Introduction to LaTeX

Dr. Yihsiang Liow (July 12, 2025)

Contents

1 Hello world

I assume you have access to my Fedora virtual machine, you know how to run the virtual machine, you know how to edit a file, and you know how to enter commands (in most cases, you can use the GUI).

First create a file, `main.tex`, containing the following:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
hello world
\end{document}
```

We're going to generate a `main.pdf`. At your bash shell prompt execute the following command

```
pdflatex main.tex
```

You now have a `main.pdf`. Open it and at the top of this pdf document you see hello world:

```
hello world
```

TeX is a typesetting system used extensively in scientific typesetting (especially CS, Engineering, Math, and Physics) and in the publishing industry. It was originally developed by a famous Math and CS professor at Stanford, Donald Knuth. LaTeX is built on top of TeX. From now on, I'll just use tex to refer to both TeX and LaTeX.

This is a short introduction to tex. Each short section explains some feature of tex and ends with a text box containing the tex code for that section. The information you want to build into your pdf document is written in

```
...
\begin{document}
... type your contents here ...
\end{document}
```

For each assignment, I will give you a folder containing several files including `main.tex`. To make it easier for you to build your pdf, I've included a special "makefile" so that in order to build your pdf, you only need to execute the command:

```
make
```

or if you're using the GUI, you just double-click on the `makefile`. This will build `main.pdf` from `main.tex`. Instead of putting all the tex code in `main.tex`, you should edit the files `q1.tex`, `q2.tex`, etc. for question 1, 2, etc. When you execute the command `make`, the file `main.tex` will read `q1.tex`, `q2.tex`, etc. The files might be named `q01.tex`, etc., or `q01a.tex`, etc.

To view the pdf, you execute the command

```
make view
```

or

```
make v
```

if you're using the GUI, you double-click on the pdf file.

There is one thing you have to do to `main.tex`: Near the top of the file, just before `\begin{document}` you should add a command to change my name to yours. For instance if you're John Doe you do this to `main.tex`:

```
...
\renewcommand\AUTHOR{John Doe}
\begin{document}
...
```

When you generate the pdf, your name will appear at the bottom left of every page.

If you have problems with LaTeX the best is to ask me or talk to the some CS students. Most CS students who took higher level CS classes, especially Discrete II, Automata theory, and Crypto, are familiar with LaTeX.

2 Basics

This is a line of text.

Note that LaTeX will figure out most of the formatting and spacing issues including the spacing after a period or semicolon.

Here's how you do bold, underline, and italics: **hello world**, hello world, *hello world*.

LaTeX code

```
This is a line of text.

Note that \LaTeX\ will figure out most of the formatting and spacing issues
including the spacing after a period or semicolon.
%I have configured it so that a blank line will begin a new paragraph
%with a blank line separating paragraphs.

Here's how you do bold, underline, and italics:
\textbf{hello world},
\underline{hello world},
\textit{hello world}.
```

3 Bullet points: Enumeration

Bullet points are useful:

- hello
- world

You can number them if you like:

(a) do the laundry
(b) goto 1

There's automatic numbering:

1. do the laundry
2. goto 1

You can remove item spacing:

1. do the laundry
2. goto 1

LaTeX code

```
Bullet points are useful:
\begin{enumerate}
\item[$\bullet$] hello
\item[$\bullet$] world
\end{enumerate}

You can number them if you like:
\begin{enumerate}
\item[(a)] do the laundry
\item[(b)] goto 1
\end{enumerate}

There's automatic numbering:
\begin{enumerate}
\item do the laundry
\item goto 1
\end{enumerate}

You can remove item spacing:
\begin{enumerate}[nosep]
\item do the laundry
\item goto 1
```

```
\end{enumerate}
```

4 Basic math

Here's a bit of math: $x = 42$. Note that the math notation is within the text.

Here's some math that is not within the text, but is centered in a paragraph of it's own:

$$x = 42 = 6 \cdot 7 = 6 \times 7 = 41 + 1 = 43 - 1 = 42/1$$

Here's the not equal sign: $3.14159 \neq 3$.

Here are some inequality symbols: $1 \leq 2$, $1 < 2$, $1 \geq 0$, $1 > 0$.

Here's a fraction within the text $\frac{1}{2}$ and another:

$$\frac{1}{1-x}$$

You can stack fractions of course:

$$1 + \frac{1}{1 + \frac{1}{1+\frac{1}{1}}}$$

By the way, note that some of the fractions have smaller fontsize. Whenever you want to force tex to use the usual fontsize you use `\displaystyle` like this:

$$1 + \frac{1}{1 + \frac{1}{1+\frac{1}{1}}}$$

and this

$$1 + \frac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1}}}$$

LaTeX code

```
Here's a bit of math: $x = 42$.
Note that the math notation is within the text.


Here's some math that is not within the
text, but is centered in a paragraph of it's own:
\[
x = 42 = 6 \cdot 7 = 6 \times 7 = 41 + 1 = 43 - 1 = 42/1
```

```
\]

Here's the not equal sign:
$3.14159 \neq 3$.

Here are some inequality symbols:
$ 1 \leq 2$, $1 < 2$, $1 \geq 0$, $1 > 0$.

Here's a fraction within the text $\frac{1}{2}$ and another:
\[
\frac{1}{1 - x}
\]
You can stack fractions of course:
\[
1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1}}}
\]

By the way, note that some of the fractions have smaller fontsize.
Whenever you want to force tex to use the usual fontsize you use
\verb!\displaystyle! like this:
\[
1 + \frac{1}{1 + \frac{\displaystyle 1}{1 + \frac{1}{1}}}
\]
and this
\[
1 + \frac{1}{1 + \frac{\displaystyle 1}{\displaystyle 1 + \frac{1}{1}}}
\]
```

5 Spacing; Text in Math

The spaces are auto computed for you

$$x = x + 1$$

If you like, you can also insert your own spacing like this:

$$x = x + \, 1$$

or this

$$x = x + \; 1$$

or even this:

$$x = x + \qquad 1$$

but you probably shouldn't.

If you display text in math mode it looks weird:

$$x = x + 1 doesnothaveasolution$$

You need to force text mode like this:

$$x = x + 1 \text{does not have a solution}$$

and probably add some spacing:

$$x = x + 1 \text{ does not have a solution}$$

LaTeX code

```
The spaces are auto computed for you
\[
x = x + 1
\]
If you like, you can also insert your own spacing like this:
\[
x = x + \, 1
\]
or this
\[
x = x + \;\, 1
\]
or even this:
```

```
\[
x = x + \hspace{1 cm} 1
\]
but you probably shouldn't.


