R
\item Python
\item \LaTeX
\item Bash
\end{itemize}
```

produeix

- SageMath
- R
- Python
- $\LaTeX$
- Bash

L'entorn `description` s'usa per a etiquetar cada entrada d'una manera diferent. Vegeu el subapartat 2.11.1. *Itemize, Enumerate and Description* de [1].

L'entorn `enumerate` produeix una llista numerada:

```
\begin{enumerate}
  \item Examen al novembre:
  \begin{itemize}
    \item SageMath
    \item R
  \end{itemize}
  \item Treballs:
  \begin{itemize}
    \item Python
    \item \LaTeX
  \end{itemize}
  \item Examen al gener:
  \begin{itemize}
    \item Bash
  \end{itemize}
\end{enumerate}
```

1. Examen al novembre:

- SageMath
- R

2. Treballs:

- Python
- $\LaTeX$

3. Examen al gener:

- Bash

Com veieu, es poden combinar diferents entorns de llistes uns dins els altres. Existeix a més un *package* anomenat `enumerate` que permet configurar molt les enumeracions. Per exemple,

```
\begin{enumerate}[{Punt} A:]
  \item Els números enters.
  \item Els números racionals.
  \item Les lletres.
\end{enumerate}
```

Punt A: Els números enters.

Punt B: Els números racionals.

Punt C: Les lletres.

Al escriure matemàtiques, de vegades necessitem destacar definicions, proposicions, teoremes, .... El mecanisme s'explica a l'apartat 3.9. *Theorems, Lemmas,...* de [1]. Cal definir explícitament cadascun d'aquests elements, que en L<sup>A</sup>T<sub>E</sub>X s'anomenen genèricament *theorems*. Per exemple, posant en el preàmbul

```
\newtheorem{prop}{Proposició}
```

queda definida una classe d'enunciats que es numeraran consecutivament. Ara podem escriure

```
\begin{prop}
Tres rectes qualssevol del pla es tallen en un punt,
si el punt es prou gros.
\end{prop}
```

i obtenim

**Proposició 1.** *Tres rectes qualssevol del pla es tallen en un punt, si el punt es prou gros.*

I podem demostrar la proposició amb l'entorn `proof`:

