

[Image: TileDown 0.4.1 ships static code color]

# TileDown 0.4.1 ships static code color

TileDown 0.4.1 is about making static pages more legible without changing the core bargain: build once, ship files.

## Static syntax highlighting

Fenced code blocks now render through a lexical highlighter during the build. The output is ordinary HTML spans styled by the shared `styles.css`; there is no Prism, Highlight.js, or other runtime highlighter on the reader path.

```
struct BuildStatus {
    let version = "0.4.1"
    let staticHighlighting = true
    let runtimeHighlighter = false
}
```

```
{
  "languages": 19,
  "runtime": false,
  "checkedByPlaywright": true
}
```

```
swift run tiledown build-site content/ dist/
swift run tiledown doctor --publish content/
```

The supported language profiles cover Bash, C, C++, C#, CSS, Go, HTML, Java, JavaScript, JSON, Kotlin, Python, Ruby, Rust, SQL, Swift, TypeScript, XML, and YAML. Unsupported fences still render as escaped code, so authoring remains safe.

## Tags got safer

Generated tag pages now keep the tag landing page out of the top navigation unless the site authors it there, and the Clear chip returns to the all-articles tag page. Clearing a tag should show every article again, not an empty archive.

## The example grew teeth

The everything example now includes a Source Code page that exercises all supported language profiles. The browser checks assert that token spans are present and styled, so this feature is covered as generated site behavior, not just as a renderer unit test.

This public site carries the same expectation: source blocks are colored at build time, tag clearing is checked through browser navigation, and the WebAssembly playground pages remain static assets copied through the website build.

TileDown 0.4.1 is available from the [GitHub release](#).