If you display text in math mode it looks weird:
\[
x = x + 1 does not have a solution
\]
You need to force text mode like this:
\[
x = x + 1 \text{does not have a solution}
\]
and probably add some spacing:
\[
x = x + 1 \hspace{0.2cm} \text{does not have a solution}
\]
```

6 Dots

Sometimes you use ellipsis (basically three dots) for a list of things. There are two types, one for list of values in an expression where the dots are in the middle of the line:

$$1 + 2 + \cdots + n$$

and just a listing of values where the dots are at the bottom of the line:

$$1, 2, \ldots, n$$

There's also the vertical and diagonal dots: $\vdots$, $\ddots$.

LaTeX code

```
Sometimes you use ellipsis (basically three dots) for a list of things.
There are two types, one for list of values in an expression where the
dots are in the middle of the line:
\[
1 + 2 + \cdots + n
\]
and just a listing of values where the dots are at the bottom of the line:
\[
1, 2, \ldots, n
\]

There's also the vertical and diagonal dots: $\vdots$, $\ddots$.
```

7 Subscript and Superscript; Radical Sign

Here's the subscript and superscript: x_5, x^5.

You can subscript or superscript more than once: x_{y_z}, x^{4^5}.

You can of course make things really hairy: $x_{5+y-z}^{a_2+b-3^{4^5}}$.

While on the subject of superscript, here's the radical sign:

$$\sqrt{1}$$

You can nest them:

$$\begin{aligned}&\sqrt{1} \\ &\sqrt{1 + \sqrt{1}} \\ &\sqrt{1 + \sqrt{1 + \sqrt{1}}}\end{aligned}$$

Of course you might want the cube root too: $\sqrt[3]{1}$. Or the 4–th root: $\sqrt[4]{1}$. Or even $\sqrt[x+y^2]{1}$.

LaTeX code

```
Here's the subscript and superscript:
$x_5$, $x^5$.

You can subscript or superscript more than once: $x_{y_z}$,
$x^{4^5}$.

You can of course make things really hairy:
$x_{5 + y - z}^{a_2 + b - 3^{4^5}}$.

While on the subject of superscript, here's the radical sign:
\[
\sqrt{1}
\]
You can nest them:
\begin{align*}
&\sqrt{1} \\
&\sqrt{1 + \sqrt{1}} \\
&\sqrt{1 + \sqrt{1 + \sqrt{1}}}
\end{align*}

Of course you might want the cube root too: $\sqrt[3]{1}$.
```

```
Or the 4--th root: $\sqrt[4]{1}$. Or even $\sqrt[x + y^2]{1}$.
```

8 Summation; Infinity

Here's infinity ∞ and the summation notation:

$$\sum_{n=0}^{\infty} a_n$$

and another:

$$\sum_{n=1,2,3} n^n$$

and another:

$$42 = \sum_{n=5}^{100-m^2} \frac{\sum_{i=0}^{\infty} f(i)^n}{a_n}$$

LaTeX code

```
Here's infinity $\infty$ and the summation notation:
\[
\sum_{n=0}^{\infty} a_n
\]
and another:
\[
\sum_{n = 1, 2, 3} n^n
\]
and another:
\[
42 = \sum_{n=5}^{100 - m^2} \frac{\sum_{i=0}^\infty f(i)^n}{a_n}
\]
```

9 Parentheses and Brackets

Sometimes you need to change the size of your parentheses:

$$\bigl(x + 1 \bigr) + \bigl(1 + x^2 \bigr) + \biggl(\frac{1}{1 - x} \biggr)$$

The same idea works for brackets:

$$\bigl[x + 1 \bigr] + \bigl[1 + x^2 \bigr] + \biggl[\frac{1}{1 - x} \biggr]$$

You can also use this version that auto-computes approximate sizes for you:

$$\left(x + 1 \right) + \left(1 + x^2 \right) + \left(\frac{1}{1 - x} \right)$$

However the `\left` must have a matching `\right`.

LaTeX code

```
Sometimes you need to change the size of your parentheses:
\[
(x + 1)
+
\bigl( 1 + x^2 \bigr)
+
\biggl( \frac{1}{1 - x} \biggr)
\]
The same idea works for brackets:
\[
[x + 1]
+
\bigl[ 1 + x^2 \bigr]
+
\biggl[ \frac{1}{1 - x} \biggr]
\]

You can also use this version that auto-computes approximate
sizes for you:
\[
\left( x + 1 \right)
+
\left( 1 + x^2 \right)
+
\left( \frac{1}{1 - x} \right)
\]
```

```
However the \verb!\left! must have a matching \verb!\right!.
```

10 Greek

Here are some greek symbols: $\alpha, \beta, \gamma, \delta$, etc. Some of them have uppercase version: Γ, Δ, Φ, etc. Here's one that's common:

$$\pi = 3.14159...$$

To find the tex command for a greek symbol, first you do the obvious, and if that does not work, you google. For instance go ahead and google "latex alpha" or "latex greek symbol alpha" or even "latex greek symbols".

LaTeX code

```
Here are some greek symbols: $\alpha, \beta, \gamma, \delta$, etc.
Some of them have uppercase version:
$\Gamma, \Delta, \Phi$, etc.
Here's one that's common:
\[
\pi = 3.14159...
\]

To find the tex command for a greek symbol, first you do the obvious,
and if that does not work, you google.
For instance go ahead and google
\lq\lq latex alpha"
or
\lq\lq latex greek symbol alpha"
or even \lq\lq latex greek symbols".
```

11 Aligning and Numbering Things

For a long sequence of computations, you can align like this:

$$\begin{aligned} x &= 1+2+3+4 = 3+3+4 \\ &= 6+4 \\ &= 10 \end{aligned}$$

using `&` for alignment and `\\` for newline. Here's another example:

$$\begin{aligned} x &= 1 \\ y &= 22 \end{aligned}$$

If you need to explain your work you do this:

$$\begin{aligned} x &= 1+2+3+4 = 3+3+4 && \text{by Formula 1324} \\ &= 6+4 && \text{by Formula 12455} \\ &= 10 && \text{by Black Magic Trick \#5} \end{aligned}$$

You might need to use two lines for a long expression:

$$\begin{aligned} x &= 1+2+3+4 \\ &\quad +5+6 \end{aligned}$$

Sometimes you need to label equations so that you don't have to say "according to the formula 5 lines above". Here's how you do it:

$$1+1=2 \tag{42}$$

That can also be used in the aligned environment together with explanatory texts:

$$\begin{aligned} x &= 2+30 && \text{by Formula (1)} \\ &= 32 && (2) \\ y &= x+10 \\ &= 32+10 && \text{by (2)} \\ &= 42 && (3) \end{aligned}$$

Here's an example for two alignments:

$$\begin{aligned} x &\equiv 2+30 && \pmod{26} \\ &\equiv 32 && \pmod{26} \end{aligned}$$

But this looks terrible. The following is better ...

Alignments with 3 visible columns:

$$\begin{aligned} x &\equiv 1 + 1 && \pmod{26} \\ &\equiv 24 && \pmod{26} \end{aligned}$$

The argument (see the LaTeX) is 2 because there are *two pairs* of columns. The `&&` has an empty column in the middle. The above is actually like this:

$$LRLR$$

where L is left–justified and R is right–justified. For instance, look at the LaTeX for this:

$$\begin{array}{rlrlrl} \text{[right]} & \text{[left]} & & \text{[left]} & \text{[right]} & \text{[right]} \\ \text{[xxxxxxxxxx]} & \text{[xxxxxxxxxx]} & \text{[xxxxxxxxxx]} & \text{[xxxxxxxxxx]} & \text{[xxxxxxxxxx]} \end{array}$$

Example with 6 columns:

$$\begin{aligned} x &\equiv 1 + 1 && \pmod{26} && \text{short} \\ &\equiv 0 && \pmod{2} && \text{very long} \end{aligned}$$

Example with 8 columns:

$$\begin{aligned} x &\equiv 1 + 1 && \pmod{26} && \text{short} && \text{short} \\ &\equiv 0 && \pmod{2} && \text{very long} && \text{very long} \end{aligned}$$

LaTeX code

