

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

---

Maxim Borisyak, Andrey Ustyuzhanin

Constructor University Bremen

February 24, 2026

**What is  $\text{\LaTeX}$ ?**

-

- A professional typesetting system for technical documents
- Widely used in Computer Science, Mathematics, Physics
- Ideal for documents with formulas, algorithms, references
- Not a word processor — you write source code that compiles into PDF

# $\text{\LaTeX}$ vs. Word Processors

---

Word:

$$KL(P \parallel Q) = E_{x \sim P} \log \frac{P(x)}{Q(x)}$$

Beamer ( $\text{\LaTeX}$  slides):

$$KL(P \parallel Q) = \mathbb{E}_{x \sim P} \log \frac{P(x)}{Q(x)}$$

Proper  $\text{\LaTeX}$ :

$$KL(P \parallel Q) = \mathbb{E}_{x \sim P} \log \frac{P(x)}{Q(x)}$$

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

---

## For academic writing:

- required by most CS conferences and journals;
- consistent, professional formatting;
- automatic numbering (sections, figures, equations, references);
- excellent math typesetting;
- reproducible builds from source.

## In practice:

- Overleaf: collaborative online L<sup>A</sup>T<sub>E</sub>X editor;
- local editors: TeXstudio, VS Code + LaTeX Workshop.

# Workflow

---

1. Write `.tex` source file;
2. Compile with `pdflatex` (or `xelatex`, `lualatex`);
3. Run `bibtex`/`biber` for references;
4. Recompile (twice) to resolve cross-references;
5. Get PDF output.

Most editors handle this automatically.

# Document Structure

—

# Minimal Document

---

A minimal  $\text{\LaTeX}$  document:

```
\documentclass{article}
```

```
\begin{document}
```

```
Hello, World!
```

```
\end{document}
```

- `\documentclass{...}` — type of document (article, report, book, beamer);
- `\begin{document}... \end{document}` — the document body.



# Preamble

---

Everything between `\documentclass` and `\begin{document}` is the **preamble**:

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

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

```
\usepackage{amsmath}
```

```
\usepackage{graphicx}
```

```
\usepackage{hyperref}
```

```
\title{My First Paper}
```

```
\author{Jane Doe}
```

```
\date{\today}
```

```
\begin{document}
```

```
\maketitle
```

```
...
```

```
\end{document}
```

# Common Document Classes

---

- `article` — short documents, papers;
- `report` — longer documents with chapters;
- `book` — books, theses;
- `beamer` — presentations (slides);
- many conferences provide their own class (e.g., `llncls`, `acmart`).

# Packages

---

Packages extend L<sup>A</sup>T<sub>E</sub>X functionality:

<code>\usepackage{amsmath}</code>	<i>% advanced math</i>
<code>\usepackage{graphicx}</code>	<i>% include images</i>
<code>\usepackage{hyperref}</code>	<i>% clickable links</i>
<code>\usepackage{booktabs}</code>	<i>% professional tables</i>
<code>\usepackage{algorithm2e}</code>	<i>% algorithm pseudocode</i>
<code>\usepackage{listings}</code>	<i>% source code</i>
<code>\usepackage{tikz}</code>	<i>% diagrams</i>

Load packages in the preamble with

`\usepackage[options]{name}.`

# Sectioning

---

L<sup>A</sup>T<sub>E</sub>X provides hierarchical sections:

```
\section{Introduction}
```

```
\subsection{Background}
```

```
\subsubsection{Details}
```

```
\paragraph{Note}
```

- Sections are automatically numbered;
- `\section*{...}` — unnumbered version;
- `\tableofcontents` — generates table of contents automatically.

## **Text Formatting**

---

## Basic Text Formatting

---

`\textbf{bold text}`

`\textit{italic text}`

`\texttt{monospace text}`

`\underline{underlined text}`

`\emph{emphasized}` % *italic in normal text,*  
                          % *normal in italic context*

**Result:** **bold text**, *italic text*, monospace text

# Lists

---

## Itemized (bullet) list:

```
\begin{itemize}
  \item First item
  \item Second item
\end{itemize}
```

## Enumerated (numbered) list:

```
\begin{enumerate}
  \item First item
  \item Second item
\end{enumerate}
```

Lists can be nested.

