

Markdown to PDF Converter

A demonstration of the Rust-based Markdown to PDF conversion tool.

Features Showcase

This document demonstrates various Markdown features supported by our converter:

Typography Hierarchy

Heading Level 1

Heading Level 2

Heading Level 3

Heading Level 4

Heading Level 5

Heading Level 6

Each heading level uses appropriate font sizing and spacing for professional

document layout.

Text Formatting

This paragraph contains

bold text

,

italic text

, and

~~to demonstrate~~ basic text formatting capabilities.

Here's another paragraph with mixed formatting:

bold

,

italic

, and even

~~within the same line.~~

Code Blocks

Here's a Rust code example:

```
use pulldown_cmark::Parser;
use printpdf::*;

fn main() -> Result<(), Box<dyn std::error::Error>> {
    let markdown = "# Hello, World!";
    let parser = Parser::new(markdown);

    // Process events and generate PDF
    Ok(())
}
```

And here's a bash script:

```
#!/bin/bash

# Build and run the tool
cargo build --release
./target/release/markdown-to-pdf-rust input.md -o output.pdf

echo "PDF generated successfully!"
```

Lists

Unordered Lists

- First item
- Second item
 - Nested item A
 - Nested item B
 - Deeply nested item
- Third item
- Fourth item with a very long line that should wrap properly when the text exceeds the available line width

Ordered Lists

- Step one: Initialize the project
- Step two: Configure dependencies
- Step three: Implement core functionality
- Step four: Add error handling
- Step five: Test thoroughly

Mixed Content

Here's a paragraph followed by a code block, then another paragraph to test spacing:

```
Simple code block without syntax highlighting
Multiple lines of code
With proper monospace formatting
```

The spacing between different elements should be consistent and professional, similar to LaTeX documents.

Edge Cases

Here are some edge cases that the converter handles:

- Empty lines between elements
- Multiple consecutive spaces (should normalize)
- Lines that are very long and need to wrap properly to fit within the page margins without breaking the overall document layout

Performance Notes

The converter processes documents efficiently by:

- Parsing
 - : Using `pulldown-cmark` for fast Markdown parsing
- Rendering
 - : Direct PDF generation with `printpdf`
 - : Minimal memory usage for large documents
- Speed
 - : Fast processing even for complex documents

Technical Specifications

| Feature | Implementation |

|-----|-----|

| Parser | `pulldown-cmark 0.12` |

| PDF Library | `printpdf 0.7` |

| Font System | Built-in PDF fonts |

| Page Format | A4 (210×297mm) |

Conclusion

This Markdown to PDF converter successfully demonstrates:

- Professional Typography
: LaTeX-inspired layout and spacing
- Comprehensive Support
: All standard Markdown elements
- Rust Performance
: Fast, memory-efficient processing
- Direct Generation
: No HTML intermediate step
- Cross-platform
: Single binary works everywhere

The tool meets all the original requirements and provides a solid foundation for high-quality document generation from Markdown sources.