```
\begin{proof}
Aneu fent gros el punt fins que les tres interseccions quedin dins.
\end{proof}
```

*Demostració.* Aneu fent gros el punt fins que les tres interseccions quedin dins. □

El títol d'un document, en la classe `article`, es defineix amb `\title{ }`, i l'autor amb `\author{ }`. L'ordre `\maketitle` els fa aparèixer explícitament.

L'ordre `\tableofcontents` fa aparèixer l'índex de capítols, seccions, etc. Es poden crear altres índexs. Per exemple, el típic índex de paraules al final dels llibres. Per a això s'utilitza un programa apart que s'anomena `makeindex`. Vegeu els detalls a l'Apartat 4.2. *Indexing* de [1]. Un altre "índex" habitual és la llista de referències (bibliografia), explicat a apartat 4.1. *Bibliography*, que es pot fer fàcilment amb el programa `bibtex`.

Aquests programes estan en totes les distribucions de T<sub>E</sub>X. Es poden usar des de la consola, o es poden cridar des dels menús del TeXstudio.

L'últim element important en l'estructura del document és el *material flotant*. El veurem en l'Apartat 8.

**Exercici:** Proveu en un document les estructures que hem vist en aquest apartat. En particular, `\section`, `\subsection`, `\subsubsection` i useu la instrucció `\tableofcontents`. Proveu de posar alguna fórmula matemàtica senzilla dins un títol de secció, a veure com queda.

## 5 Espais i dimensions

Recordeu que els espais i els salts de línia en el fitxer font no s'acumulen, i si són dins d'una fórmula s'ignoren totalment. El  $\LaTeX$  posa els espais adequats entre paraules i al final de la frase, i distribueix el text en pàgines i dins de cada pàgina de manera força adequada. Només quan volem fer un control fi del resultat necessitarem el que s'explica en aquest apartat.

Si volem introduir expressament espais horitzontals, tant dins de text com en fórmules, podem usar les següents seqüències de control (ordenades de més petit a més gran, amb formes alternatives abreviades; les de la dreta produeixen espais negatius):

<code>\thinspace</code>	<code>\,</code>	<code>\negthinspace</code>	<code>\!</code>
<code>\medspace</code>	<code>\:</code>	<code>\negmedspace</code>	
<code>\thickspace</code>	<code>\;</code>	<code>\negthickspace</code>	
<code>\enskip</code>			
<code>\quad</code>			
<code>\qquad</code>			

i també la instrucció genèrica `\hspace`. Per exemple, `\hspace{10mm}` produeix un espai horitzontal de 10mm, com aquest. En  $\LaTeX$  hi ha quantitats que representen longituds, i han de portar unitats, com ara els mil·límetres en aquest exemple. Poden prendre valors negatius. Una altra unitat molt habitual en tipografia són els *punts* (`1pt=0.3528mm`). Per introduir espais verticals entre paràgrafs, tenim `\smallskip`, `\medskip`, `\bigskip`, que són espais amb una certa flexibilitat. Podem fer un salt vertical de mida fixada amb `\vspace{}`.

Hi ha a més dues unitats relatives a la mida de lletra que s'està usant: `em` és l'amplada d'una M, i `ex` és l'alçada d'una x.

Si volem forçar que el text següent vagi a una línia o una pàgina nova, hem d'explicitar-ho amb `\newline` (equivalentment `\\`) o `\linebreak`, i `\newpage` o `\pagebreak`. Les instruccions que porten `break` intenten justificar la línia i ajustar la pàgina a la seva mida (en la mesura del possible) mentre que les de `new` simplement tallen en el lloc on s'han indicat, i omplen la resta amb espai blanc.

Si posem `\documentclass{article}` el  $\LaTeX$  pensarà que volem imprimir en paper tipus *letter* (la mida estàndard a Amèrica), i en una mida de lletra bàsica de 10pt (que és la habitual en llibres, i que segurament trobareu massa petita per a altres finalitats). Com hem vist a l'Apartat 4, podem canviar-ho usant les opcions `[a4paper, 11pt]`.

Sabent la mida del paper, el  $\LaTeX$  calcula una sèrie de mesures per a la disposició (*layout*) de la pàgina: els marges, l'espai ocupat per capçaleres, els peus de pàgina, etc. Aquestes longituds es poden canviar. Vegeu a [1], apartat 6.4. *Page Layout*, la llista completa de paràmetres i com canviar-los, mitjançant les instruccions `\setlength` i `\addtolength`. És millor establir-los sempre en el *preàmbul* del document.

Per al control del tipus de lletra i la mida, vegeu l'apartat 6.2. *Fonts and Sizes* del [1]. Per a les opcions bàsiques de capçaleres i peus de pàgina (*headers/footers*) vegeu-ne el subapartat 1.6.3. *Page Styles*.

**Exercici:** Proveu en un document les instruccions `\smallskip`, `\medskip`, i `\bigskip`, que han d'anar entre paràgrafs. Proveu `\hspace{}` i `\vspace{}`, també amb dimensions negatives.

## 6 Taules i matrius

Per fer taules, l'entorn bàsic és `tabular`. Permet fer línies horitzontals i verticals i una certa flexibilitat per col·locar els elements de la taula. Per exemple:

Carl Friedrich Gauss	Florence Nightingale	Karl Pearson	← importants
William Gosset	Ronald Fisher	Gertrude Cox	

Aquesta taula l'ha produïda el codi de més avall, on s'aprecien els elements següents:

- El posicionament respecte la línia, que pot ser `[t]` (top), `[c]` (center), `[b]` (bottom).
- El *format*: `l c r` per alinear a esquerra, centre i dreta; `|` per una línia vertical.
- Les instruccions `\hline` per les línies horitzontals.
- Els `&` per separar les columnes; els `\\` per canviar de línia.

```
\setlength\extrarowheight{4pt}
\begin{tabular}[c]{|l|c|r|}
\hline
Carl Friedrich Gauss & Florence Nightingale & Karl Pearson \\
\hline
William Gosset & Ronald Fisher & Gertrude Cox \\
\hline
\end{tabular}
$\longleftarrow$ importants
```

A més, `\extrarowheight` és un paràmetre de dimensió, al qual es pot donar un valor usant `\setlength`. Per usar-lo cal carregar el package `array`, que estén les possibilitats de l'entorn `tabular`.

En fórmules matemàtiques de vegades cal fer també disposicions en forma de taula. Tres exemples:

