Output Formats

This example demonstrates multi-format output: PDF, SVG, and PNG.

Feature Configuration

In your `Cargo.toml`, configure output formats using features:

```toml
[dependencies]
# Default: PDF only
typst-bake = "0.1"

# Add SVG support
typst-bake = { version = "0.1", features = ["svg"] }

# Add PNG support
typst-bake = { version = "0.1", features = ["png"] }

# All formats (pdf + svg + png)
typst-bake = { version = "0.1", features = ["full"] }

# SVG only (disable default PDF)
typst-bake = { version = "0.1", default-features = false, features = ["svg"] }
```

Output Behavior

PDF outputs all pages as a single file:

- `output.pdf` (contains all pages)

SVG and **PNG** return a Vec with one item per page:

- Save each page individually (e.g., `output_1.svg`, `output_1.png`, ...)

PNG Resolution

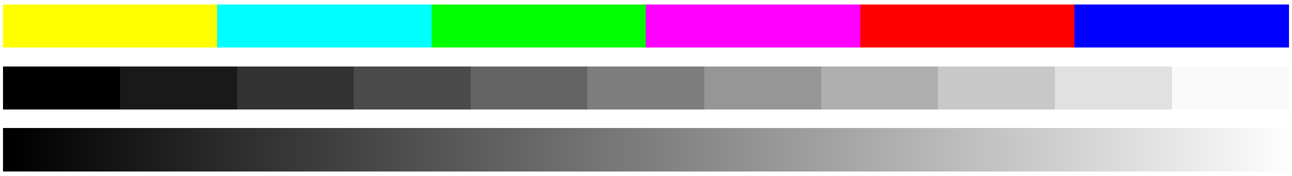
PNG requires a DPI (dots per inch) setting:

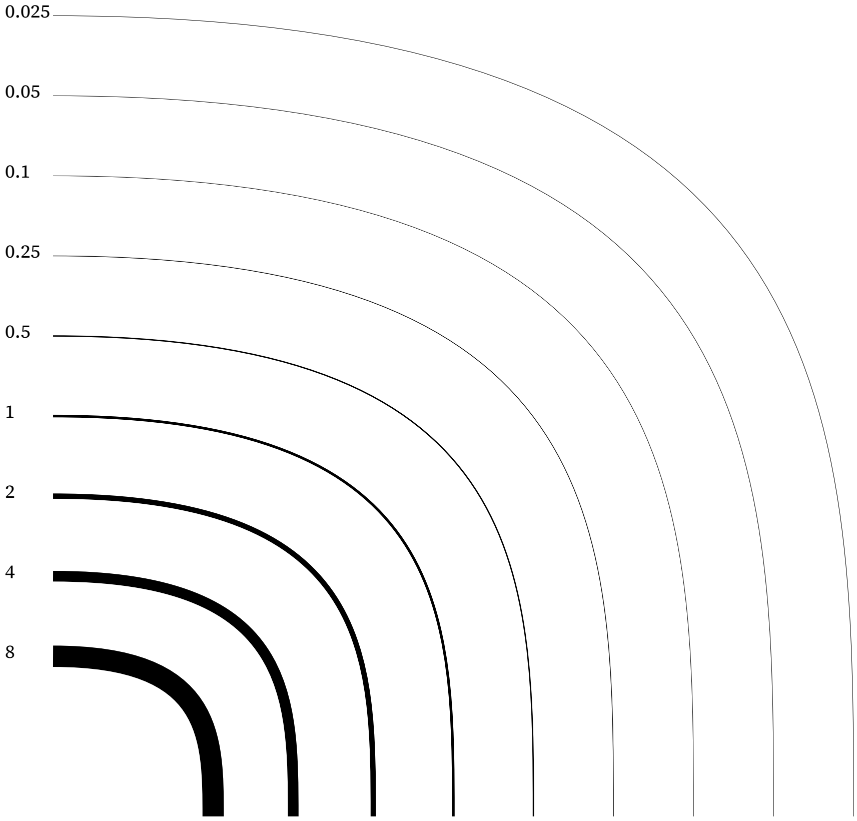
```rust
// 72 DPI (standard)
let pngs = doc.to_png(72.0)?;

// 144 DPI (2x)
let pngs = doc.to_png(144.0)?;

// 300 DPI (print quality)
let pngs = doc.to_png(300.0)?;
```

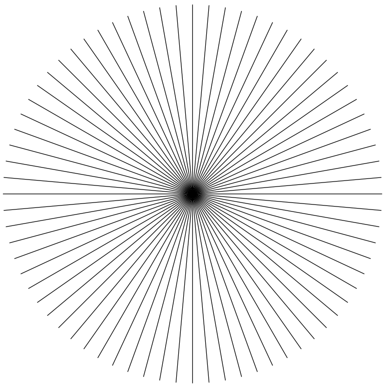
Note

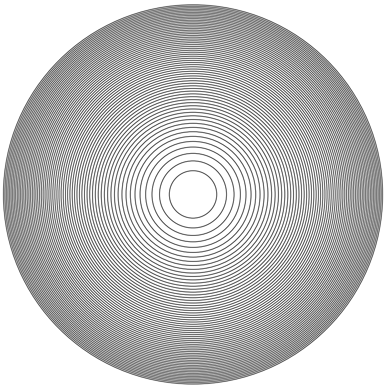
Page 2 includes test patterns to verify multi-page behavior and evaluate PDF/SVG/PNG rendering differences.

Rendering Test Patterns









2pt: The quick brown fox jumps over the lazy dog

3pt: The quick brown fox jumps over the lazy dog

4pt: The quick brown fox jumps over the lazy dog

5pt: The quick brown fox jumps over the lazy dog

6pt: The quick brown fox jumps over the lazy dog

7pt: The quick brown fox jumps over the lazy dog

8pt: The quick brown fox jumps over the lazy dog

9pt: The quick brown fox jumps over the lazy dog

10pt: The quick brown fox jumps over the lazy dog

12pt: The quick brown fox jumps over the lazy dog

14pt: The quick brown fox jumps over the lazy dog

18pt: The quick brown fox jumps over the lazy dog

24pt: The quick brown fox jumps over the

36pt: The quick brown fox