```
For a long sequence of computations, you can align like this:
\begin{align*}
x &= 1 + 2 + 3 + 4 = 3 + 3 + 4 \\
  &= 6 + 4 \\
  &= 10
\end{align*}
using $\verb!&!$ for alignment and $\verb!\\!$ for newline.
Here's another example:
\begin{align*}
x &= 1 \\
y &= 22
```

```
\end{align*}
If you need to explain your work you do this:
\begin{align*}
x &= 1 + 2 + 3 + 4 = 3 + 3 + 4 & & \text{by Formula 1324}\\
  &= 6 + 4                     & & \text{by Formula 12455} \\
  &= 10                        & & \text{by Black Magic Trick \#5}
\end{align*}
You might need to use two lines for a long expression:
\begin{align*}
x &= 1 + 2 + 3 + 4 \\
  & \hspace{0.5cm} + 5 + 6
\end{align*}

Sometimes you need to label equations so that you don't have to say
\lq\lq according to the formula 5 lines above''.
Here's how you do it:
\[
1 + 1 = 2 \tag{42}
\]
That can also be used in the aligned environment together with explanatory
texts:
\begin{align*}
 x &= 2 + 30     & & \text{by Formula (1)} \\
   &= 32         \tag{2} \\
 y &= x + 10 \\
   &= 32 + 10    & & \text{by (2)} \\
 &= 42  \tag{3}
\end{align*}
Here's an example for two alignments:
\begin{align*}
 x &\equiv 2 + 30     &\pmod{26} \\
   &\equiv 32         &\pmod{26}
\end{align*}
But this looks terrible.
The following is better ...

Alignments with 3 visible columns:
\begin{alignat*}{2}
 x &\equiv 1 + 1 &&\pmod{26} \\
   &\equiv 24    &&\pmod{26}
\end{alignat*}
The argument (see the \LaTeX) is 2 because there are \textit{two pairs} of columns.
The \verb!&&! has an empty column in the middle.
The above is actually like this:
\begin{alignat*}{2}
 L & R & L & R \\
\end{alignat*}
where $L$ is left--justified and $R$ is right--justified.
For instance, look at the \LaTeX\ for this:
\begin{alignat*}{4}
 \text{[right]}     & \text{[left]}      && \text{[left]}      & \text{[right]}     && \text{[right
```

```
 \text{[xxxxxxxxxx]} & \text{[xxxxxxxxxx]} && \text{[xxxxxxxxxx]} & \text{[xxxxxxxxxx]} && \text{[xx
\end{alignat*}

Example with 6 columns:
\begin{alignat*}{3}
 x &\equiv 1 + 1 &&\pmod{26} &&\hspace{1cm} \text{short} \\
   &\equiv 0     &&\pmod{2}  &&\hspace{1cm} \text{very long}
\end{alignat*}

Example with 8 columns:
\begin{alignat*}{4}
 x &\equiv 1 + 1 &&\pmod{26} &&\ \ \ \text{short}     &&\ \ \ \text{short} \\
   &\equiv 0     &&\pmod{2}  &&\ \ \ \text{very long} &&\ \ \ \text{very long}
\end{alignat*}
```

12 Sets

Here are some standard sets:

$$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$$

To make it easier for you, I've defined the following commands to do the same thing:

$$\N, \Z, \R, \C$$

And we all need the empty set: $\emptyset$.

Here's a set of your own by listing all the elements:

$$X = \{1, 2, 3\}$$

and here's another

$$Y = \{x \in \mathbb{Z} \mid 0 < x \leq 3\}$$

Here's membership: $1 \in \mathbb{N}$. And non-membership: $-1 \notin \mathbb{N}$.

Here's the subset-or-equal: $\mathbb{N} \subseteq \mathbb{Z}$ and the opposite $\mathbb{Z} \nsubseteq \mathbb{N}$.

Here's union and intersection: $\mathbb{N} \cup \mathbb{Z} = \mathbb{Z}$, $\mathbb{N} \cap \mathbb{Z} = \mathbb{N}$.

You might need the bar notation $\overline{X}$ for complement, closure, etc.

LaTeX code

```
Here are some standard sets:
\[
\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}
\]

To make it easier for you, I've defined the following commands to do the
same thing:
\[
\N, \Z, \R, \C
\]

And we all need the empty set: $\emptyset$.

Here's a set of your own by listing all the elements:
\[
X = \{ 1, 2, 3 \}
```

```
\]
and here's another
\[
Y = \{ x \in \Z \mid 0 < x \leq 3 \}
\]

Here's membership: $1 \in \N$.
And non-membership: $-1 \notin \N$.

Here's the subset-or-equal: $\N \subseteq \Z$
and the opposite $\Z \nsubseteq \N$.

Here's union and intersection:  $\N \cup \Z = \Z$, $\N \cap \Z = \N$.

You might need the bar notation $\overline{X}$ for complement, closure, etc.
```

13 Functions

Here's the function notation: $f : X \to Y$. You can define a function like this:

$$\begin{aligned} f &: \mathbb{R} \to \mathbb{R} \\ f(x) &= x^2 \end{aligned}$$

or this

$$\begin{aligned} f &: \mathbb{R} \to \mathbb{R} \\ x &\mapsto x^2 \end{aligned}$$

And for a piecewise definition you can do this:

$$f(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

Here are the trig functions:

$$\sin x, \cos x, \tan x$$

etc.

There are times when instead of $f(2) = 6$, you might want to say this:

$$2 \overset{f}{\mapsto} 6$$

The arrow looks too short. In this case, you should do this instead:

$$2 \overset{f}{\longmapsto} 6$$

This method of putting something over something else works in general. For instance

$$2 \overset{?!?}{>} 6$$

For extremely long arrows you might want to do this:

$$X \xrightarrow{\text{do not apply this function}} Y$$

LaTeX code

```
Here's the function notation: $f : X \rightarrow Y$.
You can define a function like this:
\begin{align*}
f : \R \rightarrow \R \\
f(x) = x^2
\end{align*}
or this
\begin{align*}
f : \R &\rightarrow \R \\
x &\mapsto x^2
\end{align*}

And for a piecewise definition you can do this:
\[
f(x) =
\begin{cases}
-1 & \text{ if } x < 0 \\
1  & \text{ if } x \geq 0
\end{cases}
\]

Here are the trig functions:
\[
\sin x, \cos x, \tan x
\]
etc.

