Docco Feature Showcase

Markdown & PDF Capabilities

2025-10-23

Engineering Team

Table of Contents

1 Introduction

This document showcases Docco's **formatting capabilities** and **Markdown syntax**.

Docco converts Markdown files with YAML frontmatter and external CSS into professional A4 PDFs.

1.1 Basic Usage

Generate a PDF with the CLI:

```
docco build document.md style.css --output example.pdf
```

This creates:

- `example.pdf` - Your rendered PDF
- `debug.html` - HTML preview for troubleshooting

2 Text Formatting

- **Bold text** with `**bold**`
- *Italic text* with `*italic*`
- `Inline code` with backticks

2.1 Lists

Unordered list:

- First item
- Second item
  - Nested item
  - Another nested item

Ordered list:

1. First step
2. Second step
3. Third step

2.2 Code Blocks

Python example:

```python
def greet(name):
    return f"Hello, {name}!"

print(greet("World"))
```

Shell command:

```shell
docco build document.md style.css --output example.pdf
```

2.3 Tables

Simple table:

Feature	Status
Markdown	✓
CSS	✓
PDF Output	✓

Table with alignment:

Left Aligned	Center	Right Aligned
Text	123	45.67
More text	456	89.01

2.4 Links & Quotes

External link: [Markdown Guide](#)
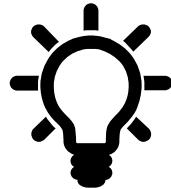
> "Simplicity is the ultimate sophistication." — Leonardo da Vinci

2.5 Images

Docco supports standard HTML `<img>` tags for inline images.
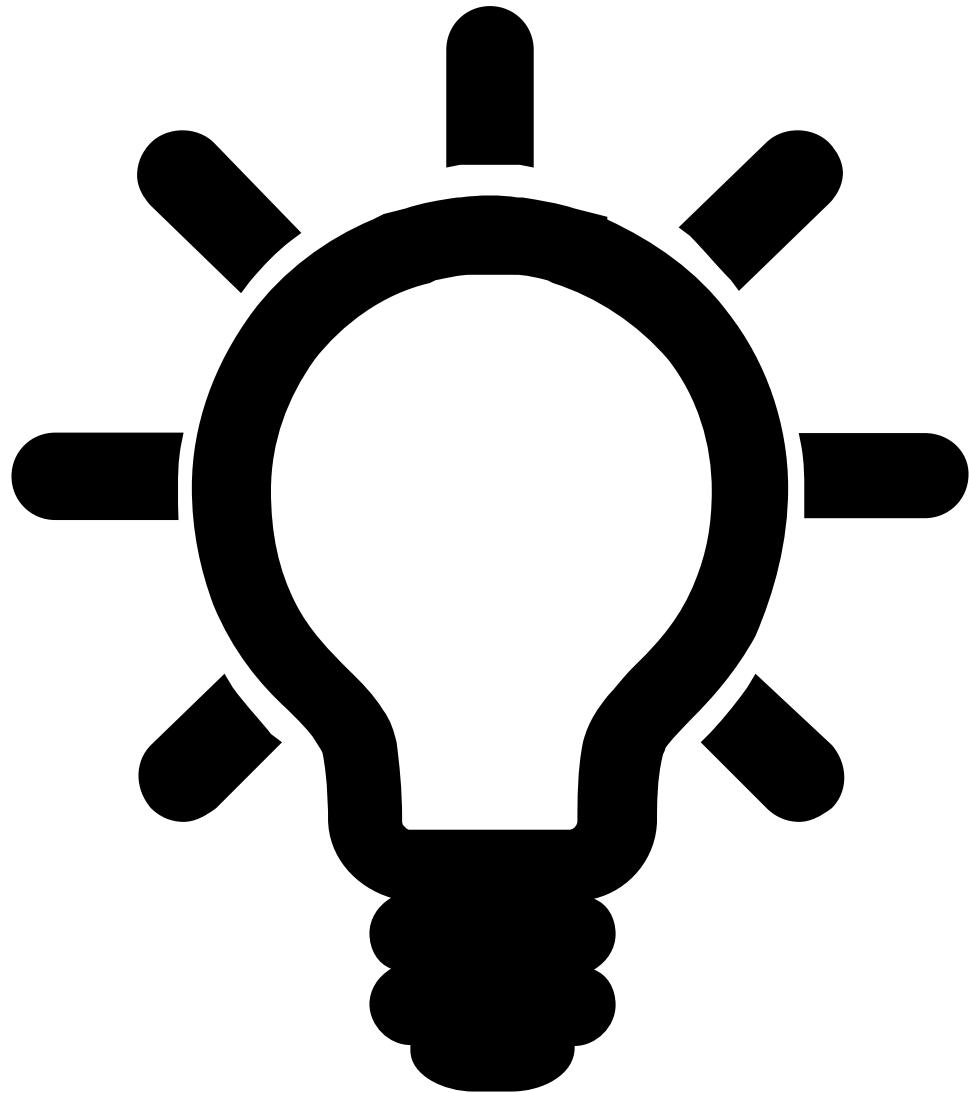
2.5.1 Inline Styled Image

