

Web-to-PDF

Convert Markdown/HTML/URL to professional PDF documents, ideal for technical documentation, reports, and archiving.

Theme Styles

This skill provides 5 carefully designed themes:

| Theme | Style Description |
|----------|---|
| default | Apple design style, clean and elegant |
| github | GitHub README style, clear and readable |
| academic | Academic paper style, two-column layout |
| sketch | Hand-drawn note style, personalized |
| magazine | Magazine layout style, drop cap effect |

Code Highlighting Examples

JavaScript

```
// Convert Markdown to PDF
import { toPdf } from './converter.js';

async function convert() {
  const result = await toPdf({
    inputPath: "README.md",
    outputPath: "output.pdf",
```

```
    style: "github",
    options: {
      format: "markdown"
    }
  });

  console.log(`PDF generated: ${result.pdfPath}`);
}
```

Python

```
def fibonacci(n: int) -> list[int]:
    """Generate Fibonacci sequence"""
    if n <= 0:
        return []
    if n == 1:
        return [0]

    sequence = [0, 1]
    while len(sequence) < n:
        sequence.append(sequence[-1] + sequence[-2])
    return sequence

# Example usage
print(fibonacci(10)) # [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Table Examples

CLI Parameters Overview

| Parameter | Description | Default |
|---------------------------|-------------------------------|-----------------------|
| <code>--input</code> | Input file path (md/html) | Required |
| <code>--output</code> | Output PDF path | Required |
| <code>--style</code> | Theme style | <code>default</code> |
| <code>--format</code> | Input format | <code>markdown</code> |
| <code>--no-html</code> | Don't keep intermediate HTML | <code>false</code> |
| <code>--no-mermaid</code> | Disable Mermaid pre-rendering | <code>false</code> |

Blockquote Styles

Tip: This skill supports Mermaid diagram pre-rendering.
Just use standard mermaid code blocks in your Markdown.

List Styles

Unordered List

- Supports Markdown syntax highlighting
- Automatically handles Mermaid diagrams
- Multiple preset themes available
- Print-ready output quality

Ordered List

1. Install dependencies: `npm install`
 2. Run Playwright: `pnpm exec playwright install chromium`
 3. Execute conversion: `node scripts/converter.js --input doc.md --output output.pdf`
-

Footnotes and Emphasis

This is a paragraph with **bold**, *italic*, `inline code`, and [links](#).

For scenarios requiring math formulas, you can use LaTeX syntax^[1].

Separation and Typography

This skill focuses on **output quality**, with each theme carefully crafted:

- Clear font hierarchy
- Comfortable line height and spacing
- Code block syntax highlighting
- Beautiful tables and blockquotes

High-quality PDF output, starting from Markdown.

1. Additional configuration required ↩