## Special Characters

---

Some characters have special meaning in  $\text{\LaTeX}$ :

`\%` `\$` `\&` `\#` `\_` `\{` `\}`  
`\textbackslash` `\textasciitilde`

- `%` — comment (rest of line is ignored);
- `$` — math mode delimiter;
- `&` — column separator in tables;
- `\\` — line break;
- `~` — non-breaking space.



## Whitespace and Line Breaks

---

- Multiple spaces are treated as one space;
- an empty line starts a new paragraph;
- `\\` forces a line break;
- `\newpage` starts a new page;
- `\noindent` suppresses paragraph indentation.

First paragraph text.

Second paragraph text.

`\noindent` Third paragraph, no indent.

# Mathematics

---

## Inline and Display Math

---

**Inline math** — within text:

The equation `$E = mc^2$` is famous.

The equation  $E = mc^2$  is famous.

**Display math** — centered on its own line:

`\[ E = mc^2 \]`

$$E = mc^2$$

## Numbered equation

---

Numbered equation:

```
\begin{equation}
  E = mc^2
  \label{eq:einstein}
\end{equation}
```

The equation~\eqref{eq:einstein} is famous.

$$E = mc^2 \tag{1}$$

The equation (1) is famous.

## Basic Math Notation

---

`x^2`                    *% superscript*  
`x_i`                    *% subscript*  
`x_{i+1}^{2}` *% grouped sub/superscript*  
`\frac{a}{b}`            *% fraction*  
`\sqrt{x}`            *% square root*  
`\sqrt[3]{x}`           *% cube root*

$$x^2 \quad x_i \quad x_{i+1}^2 \quad \frac{a}{b} \quad \sqrt{x} \quad \sqrt[3]{x}$$

# Greek Letters and Symbols

---

`\alpha \beta \gamma \delta \epsilon`  
`\theta \lambda \mu \sigma \omega`  
`\Gamma \Delta \Theta \Lambda \Omega`  
`\infty \partial \nabla \forall \exists`  
`\in \notin \subset \cup \cap`  
`\leq \geq \neq \approx \equiv`

$\alpha \beta \gamma \delta \epsilon \quad \theta \lambda \mu \sigma \omega$   
 $\Gamma \Delta \Theta \Lambda \Omega \quad \infty \partial \nabla \forall \exists$

# Sums, Products, Integrals

---

`\sum_{i=1}^n x_i`

`\prod_{i=1}^n x_i`

`\int_a^b f(x) \, dx`

`\lim_{n \rightarrow \infty} a_n`

$$\sum_{i=1}^n x_i \quad \prod_{i=1}^n x_i \quad \int_a^b f(x) \, dx \quad \lim_{n \rightarrow \infty} a_n$$

# Matrices

---

```
\begin{pmatrix}
  a & b \\
  c & d
\end{pmatrix}
```

```
\begin{bmatrix}
  1 & 0 \\
  0 & 1
\end{bmatrix}
```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$$

- `pmatrix` — round brackets;
- `bmatrix` — square brackets;
- `vmatrix` — determinant (vertical bars).



## Aligned Equations

---

The align environment (from amsmath):

```
\begin{align}
f(x) &= x^2 + 2x + 1 \\
      &= (x + 1)^2
\end{align}
```

- & marks alignment point;
- \\ separates lines;
- align\* for unnumbered version.

$$f(x) = x^2 + 2x + 1 \tag{2}$$

$$= (x + 1)^2 \tag{3}$$

## Common Math Environments

---

- `equation` — single numbered equation;
- `align` — multiple aligned equations;
- `gather` — multiple centered equations;
- `cases` — piecewise definitions:

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

$$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \quad (4)$$

## Example: Machine Learning

---

A practical example — softmax function:

$$\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c})$$

$$\mathrm{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad \mathcal{L} = -\sum_{i=1}^N \sum_{c=1}^C y_{i,c} \log(\hat{y}_{i,c})$$

# Math Alphabets

---

Different letter styles carry different meaning:

```
\mathbb{R}      % blackboard bold
\mathcal{L}     % calligraphic
\mathbf{x}      % bold upright
\boldsymbol{\theta} % bold italic
\mathit{MSE}    % italic
\mathrm{softmax} % upright roman
```

$\mathbb{R}$     $\mathcal{L}$     $\mathbf{x}$     $\boldsymbol{\theta}$     $MSE$     $\mathrm{softmax}$

Prefer `\DeclareMathOperator` over `\mathrm` for named functions.