There are times when instead of $f(2) = 6$, you might want to say this:
\[
2 \overset{f}{\mapsto} 6
\]
The arrow looks too short.
In this case, you should do this instead:
\[
2 \overset{f}{\longmapsto} 6
\]
This method of putting something over something else works in general.
For instance
\[
2 \overset{\text{?!?}}{>} 6
\]

For extremely long arrows you might want to do this:
\[
X \xrightarrow{\text{do not apply this function}} Y
\]
```

14 Calculus

Here's how you write limits:

$$\lim_{x\to 0}\frac{1}{|x|}$$

and of course there are the left- and right-limits:

$$\lim_{x\to 0^-}\frac{1}{x},\quad \lim_{x\to 0^+}\frac{1}{x},$$

Here's the derivative and integral:

$$\frac{d}{dx}x^3 \qquad\qquad \int_0^\infty \frac{1}{1+x^2}\,dx$$

and here are some multiple integrals:

$$\iint xy\,dxdy \qquad\qquad \iiint xy\,dxdydz$$

Frequently you need the long vertical line for limits of integration:

$$\int_2^5 x^3\,dx = \frac{1}{4}x^4\Big|_{x=2}^{x=5}$$

or some books might write

$$\int_2^5 x^3\,dx = \left[\frac{1}{4}x^4\right]_{x=2}^{x=5}$$

You might need this:

$$\oint_C f\ dz$$

Some books also write the vertical line for the value of the derivative at an x value:

$$\frac{d}{dx}x^3\Big|_{x=6} = 3x^2\Big|_{x=6} = 3\cdot 6^2$$

The ∂ allows you to write partial derivatives:

$$\frac{\partial}{\partial y}(x^y+\sin y)$$

LaTeX code

```
Here's how you write limits:
\[
\lim_{x \rightarrow 0} \frac{1}{|x|}
\]
and of course there are the left- and right-limits:
\[
\lim_{x \rightarrow 0^-} \frac{1}{x},
\,\,\,
\lim_{x \rightarrow 0^+} \frac{1}{x},
\]

Here's the derivative and integral:
\[
\frac{d}{dx} x^3
\hspace{1in}
\int_{0}^{\infty} \frac{1}{1 + x^2} \, dx
\]
and here are some multiple integrals:
\[
\iint xy \, dx dy
\hspace{1in}
\iiint xy \, dx dy dz
\]

Frequently you need the long vertical line for limits of integration:
\[
\int_{2}^{5} x^3 \, dx
=
\frac{1}{4} x^4 \, \biggr|_{x=2}^{x=5}
\]
or some books might write
\[
\int_{2}^{5} x^3 \, dx =
\left[
\frac{1}{4} x^4
\right]_{x=2}^{x=5}
\]
You might need this:
\[
\oint_C f \ dz
\]


Some books also write the vertical line for the value of the derivative
at an $x$ value:
\[
\frac{d}{dx} x^3 \biggr|_{x = 6}
```

```
= 3x^2 \biggr|_{x = 6}
= 3 \cdot 6^2
\]

The $\partial$ allows you to write partial derivatives:
\[
\frac{\partial}{\partial y} (x^y + \sin y)
\]
```

15 Logic

Here's "for all" and "there exists":

$$\forall x \in \mathbb{R},\ \ x^2 \geq 0 \qquad\qquad \exists x \in \mathbb{R} \text{ such that } x + 1 = 5$$

And of course we need logical and, or, and not:

$$P \wedge Q \qquad P \vee Q \qquad \neg P$$

and the various implications:

$$P \to Q \qquad P \leftarrow Q \qquad P \leftrightarrow Q$$

and their tautologies

$$P \implies Q \qquad P \impliedby Q \qquad P \iff Q$$

You might need this:

$$P \equiv Q$$

Some people use"therefore" as in:

$$\therefore\ \ x > 0$$

You can therefore do this:

$$\begin{aligned} x &> 0 \\ \therefore \quad x + 1 &> 1 \\ \therefore \quad x + 1 + 2 &> 1 + 2 \end{aligned}$$

or this:

$$\begin{aligned} x &> 0 \\ \therefore \quad x + 1 &> 1 \\ \therefore \quad x + 1 + 2 &> 1 + 2 \end{aligned}$$

LaTeX code

```
Here's \lq\lq for all'' and \lq\lq there exists'':
\[
\forall x \in \R, \,\,\,\, x^2 \geq 0
\hspace{1in}
```

```
\exists x \in \R \text{ such that } x + 1 = 5
\]
And of course we need logical and, or, and not:
\[
P \land Q \hspace{1cm}
P \lor Q \hspace{1cm}
\lnot P
\]
and the various implications:
\[
P \rightarrow Q \hspace{1cm}
P \leftarrow Q \hspace{1cm}
P \leftrightarrow Q
\]
and their tautologies
\[
P \implies Q \hspace{1cm}
P \impliedby Q \hspace{1cm}
P \iff Q
\]
You might need this:
\[
P \equiv Q
\]

Some people use\lq\lq therefore'' as in:
\[
\therefore \,\,\, x > 0
\]
You can therefore do this:
\begin{align*}
 x &> 0 \\
\therefore \hspace{0.5cm} x + 1 &> 1 \\
\therefore \hspace{0.5cm} x + 1 + 2 &> 1 + 2
\end{align*}
or this:
\begin{alignat*}{3}
&           & x                       &> 0 \\
&\therefore &\hspace{0.5cm} x + 1     &> 1 \\
&\therefore &\hspace{0.5cm} x + 1 + 2 &> 1 + 2
\end{alignat*}
```

16 Linear algebra, vectors, matrices

Here's a vector written in different styles:

$$\overrightarrow{i} \qquad \vec{i} \qquad \mathbf{i} \qquad \hat{i}$$

And for specifying components:

$$\langle 2, 4 \rangle \qquad \langle 2, 4, -1 \rangle \qquad \langle x, y, z, t \rangle$$

And length: $|v|$ or $\|v\|$.

You can write your matrix in different ways:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

You can use dots if there are too many values:

$$A = \begin{pmatrix} 1 & 2 & \cdots & 10 \\ 11 & 12 & \cdots & 20 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 2 & \cdots & 10 \\ 21 & 22 & \cdots & 20 \\ \vdots & \vdots & \ddots & \vdots \\ 91 & 92 & \cdots & 100 \end{pmatrix}$$

Here are different ways to write the determinant:

$$\det B \qquad \begin{vmatrix} \alpha & \beta \\ \gamma & \delta \end{vmatrix} = \alpha\delta - \beta\gamma \qquad \begin{Vmatrix} \alpha & \beta \\ \gamma & \delta \end{Vmatrix}$$

If you need left justification, right justification, centering of entries in a matrix, make sure you add `\usepackage{mathtools}` and do this:

$$\begin{pmatrix} 1 & 2 \\ 1111 & -2222 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ 1111 & -2222 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ 1111 & -2222 \end{pmatrix}$$

LaTeX code

```
Here's a vector written in different styles:
\[
\overrightarrow{i} \hspace{1cm}
```

```
\vec{i} \hspace{1cm}
{\mathbf i}  \hspace{1cm}
\hat{i}
\]
And for specifying components:
\[
\langle 2, 4 \rangle
\hspace{1cm}
\langle 2, 4, -1 \rangle
\hspace{1cm}
\langle x, y, z, t \rangle
\]
And length: $|v|$ or $\| v \|$.

