



TYPOGRAPHY · CALLOUTS · TABLES · FIGURES · CODE

The Style Codex

Theme Reference & Template

A living golden sample of every element
the Ubunatic book theme can render

Uwe Jügel · Ubunatic Coding

2026 Edition · rendered by mdBook

The Style Codex — Theme

Reference & Template



THIS IS NOT A BOOK. It is the reference sheet and test fixture for the fancy mdBook theme that the Ubunatic library renders through. There is no argument here to follow and no reason to read it cover to cover. Look up the class you need in the index below, copy the snippet, close the tab.

Each chapter isolates one family of devices and shows both the rendered result and the source that produced it. The theme itself is verified by `make styles-golden`, which diffs the rendered DOM and the stylesheet against the goldens in `styles/golden/`.

THE CLASS INDEX

Every opt-in class of the theme, authored as raw HTML inside Markdown — a plain Markdown file stays plain. This table is the contract: `make styles-golden` fails if a class listed here is missing from `styles/css/fancy.css` or is never rendered by any chapter.

CLASS	ELEMENT	EFFECT	CHAPTER
<code>.lead</code>	<code><p></code>	larger italic opening paragraph	Typography
<code>.kicker</code>	<code></code>	small-caps section label	Typography

CLASS	ELEMENT	EFFECT	CHAPTER
<code>.initial</code>	<code><p></code>	manual drop cap mid-book	Ornaments
<code>.pullquote</code>	<code><p></code>	centered accent quote between hairlines	Typography
<code>.epigraph</code>	<code><div></code>	chapter- opening quotation with attribution	Ornaments
<code>.attribution</code>	<code></code>	the byline inside an epigraph	Ornaments
<code>.callout</code>	<code><div></code>	note, tip, warn, or danger plate	Callouts
<code>.note</code>	<code><div class="callout"></code>	blue plate, quill icon	Callouts
<code>.tip</code>	<code><div class="callout"></code>	green plate, candle icon	Callouts
<code>.warn</code>	<code><div class="callout"></code>	amber plate, warning icon	Callouts
<code>.danger</code>	<code><div class="callout"></code>	crimson plate, flame icon	Callouts
<code>.aside-note</code>	<code><div></code>	note floated into the right margin	Callouts

CLASS	ELEMENT	EFFECT	CHAPTER
<code>.timeline</code>	<code></code>	chronology on a golden spine	Ornaments
<code>.tag</code>	<code></code>	small-caps pill; add a color class	Ornaments
<code>.green</code>	<code></code>	green pill variant	Ornaments
<code>.blue</code>	<code></code>	blue pill variant	Ornaments
<code>.red</code>	<code></code>	red pill variant	Ornaments
<code>.twocol</code>	<code><div></code>	two-column prose with a hairline rule	Ornaments
<code>.figrid</code>	<code><div></code>	grid of figure plates	Figures
<code>.compact</code>	<code><div></code>	dense table: smaller type, tighter cells	Tables
<code>.finis</code>	<code><p></code>	chapter-end ornament	Ornaments

Devices that need no class at all: drop caps (automatic on the first paragraph after a chapter title), fleuron rules (`---`), blockquotes, footnotes, `<kbd>` keys, `<figure>` / `<figcaption>` , tables, code blocks, and Mermaid diagrams.

COPY THIS TO START

Paste this into a fresh chapter and delete what you do not need:

```
# Chapter Title
```

The first paragraph gets an automatic drop cap.

```
<span class="kicker">a small-caps label</span>
```

```
<p class="lead">A larger italic opening paragraph.</p>
```

```
<div class="callout warn">
```

Warning – Blank lines around the body are required.

```
</div>
```

```
<div class="aside-note">
```

```
<span class="kicker">in the margin</span>
```

A note floated into the right margin.

```
</div>
```

```
| Column | Column |
```

```
|---|---|
```

```
| cell | cell |
```

```
<figure>
```

```
<svg viewBox="0 0 200 80" width="200" role="img" aria-label="Plate"></svg>
```

```
<figcaption>Caption set in italics.</figcaption>
```

```
</figure>
```

```
<p class="finis">🐼 🐼 🐼</p>
```

For a whole book rather than a chapter, copy the skeleton instead:

```
cp -r styles/template/ newbook/
mdbook build newbook
```

VERIFYING THE THEME

```
make styles-golden          # diff rendered DOM + stylesheet
against goldens
make styles-golden-update   # regenerate goldens – deliberately,
never to go green
```

`make preflight` runs the check. See the golden-change playbook in the main book for the update discipline; eyeballing the codex in the light and coal themes is a supplement to the golden check, not a substitute for it.