```
\begin{equation}
P_{r-j} =
\begin{cases}
0 & \text{\textit{if } $r-j$ is odd,} \\
r! \cdot (-1)^{(r-j)/2} & \text{\textit{if } $r-j$ is even.}
\end{cases}
\end{equation}
```

$$P_{r-j} = \begin{cases} 0 & \text{if } r - j \text{ is odd,} \\ r! (-1)^{(r-j)/2} & \text{if } r - j \text{ is even.} \end{cases}$$



$$\left(\sum_{k=0}^n k^{-2}\right)^{1/2} \text{ segurament és millor que } \left(\sum_{k=0}^n k^{-2}\right)^{1/2}$$

**Exercicis:**

- a) Experimenteu amb les opcions de la taula del principi de l'apartat: El posicionament [c], l'alineació de les columnes l, c, r, la presència o absència de línies | i \hline, i el paràmetre de \extrarowheight.
- b) Experimenteu amb els delimitadors, usant \big, \Big, \bigg, \Bigg.

## 7 Gràfics

*Nota:* Els subapartats 7.2 i 7.3 són opcionals en el sentit que no els necessitareu per fer el treball.

### 7.1 Inclusió de gràfics externs

Podem incloure directament en el document un fitxer en format gràfic. Necessitarem

```
\usepackage{graphicx}
```

en el preàmbul.

Suposem que tenim una foto en un fitxer en format JPEG, anomenat `emperador.jpg`. L'ordre `\includegraphics{emperador.jpg}` l'inclou en el document.



Aquí hem posat

```
\begin{center}
\includegraphics[scale=0.3]{emperador.jpg}
\end{center}
```

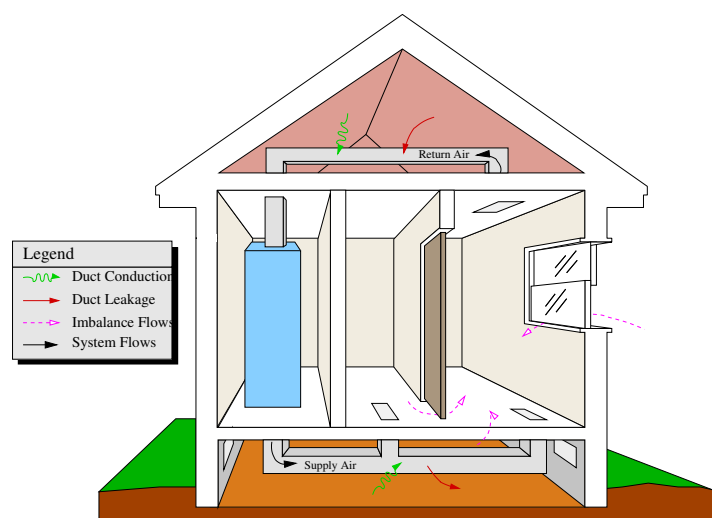


L'entorn `center` òbviament serveix per centrar la imatge i deixa espais convenients a dalt i a baix. L'opció `[scale=0.3]` fa la foto més petita (o més gran) en la proporció indicada.

Apart del JPEG, també podem incloure PNG, GIF, PDF i EPS. Aquest últim (*Encapsulated PostScript*), necessita un pas de conversió a PDF, que es fa automàticament. Per exemple, per incloure el fitxer `3dhouse.eps`, fem

```
\begin{center}
\includegraphics{3dhouse.eps}
\end{center}
```

i produeix



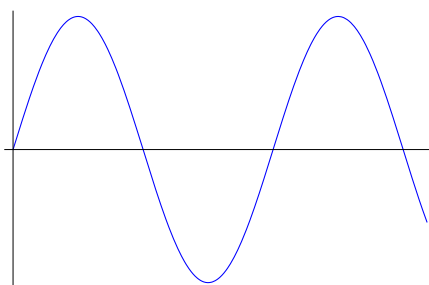
però en realitat inclou un fitxer anomenat `3dhouse-eps-converted-to.pdf` que s'ha creat sobre la marxa i que trobaríem en el mateix directori.

L'ordre `\includegraphics` admet diverses opcions per manipular la imatge en el moment d'incloure-la. En podeu veure unes quantes a l'Apartat 2.12. *Including Graphics and Images* de [1].

El Sage ens permet guardar gràfics (i més coses, vegeu 7.3) que després es poden incloure en el document  $\text{\LaTeX}$ . Per exemple,