You can write your matrix in different ways:
\[
A =
\begin{pmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{pmatrix}
\hspace{1in}
B =
\begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6
\end{bmatrix}
\]

You can use dots if there are too many values:
\[
A =
\begin{pmatrix}
1 & 2 & \cdots & 10 \\
11 & 12 & \cdots & 20
\end{pmatrix}
\hspace{1in}
B =
\begin{pmatrix}
1 & 2 & \cdots & 10 \\
21 & 22 & \cdots & 20 \\
\vdots &  \vdots & \ddots & \vdots \\
91 & 92 & \cdots & 100 \\
\end{pmatrix}
\]

Here are different ways to write the determinant:
\[
\det B
\hspace{1in}
\begin{vmatrix}
```

```
\alpha & \beta \\
\gamma & \delta
\end{vmatrix}
= \alpha \delta - \beta\gamma
\hspace{1in}
\begin{Vmatrix}
\alpha & \beta \\
\gamma & \delta
\end{Vmatrix}
\]

If you need left justification, right justification, centering of
entries in a matrix,
make sure you add \verb!\usepackage{mathtools}! and
do this:
\[
\begin{pmatrix*}[l]
   1 & 2  \\
1111 & -2222
\end{pmatrix*},
\begin{pmatrix*}[c]
   1 & 2  \\
1111 & -2222
\end{pmatrix*},
\begin{pmatrix*}[r]
   1 & 2  \\
1111 & -2222
\end{pmatrix*}
\]
```

17 Number theory

Here's divisibility and non-divisibility:

$$3 \mid 6 \qquad\qquad 5 \nmid 6$$

the gcd and congruence

$$\gcd(3,6) \qquad\qquad x \equiv 3 \pmod{43}$$

You might need the Euler ϕ-function.

$$\phi(5) = 4$$

Here's the binomial coefficient or the n–choose–r:

$$\binom{n}{r}$$

Here's the floor and ceiling:

$$\lfloor 1.5 \rfloor = 1 \qquad\qquad \lceil 1.5 \rceil = 2$$

And for the floor and ceiling of huge expressions you can do this

$$\left\lfloor \frac{1}{1-x} \right\rfloor \qquad\qquad \left\lceil \int_0^\infty \frac{1}{1+x^2}\, dx \right\rceil$$

I've defined two shorthand macros `\floor` and `\ceiling` to make it easier for you:

$$\left\lfloor \frac{1}{1-x} \right\rfloor \qquad\qquad \left\lceil \int_0^\infty \frac{1}{1+x^2}\, dx \right\rceil$$

LaTeX code

```
Here's divisibility and non-divisibility:
\[
3 \mid 6    \hspace{1in}    5 \nmid 6
\]
the gcd and congruence
\[
```

```
\gcd(3, 6)    \hspace{1in}    x \equiv 3 \pmod{43}
\]

You might need the Euler $\phi$-function.
\[
\phi(5) = 4
\]

Here's the binomial coefficient or the $n$--choose--$r$:
\[
\binom{n}{r}
\]

Here's the floor and ceiling:
\[
\lfloor 1.5 \rfloor = 1    \hspace{1in}    \lceil 1.5 \rceil = 2
\]
And for the floor and ceiling of huge expressions you can do this
\[
\left\lfloor \frac{1}{1 - x} \right\rfloor
\hspace{1in}
\left\lceil \int_0^\infty \frac{1}{1 + x^2} \ dx \right\rceil
\]

I've defined two shorthand macros \verb!\floor! and \verb!\ceiling!
to make it easier for you:
\[
\floor{\frac{1}{1 - x}}
\hspace{1in}
\ceiling{\int_0^\infty \frac{1}{1 + x^2} \ dx}
\]
```

18 Automata and languages

Here's a grammar:

$$G : \begin{cases} S \to A \mid bA \mid Ba \mid \epsilon \\ A \to aA \mid \epsilon \\ B \to Bb \mid S\epsilon \end{cases}$$

Here's a derivation:

$$S \implies bA \implies baA \implies ba$$

or

$$\begin{aligned} S &\implies Ba \\ &\implies Bba \\ &\implies ba \end{aligned}$$

Here's a computation of instantaneous descriptions:

$$(q_0, abaa) \vdash (q_1, baa)$$

Here are some complexity classes, problems, etc.:

$$\mathrm{P}, \mathrm{NP}, \mathrm{DTime}, \mathrm{NTime}, \mathrm{HaltingProblem},$$

LaTeX code

```
Here's a grammar:
\[
G :
\begin{cases}
  S \rightarrow A \mid bA \mid Ba \mid \epsilon \\
  A \rightarrow aA \mid \epsilon \\
  B \rightarrow Bb \mid S \epsilon
\end{cases}
\]
Here's a derivation:
\[
S \implies bA \implies baA \implies ba
\]
or
\begin{align*}
  S
  &\implies Ba \\
  &\implies Bba \\
```

```
  &\implies ba
\end{align*}
Here's a computation of instantaneous descriptions:
\[
(q_0, abaa) \vdash (q_1, baa)
\]
Here are some complexity classes, problems, etc.:
\[
\textsc{P},
\textsc{NP},
\textsc{DTime},
\textsc{NTime},
\textsc{HaltingProblem},
\]
```

19 Definitions, Theorems, Lemmas, etc.

Here's how you do definition:

Definition 19.1. An integer is said to be good if it is 42.

And here's how to do a lemma:

Lemma 19.1. *If an integer is good, then it is divisible by 6.*

and a theorem with proof:

Theorem 19.1. *If n is good, then $n/6$ is an integer greater than 1.*

Proof. Let n be good. We have $n/6 = 42/6 = 7$. Since 7 is greater than 1, we conclude that $n/6 = 7 > 1$. □

And a corollary with proof:

Corollary 19.1. *If n is good, then $n/3$ is greater than 0.*

Proof. If n is good, then by Theorem 1, $n/6 > 1$. Since $n > 0$, $n/3 > n/6$. Hence

$$\frac{n}{3} > \frac{n}{6} > 1$$

□

LaTeX code

```
Here's how you do definition:

\begin{defn}
An integer is said to be good if it is 42.
\end{defn}

And here's how to do a lemma:

\begin{lem}
If an integer is good, then it is divisible by 6.
\end{lem}
```

```
and a theorem with proof:

\begin{thm}
If $n$ is good, then $n / 6$ is an integer greater than 1.
\end{thm}

\begin{proof}
Let $n$ be good. We have $n / 6 = 42 / 6 = 7$. Since $7$ is greater than 1,
we conclude that $n / 6 = 7 > 1$.
\end{proof}

And a corollary with proof:

\begin{cor}
If $n$ is good, then $n / 3$ is greater than 0.
\end{cor}

\begin{proof}
If $n$ is good, then by Theorem 1, $n / 6 > 1$. Since $n > 0$, $n/3 > n/6$.
Hence
\[
\frac{n}{3} > \frac{n}{6} > 1
\]
\end{proof}
```

20 Table

Here's how you do tables:

x	x^2
0	1
1	1
2	4
3	9

Compare it with this:

x	x^2
0	1
1	1
2	4
3	9

Here's another:

this is left flushed	this is centered	this is right flushed
a	b	c

You can use the table to do this as well:

$$\begin{array}{r} p \\ q \\ \hline \therefore r \end{array}$$

You can do pretty much any table you like:

Trip Planning to Middle Earth	
Magic wand	\$2.50
Magic spell	\$20.00
Axe	\$15.50
Water	\$10.00
Bread	\$10.00
TOTAL	\$58.00

There are times when you want to draw a horizontal line only for certain columns:

		1	1		(carries)
	3	3	2	5	
+	6	4	8	7	
	9	8	1	2	

Note that in `longtable`, you use \\ (as always) for newline. If you want to prevent pagebreak (because `longtable` allows pagebreaks within a table), you use *.

LaTeX code

```
Here's how you do tables:

\begin{longtable}{|c|c|} \hline
$x$   & $x^2$ \\ \hline
0     & 1     \\ \hline
1     & 1     \\ \hline
2     & 4     \\ \hline
3     & 9     \\ \hline
\end{longtable}