## Auto-sizing Delimiters

---

Use `\left` and `\right` to make delimiters scale with content:

```
\left( \frac{a}{b} \right)
```

```
\left[ \sum_{i=1}^n x_i \right]
```

```
\left\{ x \in \mathbb{R} \mid x > 0 \right\}
```

```
\left\| A \right\|_F
```

```
\left| \sum_i \frac{1}{\sqrt{a_i}} \psi_i \right\rangle
```

$$\left(\frac{a}{b}\right) \quad \left[\sum_{i=1}^n x_i\right] \quad \{x \in \mathbb{R} \mid x > 0\} \quad \left|\sum_i \frac{1}{\sqrt{a_i}} \psi_i\right\rangle$$

## Auto-sizing Delimiters

---

`\middle` scales a delimiter in the middle:

`P\left( A \;;\middle|\;; B \right)`

`\mathrm{KL}\left(P \;;\middle|\;; Q\right)`

`\left< \sum_i \psi_i \;;\middle|\;; \sum_i \phi_i \right>`

$$P(A \mid B)$$

$$\mathrm{KL}(P \parallel Q)$$

$$\left\langle \sum_i \psi_i \middle| \sum_i \phi_i \right\rangle$$

`\right.` — invisible right delimiter.

## Auto-sizing Delimiters

---

`\left` and `\right` automatically adjusts with nesting:

```
L(\theta) = \prod_j \left[ \frac{1}{Z_j} \left( \sum_i f_j \left( \frac{P(X)}{Q(X)} \right) \right) \right]
```

$$L(\theta) = \prod_j \left[ \frac{1}{Z_j} \left( \sum_i f_j \left( \frac{P(X)}{Q(X)} \right) \right) \right]$$

## Equation Annotations

---

Annotate parts of an equation with `\underbrace` and `\overbrace`:

```
\underbrace{x^{\top} W x}_{\text{quadratic form}}  
+ \overbrace{\lambda \|\theta\|^2}^{\text{regularizer}}
```

$$\underbrace{x^{\top} W x}_{\text{quadratic form}} + \overbrace{\lambda \|\theta\|^2}^{\text{regularizer}}$$



## Equation Annotations

---

Place symbols above or below with `\overset` and `\underset`:

A `\overset{\text{def}}{=}` B

x `\underset{n \to \infty}{\longrightarrow}` 0

$$A \overset{\text{def}}{=} B \qquad x \underset{n \rightarrow \infty}{\longrightarrow} 0$$

$\text{\TeX}$  sometimes needs spacing hints inside math:

---

Command	Size	Typical use
$\!$	$-\frac{3}{18}\text{ mu}$	tighten adjacent symbols
$\,$	$\frac{3}{18}\text{ mu}$	before $dx$ in integrals
$\:$	$\frac{4}{18}\text{ mu}$	binary operators
$\;$	$\frac{5}{18}\text{ mu}$	before/after text in math
$\quad$	1 em	separating cases or labels
$\qquad$	2 em	wide alignment gaps

---

## Math Spacing: common patterns

---

<code>\int_a^b f(x) \, dx</code>	<i>% thin space before dx</i>
<code>\mathrm{d}x</code>	<i>% upright d</i>
<code>f : \mathbb{R}^n \rightarrow \mathbb{R}</code>	<i>% colon in function type</i>
<code>P\left(A \mid B\right)</code>	<i>% \mid gives correct spacing</i>

$$\int_a^b f(x) \, dx$$

$$dx$$

$$f: \mathbb{R}^n \rightarrow \mathbb{R}$$

$$P(A \mid B)$$

## **Floats: Figures and Tables**

---

## Including Figures

---

```
\begin{figure}[htbp]
  \centering
  \includegraphics[width=0.8\textwidth]{image.png}
  \caption{Description of the figure.}
  \label{fig:example}
\end{figure}
```

- [htbp] — placement hints: here, top, bottom, page;
- \centering — center the figure;
- \caption — figure caption (auto-numbered);
- \label — for cross-referencing.

## Tables: example

---

```
\begin{table}[htbp]
  \centering
  \caption{Comparison of optimization methods.}
  \label{tab:optimizers}
  \begin{tabular}{lccc}
    \toprule
    Optimizer & Learning Rate & Epochs & Test Acc. \\
    \midrule
    SGD & 0.01 & 100 & 0.88 \\
    Adam & 0.001 & 50 & 0.91 \\
    AdamW & 0.001 & 50 & \textbf{0.93} \\
    \bottomrule
  \end{tabular}
\end{table}
```

Use booktabs package for `\toprule`, `\midrule`, `\bottomrule`.