This image uses inline CSS styling:

2.5.2 Class-Based Image

This image uses a CSS class for styling:

2.5.3 Images with Captions

Images with `alt` text are automatically wrapped in `<figure>` elements with `<figcaption>`:
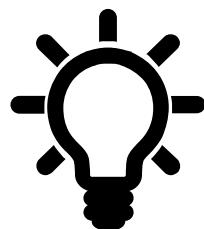
Figure 1: Innovation concept illustration

Images are resolved relative to the markdown file location.

2.6 Horizontal Rule

2.7 Inline Directives

Docco supports **inline directives** that let you create reusable components without writing HTML directly in your markdown.

Using tables with icons creates effective callout boxes. This technique is useful for highlighting important information or warnings in your documents.

Inlines are defined as markdown templates in the `inlines/` folder and invoked using HTML comment syntax.

2.8 Horizontal Rule

---

3 Landscape Orientation

This section uses **landscape orientation** with the `<!-- landscape -->` directive.

Landscape is useful for wide tables or content that benefits from horizontal space.

3.1 Wide Data Table

Month	Product A	Product B	Product C	Product D	Product E	Total Revenue	Growth %
Jan	$12,500	$8,300	$15,600	$9,200	$11,400	$57,000	-
Feb	$13,200	$9,100	$16,400	$9,800	$12,100	$60,600	+6.3%
Mar	$14,800	$10,500	$18,200	$11,300	$13,900	$68,700	+13.4%

4 Back to Portrait

This section returns to **portrait orientation** using the `<!-- portrait -->` directive.

Regular portrait content continues here.

A Appendix: Directives

This section uses the `<!-- addendum -->` directive, which creates an **appendix** with letter-based numbering (A, B, C...).

B Supported Directives

Directive	Effect
`<!-- ___DOCCO_TOC_PLACEHOLDER___ -->`	Insert table of contents at this location
`<!-- ___DOCCO_PAGEBREAK_PLACEHOLDER___ -->`	Insert a page break
`<!-- landscape -->`	Next section uses landscape orientation
`<!-- portrait -->`	Next section uses portrait orientation
`<!-- addendum -->`	Next section uses appendix numbering (A, B, C...)
`<!-- inline: name args -->...<!-- / inline -->`	Insert inline directive with arguments
`<!-- lang:XX -->...<!-- /lang -->`	Content only appears in language XX

C Appendix: YAML Frontmatter

Supported frontmatter fields at the top of the document:

Field	Type	Default	Description
`languages`	String	None	Space-separated language codes (e.g., `EN DE NL`) for multilingual PDFs
`no_headers_first_page`	Boolean	`true`	Skip headers/footers on first page