1 md2tex

A small, no-frills Markdown → LaTeX/PDF converter written in Python. One command takes a `.md` file and produces `.tex` and `.pdf` with sensible defaults.

1.1 Highlights

- One-step pipeline: `.md → .tex → .pdf` (runs a LaTeX engine for you)
- Cross-platform engine detection: `pdflatex`, `xelatex`, or `lualatex` (Windows/Linux/macOS)
- Engine-flexible LaTeX preamble (via `iftex`) so the same `.tex` compiles on Overleaf and locally with pdfLaTeX/LaTeX or XeLaTeX/LuaLaTeX
- Unicode-safe behavior:
  - Preserves raw Unicode exactly inside fenced code blocks (“‘/~~~)
  - Uses a Unicode-capable engine automatically when needed
- Emoji/sticker removal by Unicode ranges (no per-emoji lists)
- Inline formatting: `**bold**`, `[links](url)`, and ‘ inline code ‘
- Math: inline ‘..., `display $$...$$, and bracketed display blocks using lines with [ and ]`‘
- Auto-wraps common math used in text (e.g., `\alpha`, `\int`, `\vec{x}`, `x_1`, `x^2`) into ‘...‘
- Tables: GitHub-style pipe tables scaled to page width
- Lists: ordered/unordered, nested by indentation (2 spaces per level)
- Headings: `#` → `\section`, `##` → `\subsection`, `###` → `\subsubsection`, `####` → `\paragraph`
  - Leading numeric prefixes like `1. Title` are stripped from heading text
  - Subsubsections numbered as `1, 2, 3` (no `0.0.1`)
- Horizontal rules: `---`, `***`, or ____
- Newlines: a single newline in Markdown becomes a visible line break in LaTeX (`\newline`)
- Cleanup: removes LaTeX aux files after a successful build (keeps only `.md`, `.tex`, `.pdf`, `.py`)

1.2 Requirements

- Python 3.8+
- A LaTeX distribution:
  - Windows: MiKTeX or TeX Live
  - Linux/macOS: TeX Live

The script auto-detects engines via PATH and known install locations and prefers `xelatex/lualatex` when it detects non-ASCII inside fenced code blocks; otherwise it uses `pdflatex`.
The generated `.tex` includes an engine-aware preamble: pdfTeX uses `inputenc` + `T1` + `lmodern`, while Xe/LuaLaTeX use `fontspec`.

1.3 Usage

- Windows (PowerShell):

```
python md2tex.py your_file.md
```

- Linux/macOS:

```
python3 md2tex.py your_file.md
```

Outputs:

- `your_file.tex` — generated LaTeX
- `your_file.pdf` — compiled PDF (if a LaTeX engine is installed)

Tip: If you run the script without a file, or pass ”/” or ”.”, it defaults to `README.md` in the current directory. For example:

- Windows (PowerShell):

```
python md2tex.py
python md2tex.py /
```

- Linux/macOS:

```
python3 md2tex.py
python3 md2tex.py /
```

1.4 Markdown support details

- Paragraphs/newlines
  - Single newline → `\\newline` (forced line break)
  - Blank line → paragraph break
- Code
  - Fenced code blocks (“/ `) are emitted as` verbatim‘ with Unicode preserved
  - Inline code uses `\texttt{...}` with safe escaping
- Math
  - Inline: ‘...‘
  - Display: `$$...$$` or bracketed block between lines `[` and `]`
  - Literal `$$...$$` text is preserved (escaped) in regular paragraphs
  - Auto-math wrapping: if you accidentally use math commands in text (e.g., `\alpha`, `\int`, `x_1`, `x^2`, `\vec{x}`), they are wrapped into ‘...‘ automatically
- Tables
  - Pipe tables with a header and a separator line are supported and auto-scaled to `\textwidth`
- Lists
  - `-`, `*` unordered; `1.` ordered
  - Nesting by 2-space indentation per level
- Headings
  - `#`, `##`, `###`, `####` → LaTeX sectioning commands
  - Leading numbering like `1. Title` in source is removed from the title text
- Emojis/stickers
  - Removed globally by Unicode ranges (flags, pictographs, emoticons, dingbats, skin tones, VS-16/ZWJ)

1.5 Troubleshooting

- “No LaTeX engine found”
  - Install MiKTeX (Windows) or TeX Live (Linux/macOS) and ensure binaries are on PATH
- “PDF compilation failed”
  - Check the generated `.tex` next to your `.md`
  - Make sure packages like `amsmath`, `hyperref`, `adjustbox` are available in your LaTeX install
- “fontspec only works with Xe/LuaLaTeX”
  - The output `.tex` avoids loading `fontspec` on pdfLaTeX/LaTeX via `iftex`. If you manually edit the preamble, keep `fontspec` under the Xe/Lua branch only.
- “Unicode in code block breaks with pdflatex”
  - The script prefers `xelatex/lualatex` when it detects non-ASCII in code fences; install one of them if missing
- Overleaf notes
  - You can compile the same `.tex` with pdfLaTeX, XeLaTeX, or LuaLaTeX. If you hit Unicode issues, switch the Overleaf compiler to XeLaTeX or LuaLaTeX.