## Tables: example

---

**Table 1:** Comparison of optimization methods.

Optimizer	Learning Rate	Epochs	Test Acc.
SGD	0.01	100	0.88
Adam	0.001	50	0.91
AdamW	0.001	50	<b>0.93</b>

## Table Column Types

---

Column alignment specifiers in `\begin{tabular}{...}`:

- `l` — left-aligned;
- `c` — centered;
- `r` — right-aligned;
- `p{width}` — paragraph column with fixed width;
- `|` — vertical line (avoid in professional tables).

```
\begin{tabular}{l|cc|r}  
Name & Score 1 & Score 2 & Total \\  
\hline  
Alice & 95 & 88 & 183 \\  
\end{tabular}
```



## Table Column Types

---

```
\begin{tabular}{l|cc|r}  
  Name & Score 1 & Score 2 & Total \\  
  \hline  
  Alice & 95 & 88 & 183 \\  
\end{tabular}
```

Name	Score 1	Score 2	Total
Alice	95	88	183

## **Cross-references and Citations**

---

## Labels and References

---

L<sup>A</sup>T<sub>E</sub>X can reference any numbered element:

```
\section{Introduction}
```

```
\label{sec:intro}
```

As shown in Figure~\ref{fig:arch}

and Equation~\eqref{eq:loss}

(see Section~\ref{sec:intro}).

- `\label{...}` — place after `\caption`, `\section`, or in an equation;
- `\ref{...}` — produces the number;
- `\eqref{...}` — equation reference with parentheses;
- `~` — non-breaking space before the reference.

# Bibliography with BibTeX

---

Create a .bib file:

```
@article{vaswani2017attention,  
  title={Attention is all you need},  
  author={Vaswani, Ashish and others},  
  journal={NeurIPS},  
  year={2017}  
}
```

In the document:

```
Transformers~\cite{vaswani2017attention}  
have revolutionized NLP.
```

```
\bibliographystyle{plain}  
\bibliography{references}
```

## Citation Commands

---

Basic commands:

- `\cite{key}` — [1] or (Author, Year);
- `\cite{key1, key2}` — multiple citations.

With natbib package:

- `\citet{key}` — textual: Author (Year);
- `\citep{key}` — parenthetical: (Author, Year);
- `\citep[see][]{key}` — (see Author, Year).

## Compilation with References

---

To resolve all references:

```
pdflatex document
```

```
bibtex document
```

```
pdflatex document
```

```
pdflatex document
```

Multiple passes needed because:

1. first pass collects labels and citations;
2. bibtex generates bibliography;
3. second pass inserts references;
4. third pass resolves page numbers.

## **Useful Environments and Tips**

---

## Code Listings

---

Using the listings package:

```
\begin{lstlisting}[language=Python]
def hello():
    print("Hello, World!")
\end{lstlisting}
```

Or the minted package (requires Python's Pygments):

```
\begin{minted}{python}
def hello():
    print("Hello, World!")
\end{minted}
```



## Theorem-like Environments

---

Define in the preamble:

```
\newtheorem{theorem}{Theorem}  
\newtheorem{lemma}{Lemma}  
\newtheorem{definition}{Definition}
```

Use in the document:

```
\begin{theorem}  
  For all  $n \geq 1$ :  
   $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ .  
\end{theorem}  
\begin{proof}  
  By induction on  $n$ ...  
\end{proof}
```

## Theorem-like Environments

---

### Theorem

*For all  $n \geq 1$ :  $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ .*

### Proof.

By induction on  $n$ ...



### Avoid these pitfalls:

- forgetting {} for multi-character sub/superscripts:  $x^{10}$  vs.  $x^{\{10\}}$ :
  - $x^{10}$  vs.  $x^{10}$
- using " instead of and ' ' for quotes;
- not escaping special characters: 50% should be 50\%;
- placing \label before \caption (label will be wrong);
- ignoring warnings about undefined references.