THE CHAPTERS

- **Typography & Drop Caps** — headings, drop caps, pull quotes, footnotes, ornaments.
- **Ornaments & Layout** — epigraphs, highlights, tags, timelines, two-column prose.
- **Callouts & Asides** — note, tip, warning, and danger plates; margin asides.
- **Tables & Data** — framed, striped tables; the compact ledger.
- **Figures & Diagrams** — captions, inline SVG, figure grids, and Mermaid plates.
- **Code & Terminals** — code blocks and keyboard keys inside serif prose.

Typography & Drop Caps

Typography is the first thing a reader feels and the last thing they notice. This chapter opens with a drop cap, flows through small-caps subheadings, and closes with an ornamental rule. Every specimen below is followed by the source that produced it.

THE ELEMENTS OF THE PAGE

```
<span class="kicker">the elements of the page</span>
```

HEADINGS IN SMALL CAPS

Second-level headings render in gilded small caps. Third-level headings lean into italics:

A Third-Level Heading, Reclining

Body text uses a classic serif stack — Iowan Old Style, Palatino, Charter, Georgia — with generous 1.75 line height and hyphenation enabled. The stack, size, and leading are spec-driven: they live in `spec/config.yaml` under `books.styles`, not in the theme CSS.

The drop cap is automatic on the first paragraph after a chapter title — no class needed:

```
# Chapter Title
```

The first paragraph gets the drop cap.

EMPHASIS & INLINE STYLES

Prose supports *italics for voice*, **bold for weight**, and `code for machinery`. Keyboard chords render as keys: press `Ctrl` + `Shift` + `P` to summon the palette.

```
*italics for voice*, **bold for weight**, `code for machinery`,  
and <kbd>Ctrl</kbd> + <kbd>Shift</kbd> + <kbd>P</kbd> for  
chords.
```

THE LEAD PARAGRAPH

A lead paragraph states the promise of the chapter in larger italics. Use one, directly under the title.

```
<p class="lead">A lead paragraph states the promise of the  
chapter.</p>
```

PULL QUOTES

Spec just enough. Verify first. Iterate. Succeed.

A pull quote interrupts the column with the book's accent color, framed by hairlines above and below.

```
<p class="pullquote">Spec just enough. Verify first. Iterate.  
Succeed.</p>
```


BLOCKQUOTES

“

A blockquote carries a hanging quotation mark in the margin and a soft golden wash behind it. Use it for quoted voices, not for warnings — warnings get callout plates.

> A blockquote carries a hanging quotation mark in the margin.

FOOTNOTES

Footnotes step politely out of the text¹ and gather beneath a golden hairline at the foot of the page². The reference is a gold superscript; the definition is set smaller, with tighter leading.

Footnotes step politely out of the text^[^marker].

[^marker]: The definition, written at the very end of the chapter file.



The ornament above is a fleuron (🌀), rendered by CSS on every horizontal rule. It marks a scene change within a chapter, and is authored as a plain `---`.



1. Like this. Clicking the number jumps down; the label links back up. ↩

2. mdBook renders definitions where they are written, so keep them at the very end of the chapter file. ↩

Ornaments & Layout

*Ornament is not decoration added to a page; it is the page
confessing what it cares about.*

— THE STYLE CODEx, ON ITSELF

```
<div class="epigraph">
```

The quotation, set in italics above the chapter.

```
<span class="attribution">Who said it</span>
```

```
</div>
```

Every device in this chapter is opt-in: a plain Markdown file stays plain, and a decorated page declares its intent in raw HTML. This opening paragraph summons a drop cap with `class="initial"` — the automatic drop cap only follows a chapter title, so use this one mid-book.

```
<p class="initial">A manual drop cap, for paragraphs that do not  
follow a title.</p>
```