```
plot(sin(x), (x, 0, 10), ticks=[[ ], [ ]]).save("sin.pdf")
```

produeix la figura

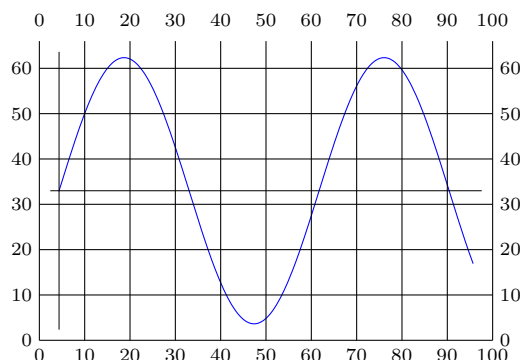


que hem inclòs aquí fent

```
\begin{center}
\includegraphics[scale=0.4]{sin.pdf}
\end{center}
```

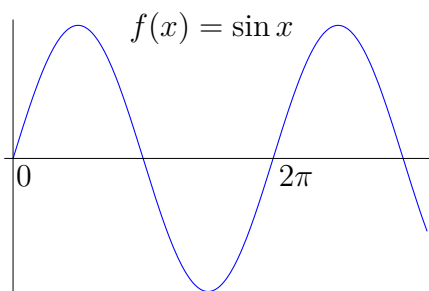
Si volem posar símbols, fórmules o text en els gràfics, una manera de fer-ho és usar el paquet `overpic`, que permet posar una graella provisional sobre el dibuix:

```
\begin{overpic}
  [grid, tics=10, scale=0.4]{sin.pdf}
\end{overpic}
```



Amb l'ajuda de la graella, podem posicionar fàcilment una etiqueta en el lloc desitjat.

```
\begin{center}
\begin{overpic}
  [scale=0.4]{sin.pdf}
  \put(30, 60){$f(x)=\sin x$}
  \put( 5,27){$0$}
  \put(63,27){$2\pi$}
\end{overpic}
\end{center}
```



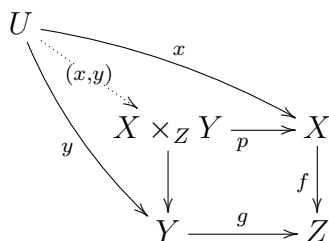
**Exercici:** Agafeu tres fotos JPG qualssevol i poseu-les petites i equiespaiades en una sola línia, com ara aquí: (*Idea:* Busqueu ajuda per a l'instrucció `\hfill`.)



## 7.2 Gràfics amb L<sup>A</sup>T<sub>E</sub>X

Hi ha paquets de L<sup>A</sup>T<sub>E</sub>X específics per a crear, de manera relativament fàcil, i sense sortir de l'ambient del propi L<sup>A</sup>T<sub>E</sub>X, figures sofisticades.

Per exemple, el paquet `xypic` permet dibuixar



amb les instruccions

