⚒ TERMINAL BRAND KIT

LazyForge

Forge database schemas without leaving your terminal. A relaxed anvil, two-tone ANSI art, and drop-in Dart — prints on startup in any terminal.

01 STARTUP BANNER stacked · two-tone · 42 cols

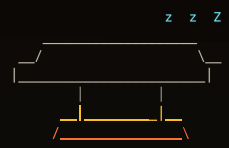
lazyforge — zsh — 88×24

```
$ lazyforge
```

```
forge database schemas without leaving your terminal
⚒ v0.1.0 · lazy by design
```

02 MASCOT & COMPACT MARKS for prompts, help headers, narrow terminals

SLEEPY ANVIL



ONE-LINE MARKS

```
⚒ LazyForge

── LazyForge ──

[⚒ lazyforge] schema.dart

▲ LazyForge · v0.1.0
```

03 DART — READY TO PASTE pass color:false for plain terminals

lib/banner.dart — copy

```dart
import 'dart:io';

/// LazyForge — terminal startup banner.
/// ANSI colors on by default; pass color:false for plain terminals.
class Banner {
  static const _reset = '\x1B[0m';
  static const _bold  = '\x1B[1m';
  static const _lazy  = '\x1B[38;5;80m';   // soft cyan  → "Lazy"
  static const _forge = '\x1B[38;5;214m';  // amber      → "Forge"
  static const _dim   = '\x1B[38;5;244m';  // grey       → tagline
  static const _ember = '\x1B[38;5;208m';  // orange     → accent

  static const _lazyArt = [
    r'',
    r'',
    r'',
    r'',
    r'',
    r'',
  ];

  static const _forgeArt = [
    r'',
    r'',
    r'',
    r'',
    r'',
    r'',
  ];

  static void show({bool color = true}) {
    String wrap(String s, String c) => color ? '$c$_bold$s$_reset' : s;
    final out = StringBuffer('\n');
    for (final l in _lazyArt)  out.writeln('  ${wrap(l, _lazy)}');
    for (final l in _forgeArt) out.writeln('  ${wrap(l, _forge)}');
    out.writeln('');
    out.writeln('  ${color ? _dim : ''}forge database schemas without leaving your terminal${color ? _reset : ''}');
    out.writeln('  ${color ? _ember : ''}⚒  v0.1.0${color ? _reset : ''}'
                '${color ? _dim : ''}  ·  lazy by design${color ? _reset : ''}');
    out.writeln('');
    stdout.write(out.toString());
  }
}

void main() => Banner.show();
```