HIGHLIGHTS & TAGS

Use a gilded highlight when the reader must not skim past a phrase — sparingly, or the gold turns to noise. Tags mark status inline without breaking the sentence: this chapter is CODEx, its theme is VERIFIED, the print run is SCHEDULED, and nothing here is BLOCKED.

```
<mark>a gilded highlight</mark>
```

```
<span class="tag">codex</span>
```

```
<span class="tag green">verified</span>
```

```
<span class="tag blue">scheduled</span>
```

```
<span class="tag red">blocked</span>
```

A TIMELINE

A timeline sets a chronology on a golden spine, with a bold lead-in per entry:

- 
- 2022** — the first book is bound: plain styles, plain pages, offline from day one.
 - 2024** — the slide compiler is written in Go; decks join the library.
 - 2026** — the Style Codex opens, and the spec system takes over fonts and colors.

```
<ul class="timeline">  
<li><strong>2022</strong> – what happened that year.</li>  
<li><strong>2024</strong> – and then this.</li>  
</ul>
```

TWO COLUMNS

Long front matter, indexes, and glossaries read better in two columns: the eye travels a shorter line, and dense reference prose stops feeling like a wall. The column rule is a hairline in the accent wash, quiet enough to ignore.

On narrow screens the columns collapse back into one, so this device costs nothing on a phone. In print both columns are kept, which is where they earn their keep — paper is wide and patience is short.

```
<div class="twocol">
```

First paragraph. Blank lines keep the Markdown inside working.

Second paragraph.

```
</div>
```

THE END OF A CHAPTER

A chapter that has said everything should say so. The *finis* mark below closes the chapter the way a colophon closes a book — reserve it for true endings, not section breaks (those get the fleuron rule).



```
<p class="finis">☛ ☛ ☛</p>
```

Callouts & Asides

Callouts are the marginalia of this codex: small plates that interrupt the prose to note, advise, warn, or forbid. They are opt-in HTML blocks so plain Markdown stays plain.

THE FOUR PLATES



NOTE — A note records a fact the reader should carry forward. It uses the scribe's quill and a cool blue wash.



TIP — A tip lights a shortcut. Green wash, candle icon, and ideally one sentence long.



WARNING — A warning marks a place where agents have burned themselves before. Amber wash. If you write more than three sentences here, it should be a chapter instead.



DANGER — A danger plate is reserved for the irreversible: deleted data, force-pushes, production credentials. Crimson wash, a thicker rule, an open flame. If the reader can undo it, it is only a warning.

MARGIN ASIDES

An aside (`<div class="aside-note">`) is for commentary that enriches but never interrupts: an etymology, a cross-reference, a wry remark. The main argument must survive with the margin torn off. This paragraph flows around the note the way body text shares a page with marginalia in a medieval manuscript — and in print, the float is preserved.

IN THE MARGIN

An aside floats into the right margin like a scribbled note. Below 900px it rejoins the main column.

```
<div class="aside-note">
  <span class="kicker">in the margin</span>
  The note itself, one or two sentences.
</div>
```

AUTHORING SYNTAX

Callouts are authored as raw HTML with blank lines around the Markdown body, so mdBook still renders the inner content. The second class picks the plate: `note` , `tip` , `warn` , or `danger` .

```
<div class="callout warn">

Warning — Snap-confined Chromium cannot print to `/tmp`.

</div>
```



NOTE — Markdown inside the plate keeps working: *italics*, `code` , and even lists:

- first
- second

Tables & Data

Tables in the codex are framed plates: a golden border, small-caps headers, and alternating row washes. They should read like a ledger, not a spreadsheet. Plain Markdown tables need no class at all.