## Useful Commands to Know

---

<code>\phantom{text}</code>	<i>% invisible space</i>
<code>\vspace{1cm}</code>	<i>% vertical space</i>
<code>\hspace{1cm}</code>	<i>% horizontal space</i>
<code>\textcolor{red}{txt}</code>	<i>% colored text (xcolor)</i>
<code>\url{https://...}</code>	<i>% typeset URL (hyperref)</i>
<code>\footnote{text}</code>	<i>% footnote</i>
<code>\todo{Fix this}</code>	<i>% margin note (todonotes)</i>

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

---

## T<sub>E</sub>X: the Engine

---

L<sup>A</sup>T<sub>E</sub>X is a **macro package** built on top of T<sub>E</sub>X:

- T<sub>E</sub>X — a Turing-complete macro expansion language + typesetting engine (Knuth, 1978);
- L<sup>A</sup>T<sub>E</sub>X provides a high-level API: `\begin`, `\newcommand`, `\usepackage`, ...;
- understanding T<sub>E</sub>X fundamentals helps when L<sup>A</sup>T<sub>E</sub>X macros need to be more powerful.

**Engine family:** pdfT<sub>E</sub>X · XeT<sub>E</sub>X · LuaT<sub>E</sub>X — all extend the original T<sub>E</sub>X engine.

**Macro packages:** plain T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt — different macro layers on the same engine.

## Tokens and Category Codes

---

$\text{\TeX}$  reads input character by character and assigns each a **category code** (catcode):

- **catcode 0** — escape character `\`: starts a control sequence;
- **catcode 1/2** — begin/end group `{ }`: creates a local scope;
- **catcode 3** — math shift `$`: enters/exits math mode;
- **catcode 11** — letter: part of a control sequence name;
- **catcode 14** — comment `%`: rest of line is ignored.

A **control sequence** like `\frac` is a single token regardless of its length.

Catcodes can be changed — this is how `\verb` works: it temporarily makes `\` an ordinary character.

## Groups and Scoping

---

`{}` create a **local group** — all assignments inside are reverted on exit:

```
{\color{red} red text}      % color reverts after }  
{\large big text}          % font size reverts after }  
{\def\x{local}}           % \x is undefined after }
```

All T<sub>E</sub>X assignments are **local by default**:

- `\setlength`, `\def`, font commands, color, etc.;
- `\global` prefix makes an assignment permanent (use sparingly).

This is why font/color commands are typically written as `{\large text}` — no explicit reset needed.



## `\def`: the TeX Primitive

---

The primitive `\def` defines macros directly:

<code>\def\R{\mathbb{R}}</code>	<i>% no arguments</i>
<code>\def\norm#1{\left  #1 \right }</code>	<i>% one argument</i>
<code>\def\pair#1#2{\langle #1, #2 \rangle}</code>	<i>% two arguments</i>

`\def` vs `\newcommand`:

- `\def` silently **overwrites** any existing command — dangerous;
- `\newcommand` **errors** if the command already exists — safe;
- `\renewcommand` explicitly signals the intent to redefine.

Use `\newcommand` in  $\text{\LaTeX}$  documents; `\def` in low-level macro code.

# Expansion

---

T<sub>E</sub>X processes tokens in two phases:

1. **Expansion:** expandable tokens (macros, `\if...`) are replaced by their definition, repeatedly, until only unexpandable tokens remain;
2. **Execution:** unexpandable tokens are executed (assignments, typesetting).

`\expandafter` expands the token *after* the next one first:

```
\def\hello{world}
```

```
\expandafter\MakeUppercase\expandafter{\hello}
```

*% equivalent to: \MakeUppercase{world}*

`\noexpand`: suppresses expansion of the following token.

## \edef: Expanding Definition

---

\edef expands the replacement text **at definition time**:

```
\newcommand{\myname}{Alice}  
\edef\greeting{Hello, \myname!}  
% \greeting is now literally "Hello, Alice!"
```

```
\renewcommand{\myname}{Bob}  
\greeting      % still outputs: Hello, Alice!
```

Compare with \def / \newcommand — expands at **use time**:

```
\newcommand{\greeting}{Hello, \myname!}  
\renewcommand{\myname}{Bob}  
\greeting      % outputs: Hello, Bob!
```

\xdef = \global\edef (global expanding def).

## Box Model

---

Everything in  $\text{T}_{\text{E}}\text{X}$  is a **box** with width, height, and depth:

- `\hbox{...}` (`\mbox`) — horizontal box, no line breaks inside;
- `\vbox{...}` (`minipage`, `parbox`) — vertical box;
- boxes can be nested, measured, and positioned freely.

```
\makebox[3cm][c]{centered}      % fixed-width centered box  
\framebox{framed content}      % box with visible border  
\raisebox{0.5ex}{raised text}  % shift box up/down
```

$\text{\LaTeX}$  builds lines from **horizontal lists** of boxes and **glue** (stretchable/shrinkable space). Tables, columns, and figures are all composed of boxes.

