

Contributing to md2pdf

Thank you for your interest in contributing to md2pdf! We welcome contributions from everyone and are grateful for even the smallest of improvements.

Table of Contents

- Code of Conduct
- Getting Started
- Development Setup
- How to Contribute
- Pull Request Process
- Style Guidelines
- Testing
- Documentation
- Community

Code of Conduct

This project adheres to a Code of Conduct that all contributors are expected to follow. Please read `CODE_OF_CONDUCT.md` before contributing.

Getting Started

1. **Fork the repository** on GitHub
2. **Clone your fork** locally
3. **Create a branch** for your changes
4. **Make your changes** and commit them
5. **Push to your fork** and submit a pull request

Development Setup

Prerequisites

- Python 3.8 or higher
- Git
- System dependencies for WeasyPrint:
- **Ubuntu/Debian:** `sudo apt-get install libcairo2-dev libpango1.0-dev libgdk-pixbuf2.0-dev`
- **macOS:** `brew install cairo pango gdk-pixbuf libffi`
- **Windows:** Install GTK3 runtime

Setting Up Your Environment

```
# Clone your fork
git clone https://github.com/YOUR_USERNAME/md2pdf.git
cd md2pdf

# Create a virtual environment
python3 -m venv venv
source venv/bin/activate # On Windows: venv\Scripts\activate

# Install the package in development mode
pip install -e ".[dev]"

# Install pre-commit hooks
pre-commit install

# Run tests to verify setup
pytest
```

Development Dependencies

The `[dev]` extra includes:

- `pytest` - Testing framework
- `pytest-cov` - Coverage reporting
- `black` - Code formatter
- `isort` - Import sorter
- `flake8` - Linter
- `mypy` - Type checker

- bandit - Security scanner
- pre-commit - Git hook framework

How to Contribute

Reporting Bugs

Before creating bug reports, please check existing issues to avoid duplicates. When creating a bug report, please include:

- A clear and descriptive title
- Steps to reproduce the issue
- Expected behavior
- Actual behavior
- System information (OS, Python version)
- Relevant logs or error messages
- Sample files that reproduce the issue (if applicable)

Suggesting Enhancements

Enhancement suggestions are welcome! Please provide:

- A clear and descriptive title
- Detailed description of the proposed feature
- Use cases and examples
- Potential implementation approach (if you have ideas)
- Any relevant mockups or diagrams

Your First Code Contribution

Unsure where to begin? Look for issues labeled:

- good first issue - Simple issues for newcomers
- help wanted - Issues where we need community help
- documentation - Documentation improvements

Pull Requests

1. **Branch naming:** Use descriptive names like `feature/add-latex-support` or `fix/unicode-handling`
2. **Commits:**

3. Write clear, concise commit messages
4. Use conventional commits format when possible: type(scope): description
5. Types: feat , fix , docs , style , refactor , test , chore
6. **Code changes:**
7. Include tests for new functionality
8. Update documentation as needed
9. Ensure all tests pass
10. Follow the style guidelines

Pull Request Process

1. **Before submitting:**

```
```bash
Format code
black src/ tests/
isort src/ tests/

Run linters
flake8 src/ tests/
mypy src/
bandit -r src/

Run tests
pytest --cov=md2pdf
```
```

1. **PR Description:**

2. Reference any related issues
3. Describe what changes you've made
4. Include screenshots for UI changes
5. List any breaking changes

6. **Review process:**

7. PRs require at least one review before merging
8. Address review feedback promptly
9. Keep PRs focused and reasonably sized

10. **After merge:**

11. Delete your feature branch

12. Pull the latest changes to your local main branch

Style Guidelines

Python Style

We use Black for code formatting with:

- Line length: 88 characters
- Python 3.8+ syntax

Code Organization

- Keep modules focused and cohesive
- Use descriptive variable and function names
- Add type hints to function signatures
- Write docstrings for all public functions and classes

Documentation Style

- Use Markdown for all documentation
- Include code examples where helpful
- Keep language clear and concise
- Update the README for user-facing changes

Testing

Running Tests

```
# Run all tests
pytest

# Run with coverage
pytest --cov=md2pdf --cov-report=html

# Run specific test file
pytest tests/test_converter.py

# Run tests matching pattern
pytest -k "test_style"
```

Writing Tests

- Place tests in the `tests/` directory
- Mirror the source code structure
- Use descriptive test names
- Include both positive and negative test cases
- Use fixtures for common test data

Example test:

```
def test_pdf_conversion_with_custom_style(tmp_path):
    """Test PDF conversion with a custom style."""
    # Arrange
    input_file = tmp_path / "test.md"
    input_file.write_text("# Test Document")

    # Act
    converter = MD2PDFConverter(
        input_path=str(input_file),
        style="technical",
        theme="github"
    )
    output = converter.convert()

    # Assert
    assert output.exists()
    assert output.suffix == ".pdf"
```

Documentation

Docstring Format

We use Google-style docstrings:

```
def convert_markdown(input_path: str, style: str = "default") -> Path:
    """Convert a Markdown file to PDF.

    Args:
        input_path: Path to the input Markdown file.
        style: Name of the style template to use.

    Returns:
        Path to the generated PDF file.

    Raises:
        FileNotFoundError: If the input file doesn't exist.
        ValueError: If the style is not recognized.
    """
```

Updating Documentation

- Update the README for significant changes
- Add entries to CHANGELOG.md following Keep a Changelog
- Update API documentation for new features
- Include examples for new functionality

Community

Getting Help

- Check the documentation
- Search existing issues
- Join our discussions

Recognition

Contributors are recognized in the AUTHORS file and in release notes. We appreciate all contributions, no matter how small!

License

By contributing to md2pdf, you agree that your contributions will be licensed under the MIT License.

Thank you for contributing to md2pdf! 🍷