A LEDGER OF BOOKS

BOOK	DIRECTORY	PURPOSE
Agentic Software Development	src/	The main playbook
Emojig Chronology	emojig/	A Zig endeavour, week by week
The Style Codex	styles/	Theme reference, template, and fixture

```
| Book | Directory | Purpose |
|---|---|---|
| Agentic Software Development | `src/` | The main playbook |
```

A LEDGER OF TOOLS

TOOL	VERIFIED BY	FAILURE MODE
mdbook	canary	missing binary, theme drift
chromium	canary	snap confinement, no /tmp access

TOOL	VERIFIED BY	FAILURE MODE
go	preflight	formatting, golden mismatch
mermaid.min.js	build	stale local asset

THE COMPACT LEDGER

Wrap a table in `<div class="compact">` for dense reference data — smaller type, tighter cells. The class index on the [front page](#) is set this way:

DEVICE	CLASS	VERIFIED BY
callout plate	<code>.callout</code>	DOM census
margin aside	<code>.aside-note</code>	DOM census
timeline	<code>.timeline</code>	DOM census
tag pill	<code>.tag</code>	DOM census
two columns	<code>.twocol</code>	DOM census
figure grid	<code>.figrid</code>	DOM census
compact table	<code>.compact</code>	this very ledger

```
<div class="compact">
```

```
| Device | Class | Verified by |
|---|---|---|
| callout plate | .callout | DOM census |
```

```
</div>
```



TIP — Keep tables to three or four columns. If a row needs a paragraph, it wants to be a section.

Figures & Diagrams

Figures are centered plates with italic, em-dash captions. The codex supports two kinds: inline SVG (offline, hand-authored) and Mermaid diagrams rendered by the local `mermaid.min.js` asset.

AN INLINE SVG PLATE



— *The verification funnel: nothing is built before the environment is proven.*

```
<figure>
<svg viewBox="0 0 420 120" width="420" role="img" aria-
label="Verification funnel">
  <!-- hand-authored, offline, diffable -->
</svg>
<figcaption>The caption, set in italics after an em dash.
</figcaption>
</figure>
```

PLATES SIDE BY SIDE

Wrap figures in `<div class="figrid">` to set them as a grid; on narrow pages the plates stack:



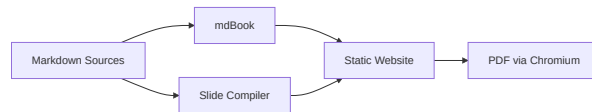
— The seal, obverse.



— The seal, reverse.

```
<div class="figrid">
<figure>...</figure>
<figure>...</figure>
</div>
```

A MERMAID PLATE



A Mermaid diagram is an ordinary fenced block with the `mermaid` language tag:

```
```mermaid
graph LR
 A[Markdown Sources] --> B[mdBook]
 B --> D[Static Website]
 C[Slide Compiler] --> D
 D --> E[PDF via Chromium]
```



**NOTE** — Mermaid renders client-side from the vendored `assets/mermaid.min.js`. No CDN is ever contacted; the codex works on a plane.

# Code & Terminals

---

Code is the one place where the codex abandons its serifs. Blocks keep mdBook's monospace machinery and syntax highlighting; the surrounding prose stays bookish, which makes the contrast deliberate and pleasant. No class is needed anywhere in this chapter — fenced blocks are plain Markdown.

## A Go BLOCK

```
// canary probes the environment before any feature work begins.
func canary() error {
 for _, tool := range []string{"mdbook", "chromium", "go"} {
 if _, err := exec.LookPath(tool); err != nil {
 return fmt.Errorf("missing tool %q: %w", tool, err)
 }
 }
 return nil
}
```

```
```go
func canary() error {
    return nil
}
```
```

# A BASH BLOCK

```
set -euo pipefail
```

```
make canary # prove the environment
make preflight # prove the code
make build # only now, build the site
```

```
```bash
make canary      # prove the environment
```
```

## INLINE MACHINERY

Inline code like `make preflight` sits inside serif prose without breaking the line rhythm. Keyboard keys such as `q` and `Ctrl+C` get raised keycaps.

Inline code like ``make preflight``, and keys such as `<kbd>q</kbd>` and `<kbd>Ctrl</kbd>+<kbd>C</kbd>`.

## NESTING A FENCE INSIDE A FENCE

To show a fenced block as source — as this chapter does throughout — wrap it in a longer fence:

```
```markdown
```bash
make preflight
```
```
```



**WARNING** — Never paste terminal screenshots into the codex. Paste the

text; the theme will dress it.