

# Technical Documentation Example

## Overview

This document demonstrates more advanced features of the Markdown to PDF converter, including complex formatting, code blocks, and technical content.

## System Requirements

The following system requirements must be met:

- Operating System  
: Linux, macOS, or Windows
- Memory  
: At least 4GB RAM
- Disk Space  
: 100MB free space
- Runtime  
: Rust 1.70 or higher

## Installation

### Quick Start

Install using Cargo:

```
git clone https://github.com/example/markdown-to-pdf-rust.git
cd markdown-to-pdf-rust
cargo build --release
```

## Configuration

Create a configuration file:

```
[settings]
font_size = 11
line_height = 1.4
margins = {top = 25, bottom = 25, left = 25, right = 25}

[typography]
heading_font = "Times-Bold"
body_font = "Times-Roman"
code_font = "Courier"
```

## API Reference

### Core Functions

```
render_markdown(input: &str) -> Result<Document, Error>
```

Parses and renders Markdown content.

Parameters:

**input:** The Markdown content as a string

Returns:

**Result<Document, Error>** The rendered document or an error

Example:

```
use markdown_to_pdf::{render_markdown, save_pdf};

fn main() -> Result<(), Box<dyn std::error::Error>> {
    let markdown = r#"
    # Hello World

    This is bold text.
    "#;
```

```

    let document = render_markdown(markdown)?;
    save_pdf(&document, "output.pdf"?);

    Ok(())
}

```

## Error Handling

The library uses  
~~for~~ error handling:

```

match render_markdown(content) {
    Ok(doc) => println!("Success!"),
    Err(e) => eprintln!("Error: {}", e),
}

```

## Performance Considerations

### Benchmarks

Document Size	Processing Time	Memory Usage
Small (1KB)	5ms	2MB
Medium (100KB)	45ms	15MB
Large (1MB)	420ms	125MB

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

| Small (1KB) | 5ms | 2MB |

| Medium (100KB) | 45ms | 15MB |

| Large (1MB) | 420ms | 125MB |

### Optimization Tips

- Batch Processing
  - : Process multiple files in a single run
- Memory Management
  - : Use streaming for large documents

- Font Caching  
: Reuse font resources when possible

## Advanced Features

### Custom Styling

Override default styles:

```
let mut config = Config::default();
config.set_font_size(FontSize::Points(12.0));
config.set_margins(Margins::all(30.0));

let renderer = Renderer::with_config(config);
```

### Plugins

The system supports plugins for extended functionality:

- Syntax Highlighting  
: Color code blocks
- Math Rendering  
: LaTeX-style equations
- Diagrams  
: Mermaid and PlantUML support

## Troubleshooting

### Common Issues

Issue

: PDF generation fails with "Font not found"

#### Solution

: Ensure required fonts are installed on the system.

#### Issue

: Text appears too small

#### Solution

: Adjust the  
~~setting in the~~ configuration.

#### Issue

: Code blocks don't wrap properly

#### Solution

: Use the  
~~option in the~~ configuration.

## Contributing

We welcome contributions! Please see our  
Contributing Guide  
for details.

## Development Setup

```
# Clone the repository
git clone https://github.com/example/markdown-to-pdf-rust.git

# Install dependencies
cd markdown-to-pdf-rust
cargo build

# Run tests
cargo test

# Run with sample
cargo run -- samples/simple.md -o test.pdf
```

## License

This project is licensed under the MIT License. See  
LICENSE  
for details.

This document was generated using the Markdown to PDF tool itself, demonstrating  
its capabilities for technical documentation.