1.6 Known limitations (by design)

- Images, blockquotes, and task lists are not implemented (kept intentionally simple)
- This is not a full Markdown parser; it covers the most common patterns used in notes/technical docs

1.7 Comprehensive sample (README as test)

This README doubles as the end-to-end test document. You can run the converter directly on it to produce a PDF:

- Windows (PowerShell):

```
python md2tex.py README.md
```

- Linux/macOS:

```
python3 md2tex.py README.md
```

Below is the full “hard cases” sample previously in `test.md`.

2 Advanced Mathematical Document

This document tests all markdown features including special characters, equations, tables, and more.

2.1 Mathematical Equations

1 Inline Mathematics

The quadratic formula is $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ and Euler’s identity is $e^{i\pi} + 1 = 0$.

The area of a circle: $A = \pi r^2$ where r is the radius.

2 Block Equations

Partial Differential Equation (Heat Equation):

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Navier-Stokes Equation:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$$

Integral Example:

$$\int_0^\infty e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$$

Double Integral:

$$\iint_D f(x,y)\, dA = \int_a^b \int_c^d f(x,y)\, dy\, dx$$

Matrix Example:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Matrix Multiplication:

$$\mathbf{C} = \mathbf{A} \times \mathbf{B} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

Square Root and Fractions:

$$\sqrt{x^2 + y^2} = \sqrt{\frac{a}{b} + \frac{c}{d}}$$

Summation and Product:

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6} \quad \text{and} \quad \prod_{i=1}^n i = n!$$

Limit Example:

$$\lim_{x \to \infty} \frac{1}{x} = 0$$

Taylor Series:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

2.2 Special Characters & Symbols

1 Greek Letters

α (alpha), β (beta), γ (gamma), δ (delta), ε (epsilon), ζ (zeta), η (eta), θ (theta), λ (lambda), μ (mu), π (pi), σ (sigma), τ (tau), φ (phi), ω (omega)

Uppercase: Γ (Gamma), Δ (Delta), Θ (Theta), Λ (Lambda), Ξ (Xi), Π (Pi), Σ (Sigma), Φ (Phi), Ψ (Psi), Ω (Omega)

2 Mathematical Operators

$\pm \mp \times \div \cdot \surd \sqrt[3]{} \sqrt[4]{} \infty \propto \approx \neq \equiv \leq \geq \subset \supset \subseteq \supseteq \cap \cup \int \oint \partial \nabla \Delta \prod \sum$

3 Other Symbols

© ® ™ € £ ¥ ¢ § ¶ † ‡ • ‰ ′ ″ ‴ → ← ↑ ↓ ↔ ⇒ ⇐ ⇔

2.3 Complex Tables

1 Table 1: Special Characters in Cells

Symbol	Name	LaTeX	Unicode
α	Alpha	`\alpha`	U+03B1
β	Beta	`\beta`	U+03B2
$\int$	Integral	`\int`	U+222B
$\sum$	Sum	`\sum`	U+2211
$\surd$	Square Root	`\sqrt{}`	U+221A
∞	Infinity	`\infty`	U+221E
$\approx$	Approximately	`\approx`	U+2248
$\neq$	Not Equal	`\neq`	U+2260

2 Table 2: Mathematical Constants

Constant	Symbol	Approximate Value	Formula
Pi	π	3.14159265359	$\pi = \frac{C}{d}$
Euler's Number	e	2.71828182846	$e = \lim_{n \to \infty}(1 + \frac{1}{n})^n$
Golden Ratio	φ	1.61803398875	$\phi = \frac{1+\sqrt{5}}{2}$
Planck's Constant	h	$6.62607015 \times 10^{-34}$ J·s	$E = h\nu$

3 Table 3: Programming Languages & Operators

Language	Addition	Multiplication	Division	Modulo	Power
Python	`a + b`	`a * b`	`a / b`	`a % b`	`a ** b`
C++	`a + b`	`a * b`	`a / b`	`a % b`	`pow(a, b)`
JavaScript	`a + b`	`a * b`	`a / b`	`a % b`	`a ** b`
Java	`a + b`	`a * b`	`a / b`	`a % b`	`Math.pow(a, b)`