Compare it with this:

\begin{longtable}{|c|c|} \hline
$x$   & $x^2$ \\ \hline
0     & 1     \\
1     & 1     \\
2     & 4     \\
3     & 9     \\ \hline
\end{longtable}

Here's another:
\begin{longtable}{|l|c|r|} \hline
this is left flushed & this is centered & this is right flushed \\ \hline
a & b & c \\ \hline
\end{longtable}

You can use the table to do this as well:
\begin{longtable}{r}
$p$ \\
$q$ \\ \hline
$\therefore$ $r$
\end{longtable}

You can do pretty much any table you like:
\begin{longtable}{|l r|} \hline
```

```
\multicolumn{2}{|c|}{Trip Planning to Middle Earth}\\
\hline \hline
Magic wand  & \$2.50  \\
Magic spell & \$20.00 \\
Axe         & \$15.50 \\ \hline
Water       & \$10.00 \\
Bread       & \$10.00 \\ \hline \hline
TOTAL       & \$58.00 \\ \hline
\end{longtable}

There are times when you want to draw a horizontal line only
for certain columns:

\begin{longtable}{cccccc}
  &   & 1 & 1 &   & (carries) \\
  & 3 & 3 & 2 & 5 & \\
+ & 6 & 4 & 8 & 7 & \\
\cline{1-5}
  & 9 & 8 & 1 & 2 &  \\
\cline{1-5}
\end{longtable}


Note that in \verb!longtable!, you use \verb!\\! (as always) for newline.
If you want to prevent pagebreak (because \verb!longtable! allows
pagebreaks within a table), you use \verb!\\*!.
```

21 Multiple files

If you have a huge tex source file, you might want to break it up into several files (for instance one tex file per section).

For instance say you have a tex file called `gold.tex`. You take out a chunk of tex code and put that into another file called `howtomakegoldfrommud.tex`. In your main tex file `gold.tex`, to include the contents of `howtomakegoldfrommud.tex` you do this:

```
\input{howtomakegoldfrommud.tex}
```

LaTeX code

```
If you have a huge tex source file, you might want to break it up into
several files (for instance one tex file per section).

For instance say you have a tex file called \verb!gold.tex!.
You take out a chunk of tex code and put that into another file called
\verb!howtomakegoldfrommud.tex!.
In your main tex file \verb!gold.tex!, to include the contents of
\verb!howtomakegoldfrommud.tex! you do this:
\begin{console}
\input{howtomakegoldfrommud.tex}
\end{console}
```

22 Comments

You can't see the hello world on the next line because I've commented it out:

It's done using the line comment:
%hello world

A block comment looks like this:
\begin{comment}
hello world
\end{comment}

Comments can be useful for debugging, If you have an error that you can't find, just continually comment out tex code and recompile until tex stops complaining. The last thing commented out must be the error.

LaTeX code

```
You can't see the hello world on the next line because I've commented it out:
%hello world

It's done using the line comment: \\
\verb!%hello world!

A block comment looks like this:
\\
\verb!\begin{comment}! \\
\verb!hello world! \\
\verb!\end{comment}!

%\begin{console}
%\begin{comment}
%hello world
%hello world
%\end{comment}
%\end{console}

Comments can be useful for debugging,
If you have an error that you can't find, just continually comment out tex
code and recompile until tex stops complaining.
The last thing commented out must be the error.
```

23 Your own Commands

You can define your own command. For instance after this tex code

```
\newcommand\ddx{\frac{d}{dx}}
```

you can do this

```
\[
\ddx x^3 = 3x^2
\]
```

and get

$$\frac{d}{dx}x^3 = 3x^2$$

Here's another example.

```
\newcommand\infsum{\sum^\infty_{n=0}}
\[
\infsum \frac{1}{2^n} x^n = \infsum \left( \frac{x}{2} \right)^n
\]
```

produces

$$\sum_{n=0}^{\infty} \frac{1}{2^n}x^n = \sum_{n=0}^{\infty} \left(\frac{x}{2}\right)^n$$

LaTeX code

```
You can define your own command.
For instance after this tex code
\begin{console}
\newcommand\ddx{\frac{d}{dx}}
\end{console}
you can do this
\begin{console}
\[
\ddx x^3 = 3x^2
\]
\end{console}
and get
\newcommand\ddx{\frac{d}{dx}}
\[
\ddx x^3 = 3x^2
\]
Here's another example.
\begin{console}
\newcommand\infsum{\sum^\infty_{n=0}}
\[
\infsum \frac{1}{2^n} x^n = \infsum \left( \frac{x}{2} \right)^n
```

```
\]
\end{console}
produces
\newcommand\infsum{\sum^\infty_{n=0}}
\[
\infsum \frac{1}{2^n} x^n = \infsum \left( \frac{x}{2} \right)^n
\]
```

24 Page Control

Page control is easy. To force a new page, just do `\newpage`. I'll do it here ...

... and you see now I'm on the next page.

There are times when you do *not* want a chunk of information to be broken up, with part of it on the bottom of a page and the other half on the top of the next. This happens a lot to me for instance when I need to do some ASCII art. To prevent a new page you can do this:

```
\begin{samepage}
THIS IS AN IMPORTANT FACT AND I DO NOT WANT IT TO SPLIT UP:
IF YOU BOIL 1 POUND OF GRASS WITH 2 POUNDS OF BIRD FEATHERS IN
10 GALLONS OF RAINWATER FOR 200 HOURS,
YOU WILL GET 3.14159 POUNDS OF GOLD.
\end{samepage}
```

LaTeX code

```
Page control is easy. To force a new page, just do \verb!\newpage!.
I'll do it here ... \newpage
... and you see now I'm on the next page.

There are times when you do \textit{not} want a chunk of information to be
broken up, with part of it on the bottom of a page and the other half
on the top of the next.
This happens a lot to me for instance when I need to do some ASCII
art.
To prevent a new page you can do this:
\begin{console}
\begin{samepage}
THIS IS AN IMPORTANT FACT AND I DO NOT WANT IT TO SPLIT UP:
IF YOU BOIL 1 POUND OF GRASS WITH 2 POUNDS OF BIRD FEATHERS IN
10 GALLONS OF RAINWATER FOR 200 HOURS,
YOU WILL GET 3.14159 POUNDS OF GOLD.
\end{samepage}
\end{console}
```

25 Verbatim

If you need to write code (or pseudocode) you can do this: `int x = 42;` or this

```
int x = 42;
```

or this if you want to have a framed border:

```
int x = 42;
```

Note that in all the above cases, there's no formatting done to the contents, i.e., the contents are processed verbatim. This is great because for source code, you want to retain your indentation:

```
if (x > 0)
{
    x = 42;
}
```

Note that because of this, if you try to go into math mode you won't get math notation:

```
$\frac{1}{2} + \int_0^\infty x^{-3} \, dx$
```

You can show the spaces like this:

```
int␣main()
{
␣␣␣␣std::cout␣<<␣"hello␣world"␣<<␣std::endl;
␣␣␣␣return␣0;
}
```

LaTeX code

```
If you need to write code (or pseudocode) you can do this:
\verb!int x = 42;!
or this
\begin{Verbatim}
int x = 42;
\end{Verbatim}
or this if you want to have a framed border:
\begin{Verbatim}[frame=single]
int x = 42;
```

```
\end{Verbatim}
Note that in all the above cases, there's no formatting done to the
contents, i.e., the contents are processed verbatim.
This is great because for source code, you want to retain your indentation:
\begin{Verbatim}[frame=single]
if (x > 0)
{
    x = 42;
}
\end{Verbatim}
Note that because of this,
if you try to go into math mode you won't get math notation:
\begin{Verbatim}[frame=single]
$\frac{1}{2} + \int_0^\infty x^{-3} \, dx$
\end{Verbatim}

You can show the spaces like this:
\begin{Verbatim}[frame=single,showspaces=true]
int main()
{
    std::cout << "hello world" << std::endl;
    return 0;
}
\end{Verbatim}
```

26 Verbatim: Tex Commands

If you *do* want tex to process some part of your contents in the verbatim mode as tex commands, you can do this. I'm going to underline some of the contents in the verbatim mode.

