WeasyPrint Multi-Page Document

Introduction

This document demonstrates WeasyPrint's ability to create professional multi-page PDFs with automatic page numbering, headers, and footers. The @page CSS rule allows precise control over page layout.

Page Layout Features

WeasyPrint supports the CSS Paged Media Module, which includes:

- **@page rule:** Define page dimensions and margins
- **@top-*/@bottom-* regions:** Add headers and footers
- **page counters:** Automatic page numbering
- **page-break properties:** Control pagination

Sample Data Table

Feature	Description	Supported
CSS3 Selectors	Modern CSS selector support	✓ Yes
Flexbox	Flexible box layout	✓ Yes
CSS Grid	Two-dimensional layouts	⚠ Partial
Web Fonts	@font-face support	✓ Yes
SVG Images	Scalable vector graphics	✓ Yes

Page 2: Additional Content

This content appears on the second page due to the page-break-before CSS property. WeasyPrint handles page breaks automatically when content exceeds page height, but you can also control breaks explicitly.

Technical Specifications

WeasyPrint is built on several key technologies:

- **Pango:** Text layout and rendering
- **Cairo:** 2D graphics library for PDF generation
- **cssselect2:** CSS selector implementation
- **tinycss2:** CSS parser

Code Example

Basic usage of WeasyPrint in Python:

```
from weasyprint import HTML

# Convert HTML to PDF
HTML('document.html').write_pdf('output.pdf')

# Or from a string
html_string = '<h1>Hello PDF!</h1>'
HTML(string=html_string).write_pdf('output.pdf')
```

Page 3: Conclusion

WeasyPrint provides a powerful and flexible solution for converting HTML and CSS to high-quality PDF documents. Its support for modern CSS features and print-specific layouts makes it ideal for generating reports, invoices, and other professional documents.

Resources

- Documentation: https://doc.courtbouillon.org/weasyprint/
- GitHub: https://github.com/Kozea/WeasyPrint
- CSS Paged Media: https://www.w3.org/TR/css-page-3/