2.4 Code Blocks

1 Python Code with Special Characters

```
import numpy as np
import matplotlib.pyplot as plt

# Calculate π using Monte Carlo method
def estimate_pi(n_samples=1000000):
    """Estimate π using random points in a square"""
    x = np.random.uniform(-1, 1, n_samples)
    y = np.random.uniform(-1, 1, n_samples)
    inside_circle = (x**2 + y**2) <= 1
    pi_estimate = 4 * np.sum(inside_circle) / n_samples
    return pi_estimate

# Test with special operators: +, -, *, /, %, **, //, &, |, ^, ~, <<, >>
result = (2 ** 3) * (10 // 3) + (15 % 4) - (100 / 7)
print(f"Result: {result:.4f}")

# Unicode in strings
symbols = "α β γ δ ε ζ η θ λ μ π σ τ φ ω"
operators = "± × ÷ √ ∞ ≈ ≠ ≤ ≥ ∫ ∑"
```

2 LaTeX Equation

```
\begin{equation}
\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}
\end{equation}

\begin{align}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \times \mathbf{B} &= \mu_0 \mathbf{J} + \mu_0 \epsilon_0 \frac{\part
ial \mathbf{E}}{\partial t}
\end{align}
```

2.5 Lists with Special Characters

1 Unordered List

- Item with α (alpha) and β (beta)
- Mathematical operators: $\int \sum \prod \surd$
- Comparison: $\approx \neq \leq \geq \infty$
- Arrows: $\rightarrow \leftarrow \uparrow \downarrow \leftrightarrow$
- Symbols: © ® ™ € £ ¥

2 Ordered List

1. First: Calculate $\int_0^1 x^2 dx = \frac{1}{3}$
2. Second: Evaluate $\sum_{i=1}^{10} i = 55$
3. Third: Solve $\frac{dy}{dx} = 2x$ to get $y = x^2 + C$
4. Fourth: Matrix multiplication $\mathbf{A} \times \mathbf{B}$
5. Fifth: Compute $\lim_{x \to 0} \frac{\sin x}{x} = 1$

3 Nested Lists

- Top level with $\pi \approx 3.14159$
  - Nested with $e^{i\pi} + 1 = 0$
  - Another nested: $\sqrt{-1} = i$
- Another top: ∞ (infinity)
  - Sub-item: $\lim_{n \to \infty}$

2.6 Text Formatting Tests

Bold text with _italic text_ and `inline code` with special chars: `α_β^γ`

Regular text with **bold**, *italic*, and ***bold italic*** combined.

Text with special characters: @ # $ % ^ & * ( ) _ + = { } [ ] | \ : ; ” ’ < > , . ? /

Escaped characters test: _underscore_ *asterisk* \#hash\# \\dollar$\%*percent*\%

2.7 Links and References

Visit Python Official for documentation.

Check out NumPy for numerical computing.

Mathematical reference: Wolfram MathWorld

2.8 Advanced Equations Section

1 Schrödinger Equation

$$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$$

2 Maxwell’s Equations

$$\begin{aligned}
\nabla \cdot \mathbf{E} &= \frac{\rho}{\epsilon_0} \\
\nabla \cdot \mathbf{B} &= 0 \\
\nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\
\nabla \times \mathbf{B} &= \mu_0\mathbf{J} + \mu_0\epsilon_0\frac{\partial \mathbf{E}}{\partial t}
\end{aligned}$$

3 Einstein Field Equations

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}$$

4 Fourier Transform

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)e^{-2\pi i x\xi}dx$$

2.9 Conclusion

This document contains:

- Multiple heading levels (# ## ### ####)
- Tables with special characters (α β γ π $\sum$ $\int$)
- Mathematical equations (*inline* and $$block$$)
- Code blocks with various languages
- Lists (ordered, unordered, nested)
- Special symbols (©®™€£¥)

- Links and references
- Text formatting (**bold**, *italic*, `code`)
- Greek letters (α β γ δ ε ζ η θ λ μ π σ τ φ ω Γ Δ Θ Λ Ξ Π Σ Φ Ψ Ω)
- Mathematical operators ($\pm$ $\times$ $\div$ $\surd$ ∞ $\approx$ $\neq$ $\leq$ $\geq$ $\int$ $\sum$ $\prod$ ∂ ∇)
- Complex LaTeX equations with matrices, integrals, partial derivatives

2.10 Credits

- Author: SDNT8810