

The zerp style system

A short field guide for designers building decks on `zerp`. Every example on this page is rendered live by the framework's own stylesheet — what you see is what a slide gets.

00 The layers at a glance

01 Baseline — a global reset plus default styles for every content tag. A plain Markdown slide is fully designed with zero classes.

02 Type — one voice pair: Montserrat speaks, Roboto Mono counts, plus an optional *display* role for `h1`. Fonts ship inlined, so decks work offline as a single file.

03 Design tokens — ~40 CSS variables for color and elevation, generated from the Evil Martians *Harmony* palette in a dark and a light theme.

04 Utilities — small single-purpose classes for color, size, and layout. They always win over component defaults.

05 Components — opt-in slide patterns: cards, stats, comparisons, flows, steps, timelines, takeaways.

06 Guardrails — APCA contrast, type-size, and layout rules enforced by `zerp check`, in both themes.

Live examples (dashed frames) are rendered at reduced scale; slide styles are em-based, so proportions are exact.

01 Baseline

A universal reset zeroes margins and padding and sets `border-box` sizing. The page paints `--zerp-bg` behind `--zerp-text` in Montserrat. Each slide is a full-viewport flex column with generous padding (50×70px); content centers vertically (`.slide.top` top-aligns a busy slide), and direct children space themselves with an automatic 14px rhythm — you never add margins for ordinary spacing.

On that canvas, every content tag is pre-styled. This whole example uses no classes at all:

Headings are set in black weights

Body text runs at 1.25em with a relaxed line. Links look **like this**, emphasis is **bold** or *italic*, and inline `code` sits in a chip.

- Unordered lists get accent arrows
- No bullets to restyle, nothing to align

01 Ordered lists become numbered takeaway rows

02 With mono counters and hairline dividers

Quotes get an accent spine, a surface wash, and italics.

Tables	Are	Styled
hairline rows	roomy cells	no borders to fight
add <code>.mono</code>	or <code>.compact</code>	for data grids

Also covered: `pre` blocks, `kbd`, `hr`, images (rounded, height-capped), and `figure` / `figcaption`.

02 Type

Three roles, three jobs. **Montserrat** is the body voice — headings other than `h1`, body text, labels — in weights 400, 600, 700, and 900 (plus 400 italic). **Roboto Mono** (400, 700) is the data voice: code, `kbd`, stat values, list and step counters, timeline years, the slide counter. If it's a number doing work, it's mono. A third, optional **display** role applies to `h1` only — a deck that names a `display` font in its own `package.json` gets it there and nowhere else by default. Left unset, `h1` renders in the body family, because `display` with no configuration simply follows `body` — family, package and weights alike, so a deck naming only `body` gets one consistent voice everywhere. Move other elements onto the display face from your own

stylesheet (`h2 { font-family: var(--zerp-font-display) }`) or move `h1` off it (`h1 { font-family: var(--zerp-font-body) }`) — the framework rule carries zero specificity, so a plain element selector wins.

Montserrat 400 — body text and quiet lines

Montserrat 600 — gentle emphasis

Montserrat 700 — bold, labels, colored emphasis

Montserrat 900 — headlines and hero numbers

*Montserrat 400 italic — quotes & asides · Съешь ещё этих
мягких булок*

Roboto Mono 400 – code · 0123456789

Roboto Mono 700 – counters · 01 02 03

Element	Size	Weight	Color
<code>h1</code>	3.2em (4em on title slides)	900	text
<code>h2</code>	2.2em	700	text
<code>h3</code>	1.4em	700	muted (text inside cards)
<code>p</code> , <code>li</code>	1.25em (~20px)	400	text
<code>.sub</code>	1em	700	muted — the safe detail line

The floor is 1em. Nothing renders smaller except inline `.sm` spans (0.9em) inside body text — and the checker holds the line. Latin and Cyrillic subsets (with extensions) are embedded as woff2 data URLs, so a built deck is one self-sufficient file.