## Low-level Conditionals

---

TeX primitives for conditionals (no package needed):

```
\ifnum\value{page} > 1
```

```
    Not the first page.
```

```
\fi
```

```
\ifdim\textwidth > 10cm
```

```
    Wide page.
```

```
\else
```

```
    Narrow page.
```

```
\fi
```

```
\ifx\myvar\undefined
```

```
    \myvar is not defined.
```

```
\fi
```

## Low-level Conditionals

---

- `\ifnum`: integer comparison (`<`, `>`, `=`);
- `\ifdim`: dimension comparison;
- `\ifx`: tests if two tokens identical;
- `\if`: compare character codes of two tokens;
- every conditional must end with `\fi`.

## Custom Commands: `\newcommand`

---

Define reusable shorthands, especially useful for math notation:

```
\newcommand{\R}{\mathbb{R}}
\newcommand{\N}{\mathbb{N}}
\newcommand{\norm}[1]{\left\| #1 \right\|}
\newcommand{\inner}[2]{\left\langle #1, #2 \right\rangle}
```

Usage:

Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $x \in \mathbb{R}^n$ .

Then  $\|x\| = \sqrt{\langle x, x \rangle}$ .

Let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  and  $x \in \mathbb{R}^n$ . Then  $\|x\|_{L^2} = \sqrt{\langle x, x \rangle}$ .

Define once in the preamble, use everywhere, change notation globally by editing one line.

`\DeclareMathOperator` defines named operators typeset in upright font:

```
\DeclareMathOperator{\argmax}{arg\,max}
```

```
\DeclareMathOperator{\rank}{rank}
```

```
\DeclareMathOperator{\tr}{tr}
```

```
\DeclareMathOperator{\diag}{diag}
```

```
\DeclareMathOperator{\KL}{KL}
```

$$\arg \max_{\theta} \mathcal{L}(\theta) \qquad \text{rank}(A) \qquad \text{tr}(M) \qquad \text{KL}(P\|Q) \qquad (5)$$



The \* variant allows limits above/below in display math:

```
\DeclareMathOperator*{\E}{\mathbb{E}}
```

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

$$\mathbb{E}_{x \sim p} f(x) \qquad \arg \min_{\theta \in \Theta} \mathcal{L}(\theta)$$

Use `\DeclareMathOperator` instead of `\mathrm` or `\text`, it handles spacing around the operator correctly:

```
\DeclareMathOperator \mathrm
```

$$\arg \max_{\theta \in \Theta} \mathcal{L}(\theta) \qquad \arg \max_{\theta \in \Theta} \mathcal{L}(\theta);$$

$$\arg \min_{\theta \in \Theta} \mathcal{L}(\theta) \qquad \arg \min_{\theta \in \Theta} \mathcal{L}(\theta);$$

$$\mathbb{D}_{x \sim p} f(x) \qquad \mathbb{D}_{x \sim p} f(x).$$

## Custom Commands: optional arguments

---

`\newcommand` supports optional arguments with defaults:

*% [1] = 1 argument, [L^2] = default value*

```
\newcommand{\norm}[2][L^2]{%  
  \left\| #2 \right\|_{#1}%  
}
```

`$\norm{f}$`  $\Rightarrow$  `\left\|f\right\|_{L^2}`

`$\norm[L^\infty]{f}$`  $\Rightarrow$  `\left\|f\right\|_{L^\infty}`

$$\|f\|_{L^2} \qquad \|f\|_{L^\infty}$$

- #1 — first argument (optional, with default);
- #2 — second argument (mandatory);
- % at end of line, suppress unwanted whitespace.

Override existing commands:

```
% Change equation numbering to include section  
\renewcommand{\theequation}{%  
  \thesection.\arabic{equation}%  
}
```

```
% Redefine abstract name  
\renewcommand{\abstractname}{Executive Summary}
```

- Use `\renewcommand` for commands that already exist;
- use `\newcommand` for new commands, it errors if the command already exists (safety check).