```
1 2 3 4 5 6
```

If you look at the tex code, you will see `commandchars=\\\{\}` which tells tex that `\...{...}` is a tex command, include the case of `\underline{...}`.

Here's another example. I'll color some text red:

```
1 2 3 4 5 6
```

Instead of using `\` and `{` and `}` to denote the presence of tex commands in your verbatim, you can use other characters. This is the case if you have C++ code in your verbatim since `{` and `}` are used in C++ code. The following uses `\` and `@` and `~` to denote tex commands in the verbatim text.

```
#include <iostream>
int main()
{
    std::cout << "hello world" << std::endl;
    return 0;
}
```

LaTeX code

```
If you \textit{do} want tex to process some part of your contents
in the verbatim mode as tex commands, you can do this.
I'm going to underline some of the contents in the verbatim mode.

\begin{Verbatim}[frame=single, commandchars=\\\{\}]
1 2 3 \underline{4 5} 6
\end{Verbatim}

If you look at the tex code, you will see
\verb!commandchars=\\\{\}!
which tells tex that \verb!\...{...}! is a tex command,
include the case of \verb!\underline{...}!.

Here's another example.
```

```
I'll color some text red:
\begin{Verbatim}[frame=single, commandchars=\\\{\}]
1 2 3 \redtext{4 5} 6
\end{Verbatim}

Instead of using \verb!\! and \verb!{! and \verb!}! to denote
the presence of tex commands in your verbatim, you can use other characters.
This is the case if you have C++ code in your verbatim
since \verb!{! and \verb!}! are used in C++ code.
The following uses \verb!\! and \verb!@! and \verb!~! to denote tex
commands in the verbatim text.

\begin{Verbatim}[frame=single, commandchars=\\\@\~]
#include <iostream>
int main()
{
    std::cout << \underline@"hello world"~ << std::endl;
    return 0;
}
\end{Verbatim}
```

27 Verbatim: Source Files

Now I'm going to do the same but reading the contents from a file:

```
#include <iostream>

int main()
{
    std::cout << "hello world" << std::endl;
    return 0;
}
```

You can also add line numbers:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "hello world" << std::endl;
6     return 0;
7 }
```

and also include tex commands as before:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "hello world" << std::endl;
6     return 0;
7 }
```

LaTeX code

```
Now I'm going to do the same but reading the contents from a file:
\VerbatimInput[frame=single]{source_code_example.cpp}

You can also add line numbers:
\VerbatimInput[frame=single, numbers=left]
{source_code_example.cpp}
and also include tex commands as before:
```

```
\VerbatimInput[frame=single, numbers=left, commandchars=+\[\]]
{source_code_example2.cpp}
```

28 Orientation

You can have landscape orientation

- whole document
- a page
- a fragment of a page

Here's how you do landscape for a page. Make sure you use the package `lscape`. The next page is in landscape:

hello world

LaTeX code

```
You can have landscape orientation
\begin{itemize}
\li whole document
\li a page
\li a fragment of a page
\end{itemize}

Here's how you do landscape for a page.
Make sure you use the package \verb!lscape!.
The next page is in landscape:

\begin{landscape}
hello world
\end{landscape}
```

29 Python

If you want to run python code, do this:

```
\documentclass[a4paper,12pt]{article}
\usepackage{python}

\begin{document}
\begin{python}
import math
print(r"\sqrt{2} = ", math.sqrt(2))
\end{python}
\end{document}
```

If your tex file is `main.tex` and the python code has an error, then the error message might be written to `main.py.err`.

Advice: Test the python code outside tex. Make sure it works before using it inside latex.

LaTeX code

```
If you want to run python code, do this:
\begin{console}
\documentclass[a4paper,12pt]{article}
\usepackage{python}

\begin{document}
\begin{python}
import math
print(r"\sqrt{2} = ", math.sqrt(2))
\end{python}
\end{document}
\end{console}

If your tex file is \verb!main.tex!
and the python code has an error,
then the error message
might be written to \verb!main.py.err!.

Advice: Test the python code outside tex.
Make sure it works before using it inside latex.
```

30 mathsc

Use the `mathsc` macro to typeset textsc (sc = small caps) content in math mode. Use this for naming algorithms and defining complexity problems/languages/classes/etc. in automata theory.

Examples:

- $\textsc{Lang}_\Sigma$, $\textsc{Reg}_\Sigma$, $\textsc{CFL}_\Sigma$
- $\textsc{Accept}_{\textsc{TM}}$, $\textsc{NonEmpty}_{\textsc{TM}}$, $\textsc{Halt}_{\textsc{TM}}$, $\textsc{Equal}_{\textsc{TM}}$
- $\textsc{Time}(n^3)$, $\textsc{NTime}(n^3)$
- 3Sat, Clique

LaTeX code

```
Use the \verb!mathsc! macro to typeset textsc (sc = small caps)
content in math mode.
Use this for naming algorithms and defining complexity
problems/languages/classes/etc. in automata
theory.

Examples:
\begin{tightlist}
\li $\mathsc{Lang}_\Sigma$, $\mathsc{Reg}_\Sigma$, $\mathsc{CFL}_\Sigma$
\li $\mathsc{Accept}_\mathsc{TM}$, $\mathsc{NonEmpty}_\mathsc{TM}$,
$\mathsc{Halt}_\mathsc{TM}$, $\mathsc{Equal}_\mathsc{TM}$
\li $\mathsc{Time}(n^3)$, $\mathsc{NTime}(n^3)$
\li $\mathsc{3Sat}$, $\mathsc{Clique}$
\end{tightlist}
```

31 References: theorem

The following uses `\label` and `\ref`.

Referencing a theorem:

Theorem 31.1. *1 + 1 = 2*

Referencing a definition:

Definition 31.1. A **prime** is a positive integer divisible only by 1 or itself.

Look at the next page ...

The above Theorem 31.1 is on page 61. The above Definition 31.1 is on page 61.

This is helpful since labels will remember the numberings and page numbers for you.

LaTeX code

```
The following uses \verb!\label! and \verb!\ref!.

Referencing a theorem:
\begin{thm} \label{thm:important}
1 + 1 = 2
\end{thm}

Referencing a definition:
\begin{defn} \label{def:prime}
A \textbf{prime} is a positive integer divisible only
by 1 or itself.
\end{defn}

Look at the next page ...

\newpage

The above Theorem \ref{thm:important} is on page \pageref{thm:important}.
The above Definition \ref{def:prime} is on page \pageref{def:prime}.

This is helpful since labels will remember the numberings and
page numbers for you.