```

\begin{equation*}
\begin{array}{c}
\begin{array}{c}
U \xrightarrow{\quad} Y \xrightarrow{\quad} X \\
\downarrow \quad \downarrow \quad \downarrow \\
X \times_Z Y \xrightarrow{p} X \\
\downarrow \quad \downarrow \quad \downarrow \\
Y \xrightarrow{g} Z
\end{array}
\end{array}
\end{equation*}

```

A les Figures 1, 2, 3 teniu alguns exemples dels resultats que es poden aconseguir amb paquets especialitzats en diferents àmbits, diferents de les matemàtiques.

Altres solucions que permeten fer elements gràfics generals amb el L<sup>A</sup>T<sub>E</sub>X mateix, són l'entorn `picture` i els paquets `tikz` i `PGF` (vegeu [1], capítol 5. *Producing Mathematical Graphics*). Una mostra del que es pot fer amb aquests últims la trobareu a <http://www.texample.net/tikz/>.

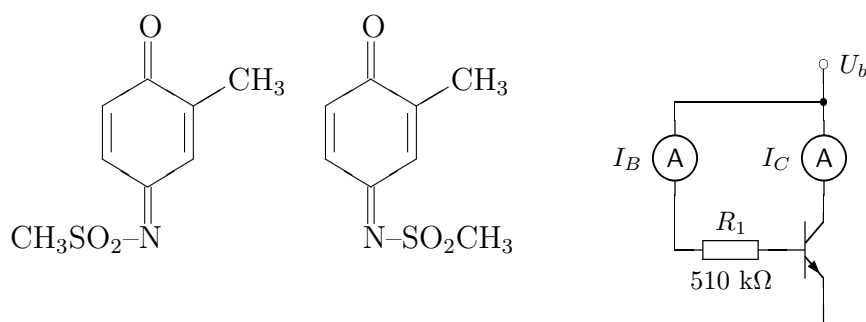


Figura 1: Química orgànica i electrònica

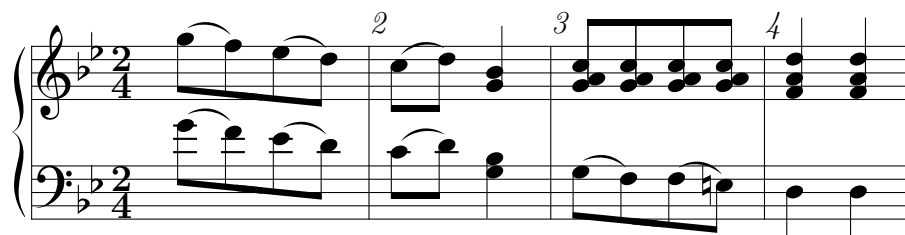


Figura 2: Música

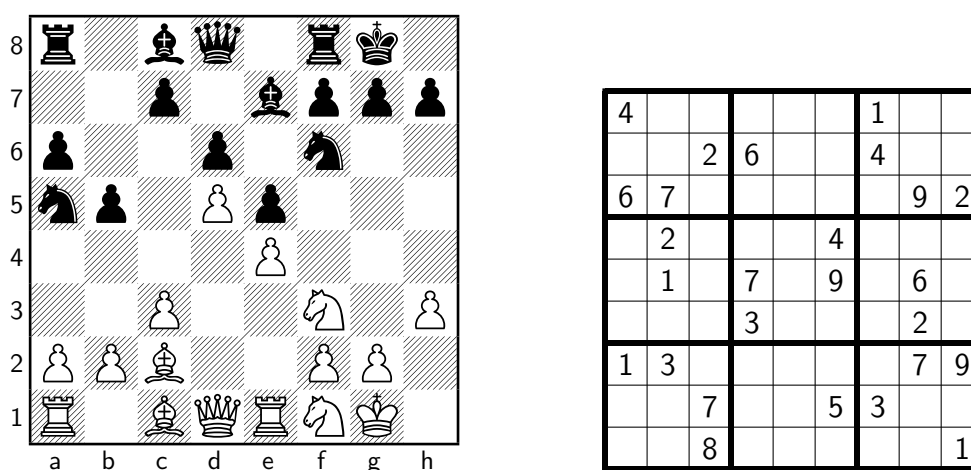


Figura 3: Escacs i sudokus

### 7.3 Interacció amb SageMath/Jupyter

Obriu un notebook de SageMath nou i executeu, per exemple,

```
var('y')
polinomi= x^3 + 3*x^2*y^3 - 2*y^2*x + 3*y + 5
```

Qualsevol expressió que hàgiu definit en el notebook es pot convertir al format  $\LaTeX$  mitjançant la instrucció `latex` de SageMath:

```
latex(polinomi)
```

El resultat està llest per copiar i enganxar en un document de  $\LaTeX$ . És un truc que va bé saber i ocasionalment ens pot estalviar temps, però amb la pràctica preferirem escriure directament en  $\LaTeX$ .

Recíprocament, i potser més interessant, la sintaxi de  $\LaTeX$  es pot usar en instruccions de SageMath, com en les etiquetes (`legend_label`) dels gràfics.

## 8 Material flotant

A l'apartat anterior tenim tres figures que van numerades, i amb una llegenda. Aquestes figures són *material flotant*. No apareixen exactament en el lloc on estan posades en el fitxer font, sinó que el  $\LaTeX$  les situa de manera automàtica.

Carl Friedrich Gauss	Florence Nightingale	Karl Pearson
William Gosset	Ronald Fisher	Gertrude Cox

Taula 1: Algunes persones importants

Hi ha dos tipus bàsics de material flotant: Les figures i les taules. Les figures es defineixen amb l'entorn `figure` i les taules amb l'entorn `table`. Per exemple,

```
\begin{figure}
  material-de-la-figura
\caption{La llegenda de la figura}
\label{etiqueta}
\end{figure}
```

Se la pot referenciar amb `\ref{etiqueta}`, de manera similar a seccions i fórmules.

A la Taula 1 tenim la mateixa taula del Apartat 6, posada com a material flotant dins de l'entorn `table`. El codi corresponent és:

```
\begin{table}
\centering
\begin{tabular}[c]{|l|c|r|}
\hline
Emmy Noether & Bernhard Riemann & Gottfried Leibniz \\
\hline
Carl Friedrich Gauss & Henri Poincaré & Sofya Kovalevskaya \\
\hline
\end{tabular}
\caption{Algunes persones importants}
\label{importants}
\end{table}
```

Normalment volem que la figura o taula quedi centrada horitzontalment. Per això la instrucció `centering`. Aquí no és massa correcte usar l'entorn `center`, que afegeix espais a dalt i abaix, perquè els entorns `table` i `figure` ja posen l'espaiat vertical que cal.

**Exercici:** Convertiu les tres fotos de l'exercici de l'apartat 7.1 en material flotant, posant-hi un *caption* i referencieu la figura des del text.

## 9 Creació de noves instruccions

Per denotar el conjunt dels nombres naturals, escrivim habitualment  $\mathbb{N}$ . Aquest tipus de lletra especial s'anomena *blackboard bold* i s'obté fent `\mathbb{N}`. Però si hem de referir-nos molts cops a aquest conjunt, es fa pesat repetir aquesta la instrucció cada vegada.

El que voldríem és que una seqüència de control curta, per exemple, `\N`, o fàcil de recordar, com ara `\Naturals`, fes tota la feina. Això s'aconsegueix així

```
\newcommand{\N}{\mathbb{N}}
```

A partir del moment en què es fa això, la seqüència de control `\N` és a tots els efectes equivalent a `\mathbb{N}`.

Si intentem definir una instrucció que ja ha estat definida abans, sigui per nosaltres o sigui una instrucció estàndard, provocarem un error. En tal cas, o bé escollim un altre nom per a la nova instrucció o bé “redefinim” la instrucció. Només cal substituir `\newcommand` per `\renewcommand`.

Suposeu que heu d'escriure moltes vegades expressions com ara  $(x_1, \dots, x_n)$ ,  $(y_1, \dots, y_m)$ , etc, que s'assemblen però difereixen una mica. Podem definir una instrucció amb paràmetres:

```
\newcommand{\vect}[2]{(#1_1,\dots,#1_#2)}
```

El `[2]` indica que hi ha dos paràmetres. El primer està representat per `#1` i el segon per `#2`. Podrem aleshores escriure

```
$ \vect{x}{n}, \vect{y}{m} $
```

per obtenir  $(x_1, \dots, x_n)$ ,  $(y_1, \dots, y_m)$ . Per més informació, consulteu [1], apartat 6.1. *New Commands, Environments and Packages*.

## Apèndix: La consola

Si no heu usat mai la *consola* (o *terminal*) del vostre sistema, aquí teniu instruccions per obrir-la de diverses maneres:

Linux (Ubuntu): [https://www.wikihow.com/](https://www.wikihow.com/Open-a-Terminal-Window-in-Ubuntu)

Open-a-Terminal-Window-in-Ubuntu

MacOS: [https://www.idownloadblog.com/](https://www.idownloadblog.com/2019/04/19/ways-open-terminal-mac/)

2019/04/19/ways-open-terminal-mac/

Windows: [https://www.isunshare.com/](https://www.isunshare.com/windows-10/4-ways-to-open-command-prompt-in-windows-10.html)

windows-10/4-ways-to-open-command-prompt-in-windows-10.html

Situeu-vos en el directori on teniu el fitxer `.tex`. Podeu veure el contingut d'un directori escrivint `dir` o `ls`; podeu entrar en un subdirectori fent `cd nom_directori`, i podeu pujar a un directori superior amb `cd ...`

Quan estigueu en el directori del fitxer, escriviu `pdflatex nom_fitxer.tex`.

**Exercici:** Proveu-ho!

## Referències

- [1] T. Oetiker, H. Partl, I. Hyna, E. Schlegl. *The Not So Short Introduction to LaTeX*. Version 6.3, March 26, 2018 (en PDF a internet).
- [2] S. Pakin. *The Comprehensive LaTeX Symbol List*. Version 25 June 2020 (en PDF a internet).
- [3] Wikibooks: L<sup>A</sup>T<sub>E</sub>X. <http://en.wikibooks.org/wiki/LaTeX>