`\newenvironment{name}[args]{begin-code}{end-code}:`

```
\newenvironment{important}{%  
  \begin{center}  
    \begin{minipage}{0.9\textwidth}  
      \textbf{Important:} \itshape  
    }{%  
      \end{minipage}  
      \end{center}  
}
```

Usage:

```
\begin{important}
```

Always compile twice to resolve references.

```
\end{important}
```

Result:

**Important:** *Always compile twice to resolve references.*

## Custom Environments with Arguments

---

Environments can also accept arguments:

```
\newenvironment{method}[1]{%  
  \begin{block}{Method: #1}  
}{%  
  \end{block}  
}
```

## Custom Environments with Arguments

---

```
\begin{method}{Gradient Descent}
  Update rule:  $\theta \leftarrow \theta$ 
    -  $-\alpha \nabla_{\theta} \mathcal{L}$ 
\end{method}
```

### Method: Gradient Descent

Update rule:  $\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}$

Combine with `\newcommand` to build a consistent notation system for your paper or thesis.

# Counters

---

L<sup>A</sup>T<sub>E</sub>X uses counters for automatic numbering:

```
\newcounter{example}[section]    % reset per section
```

```
\newcommand{\example}{%  
  \stepcounter{example}%  
  \textbf{Example~\thesection.\theexample}%  
}
```

Useful counter commands:

- `\newcounter{name}[parent]` — create a counter;
- `\setcounter{name}{value}` — set value;
- `\stepcounter{name}` — increment by 1;
- `\arabic{name}`, `\roman{name}` — display formats.



Control spacing and sizes with length registers:

```
\newlength{\mywidth}
```

```
\setlength{\mywidth}{0.45\textwidth}
```

```
\addtolength{\mywidth}{-1cm}
```

- `\newlength` — define a length register;
- `\setlength` — assign a value;
- `\addtolength` — add to current value;
- units: pt, cm, em, ex, or relative like `0.5\textwidth`.

## Conditionals

---

The `ifthen` package provides readable conditionals:

```
\usepackage{ifthen}

\newcommand{\grade}[1]{%
  \ifthenelse{#1 > 89}{A}{%
    \ifthenelse{#1 > 79}{B}{%
      \ifthenelse{#1 > 69}{C}{F}}}%
}
```

Score 95: <code>\grade{95}</code>	Score 95: A
Score 72: <code>\grade{72}</code>	Score 72: C

Also available: `\ifdefined`, `\ifx` (low-level T<sub>E</sub>X conditionals) and the `etoolbox` package for modern alternatives.

## Loops with pgffor

---

The pgffor package (part of TikZ/PGF) provides `\foreach`:

```
\usepackage{pgffor}
```

```
\foreach \x in {1,2,...,5} {  
  Item \x;\,  
}
```

```
\foreach \name/\score in {  
  Alice/95, Bob/87, Carol/91} {  
  \name: \score\\  
}
```

Item 1; Item 2; Item 3; Item 4; Item 5;

Useful for generating repetitive content: tables, TikZ diagrams, lists.

## Practical Example: Paper Preamble

---

A real-world preamble with custom commands:

```
% Number sets
\newcommand{\R}{\mathbb{R}}

% Operators
\DeclareMathOperator*{\E}{\mathbb{E}}
\DeclareMathOperator*{\argmin}{arg\,min}
\DeclareMathOperator{\KL}{KL}

% Shortcuts
\newcommand{\loss}{\mathcal{L}}
\newcommand{\model}{f_{\theta}}
\newcommand{\dataset}{\mathcal{D}}

% Norms and brackets
\newcommand{\norm}[1]{\left\| \ #1 \right\|}
\newcommand{\abs}[1]{\left| \ #1 \right|}
```

## Practical Example: Usage

---

With the preamble above, you write concise math:

```
\begin{align*}
\loss(\theta) &= \mathbb{E}_{(x,y) \sim \text{dataset}}
               \|\text{model}(x) - y\|^2; \\
\theta^* &= \operatorname{argmin}_{\theta} \loss(\theta).
\end{align*}
```

$$\mathcal{L}(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \|f_{\theta}(x) - y\|_{L^2}^2;$$
$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta).$$

Readable source, beautiful output — and easy to change notation globally.

## Conclusion

---

L<sup>A</sup>T<sub>E</sub>X is essential for academic CS writing:

- **document structure:** `\documentclass`, `preamble`, `environments`;
- **math typesetting:** `inline` `$...$`, `display` `\[...\]`, `align`;
- **floats:** `figure`, `table` with `\caption` and `\label`;
- **references:** `\label/\ref`, BibTeX for citations;
- **packages:** extend functionality as needed.