%You can create a reference to a chapter, section, subsection, etc.
%This set of notes does not talk about book style.
```

32 References: equations

You can create labels for equations like this:

$$x = 1 \tag{1}$$

The above equation (1) on page 63

You can also create labels in align:

$$\begin{aligned} x &= 1 + 1 + 1 && (2)\\ &= 2 + 1 && (3)\\ &= 3 && (4) \end{aligned}$$

The above equations are (2), (3), and (4) on page 63.

LaTeX code

```
You can create labels for equations like this:
\begin{equation} \label{eq:x-is-1}
x = 1
\end{equation}
The above equation
(\ref{eq:x-is-1}) on page \pageref{eq:x-is-1}

You can also create labels in align:
\begin{align}
x &= 1 + 1 + 1 \label{eq:three-1} \\
  &= 2 + 1     \label{eq:three-2} \\
  &= 3         \label{eq:three-3}
\end{align}
The above equations are
(\ref{eq:three-1}),
(\ref{eq:three-2}), and
(\ref{eq:three-3})
on page \pageref{eq:three-1}.
```

33 References: items

You can label items in a itemize or enumerate environment:

1. first point
2. second point

Please look at point 1 on page 64.

Note that it does not work for itemize.

(a) first point
(b) second point

Please look at point 33 on page 64.

LaTeX code

```
You can label items in a itemize or enumerate environment:
\begin{enumerate}
\item first point \label{itm:firstpoint}
\item second point \label{itm:secondpoint}
\end{enumerate}
Please look at point \ref{itm:firstpoint} on page \pageref{itm:firstpoint}.

Note that it does not work for itemize.
\begin{itemize}
\item[(a)]\label{itm:thirdpoint} first point
\item[(b)]\label{itm:fourthpoint} second point
\end{itemize}
Please look at point \ref{itm:thirdpoint} on page \pageref{itm:thirdpoint}.
```

34 References: text

To reference an arbitrary text do this:

Exercise 34.1. What is 1 + 1? (Go to Solution on page 66)

See next page ...

Solution (Exercise 34.1 on page 65.)
$1 + 1 = 2$. □

LaTeX code

```
To reference an arbitrary text do this:

\begin{ex} \label{ex-one-plus-one}
  What is 1 + 1?
  (Go to \hyperref[solution-my-fav-problem]{Solution} on
  page \pageref{sol-one-plus-one})
\end{ex}

See next page ...

\newpage

\textbf{Solution} \label{sol-one-plus-one}
(Exercise \ref{ex-one-plus-one} on page \pageref{ex-one-plus-one}.)
\\
$1 + 1 = 2$.
\qed
```

35 Index

Go to the last page of this pdf and you'll see an index.

If you want to have an index, you need to add `makeindex` and `printindex` to your LaTeX document:

```
...
\makeindex
\begin{document}
...
\printindex
\end{document}
```

To compile the pdf, you should execute this:

```
pdflatex main.tex
makeindex main.idx
pdflatex main.tex
```

To add an entry to your index, use the `index` command:

> "An integer d **divides** integer m if there is an integer x such that $dx = m$."

You can change the order of appearance of your index item. For instance if you are defining a symbol, such as |, you can do this:

> "$d \mid m$ if there is an integer x such that $dx = m$."

In this case, | will appear in your index but its order of appearance is determined using the string "divides".

Here's another example: $\int$.

Note that in the above I'm indexing mathematical symbols | and $\int$. I order | in the index using the word "divides". You can tell LaTeX that | is a mathematical symbol by doing , i.e., put a `$` in front of the indexing word. This will put symbols earlier in the index page. For instance ,

LaTeX code

```
Go to the last page of this pdf and you'll see an index.
```

```
If you want to have an index, you need to
add \verb!makeindex! and \verb!printindex!
to your \LaTeX\ document:
\begin{Verbatim}[frame=single, fontsize=\small]
...
\makeindex
\begin{document}
...
\printindex
\end{document}
\end{Verbatim}
To compile the pdf, you should execute this:
\begin{Verbatim}[frame=single, fontsize=\small]
pdflatex main.tex
makeindex main.idx
pdflatex main.tex
\end{Verbatim}

To add an entry to your index, use the \verb!index! command:

\begin{enumerate}
\item[]
\lq\lq An integer $d$ \textbf{divides}\index{divides}
integer $m$ if there is an integer $x$
such that $dx = m$."
\end{enumerate}

You can change the order of appearance of your
index item.
For instance if you are defining a symbol, such as $\mid$, you can do this:

\begin{enumerate}
\item[]
\lq\lq $d \mid m$ \index{divides@$\mid$} if there is an integer $x$
such that $dx = m$."
\end{enumerate}

In this case, $\mid$ will appear in your index but its
order of appearance is determined using the string \lq\lq divides".

Here's another example: $\int$ \index{integral@$\int$}.

Note that in the above I'm indexing mathematical symbols $\mid$ and $\int$.
I order $\mid$ in the index using the word \lq\lq divides".
You can tell \LaTeX\ that $\mid$ is a mathematical symbol by
doing \index{$divides@$\mid$}, i.e., put a \verb!$! in front of the
indexing word.
This will put symbols earlier in the index page.
For instance \index{$zeta@$\zeta$},
```

36 Document styles

I've create several "styles" that you can use. Here are some templates to get started.

This template is used for writing articles:

```
\input{myarticlepreamble}
\renewcommand\TITLE{How to add}
\renewcommand\AUTHOR{John Doe}
\begin{document}
\topmatter

\section{This is the first section}
hello world

\end{document}
```

This template is used for writing books:

```
\input{mybookpreamble}
\renewcommand\TITLE{How to add}
\renewcommand\AUTHOR{John Doe}

\begin{document}
\topmatter

\chapter{This is the first chapter}

\section{This is the first section}
hello world

\end{document}
```

By default the page width is 5in. You can change it by doing this:

```
\input{mybookpreamble}
\renewcommand\TITLE{How to add}
\renewcommand\AUTHOR{John Doe}

\textwidth=6in

\begin{document}
...
```

37 Resources

There are lots of people using LaTeX (especially CS, Engineering, Math, Physics people) and there are lots of LaTeX webpages and forums. You can easily find information on the web or help from people on the web.

Math notation: https://en.wikibooks.org/wiki/LaTeX/Mathematics

Stackexchange forum for TeX, LaTeX, etc.: https://tex.stackexchange.com/

Wiki guide for LaTeX: https://en.wikibooks.org/wiki/LaTeX

38 Miscellaneous

To negating a binary relation use `\not` or `\centernot` (`\centernot` is usually more accurate):

$$\not=, \quad \not\implies$$
$$\not=, \quad \centernot\implies$$

Parentheses for functions:

$$g(x) f \left(\frac{y}{z} \right)$$
$$g(x) f \left(\frac{y}{z} \right)$$

Use `f \mleft( \frac{y}{z} \mright)`.

LaTeX code

```
To negating a binary relation use \verb!\not! or \verb!\centernot!
(\verb!\centernot! is usually more accurate):
\begin{align*}
&\not=, \,\,\,\,\,\, \not\implies \\
&\centernot=, \,\,\,\,\,\, \centernot\implies
\end{align*}

Parentheses for functions:
\begin{align*}
&g(x) f \left( \frac{y}{z} \right) \\
&g(x) f \mleft( \frac{y}{z} \mright)
\end{align*}
Use \verb!f \mleft( \frac{y}{z} \mright)!.
```

Index