03 Design tokens

Every color on every slide comes from a `--zerp-*` variable. Tokens are generated at build time from the Evil Martians *Harmony* palette in two themes, and both themes

ship with every deck — so you pick tokens by **meaning**, never by hex. Hardcoded colors are forbidden; they can't survive the theme flip.

Neutrals carry the structure: `bg` (page) · `surface` (cards, panels) · `border` · `text` · `muted` (secondary text) · `faint` (decorative glyphs only — never text).

Hues — blue, green, orange, purple, red, amber, teal — each come in four roles: colored *text* on the page (`--zerp-<hue>`), a *solid* fill paired with `--zerp-on-solid`, and a soft *tint* wash paired with its own `on-tint` text color. The pairs are designed together; use them together.

Semantic aliases say why, not which: `--zerp-accent` (blue) · `--zerp-ok` (green) · `--zerp-warn` (amber) · `--zerp-danger` (red). Prefer them when color carries meaning; use hue names for categorical coding. One more: `--zerp-shadow` is the elevation shadow behind `.raised`.

DARK · PAGE IS --ZERP-BG #12141C

bg

surface

border

text

muted

faint

#12141c

#2c2e37

#484a54

#f0f3ff

#cbceda

#81848e

hue

text on bg

solid + on-solid

tint + on-tint

blue

Aa #6fc2fa

#0c8aca

#03314a · Aa #cbe8fd

green

Aa #8fc982

#4f9340

#193413 · Aa #d4ead0

orange

Aa #fba581

#c9683d

#4a2211 · Aa #ffddcf

purple

Aa #c9acfc

#9572cd

#35274b · Aa #e9dffe

red

Aa #ffa09f

#cd6265

#4b2021 · Aa #ffdbda

amber

Aa #eead69

#bb7305

#442701 · Aa #f8e0c8

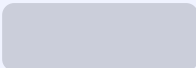
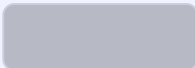
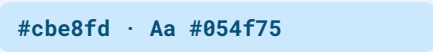
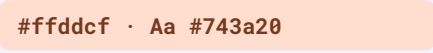
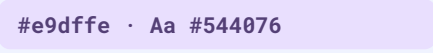
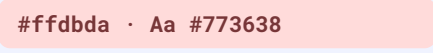
teal

Aa #6ecc9d

#009662

#003520 · Aa #cbebd8

LIGHT · PAGE IS --ZERP-BG #F0F3FF

					
bg #f0f3ff	surface #fafbfe	border #cbceda	text #2c2e37	muted #676973	faint #b7bac5
hue	text on bg	solid + on-solid		tint + on-tint	
blue	Aa #0c8aca	 #0c8aca		 #cbe8fd · Aa #054f75	
green	Aa #4f9340	 #4f9340		 #d4ead0 · Aa #2b5321	
orange	Aa #c9683d	 #c9683d		 #ffdccf · Aa #743a20	
purple	Aa #9572cd	 #9572cd		 #e9dffe · Aa #544076	
red	Aa #cd6265	 #cd6265		 #ffdbda · Aa #773638	
amber	Aa #bb7305	 #bb7305		 #f8e0c8 · Aa #6c4002	
teal	Aa #009662	 #009662		 #cbebd8 · Aa #005636	

Swatches above are read live from the shipped stylesheet — they can't drift from the framework. Tint chips are labeled in their paired on-tint color: the label demonstrates the contract.

04 Utilities

One class, one job. Component defaults are declared with `:where()` (zero specificity), so a utility placed on any element always wins — `h3.accent` recolors a heading, `.value.red` recolors a stat, `p.x1` enlarges a paragraph. Say what you mean with classes instead of inline styles.

- **Emphasis** — `.accent` `.ok` `.warn` `.danger` and the seven hue names: bold colored text by design (bold is what keeps small colored text readable — don't fight the weight). Plus `.muted` and `.mono`.
- **Size** — `.lg` (1.25em) · `.x1` (1.6em) · `.sm` (0.9em, inline only) · `.sub` (the safe muted detail line).
- **Surface** — `.tint-<hue>` colored wash · `.edge-<hue>` (and `.edge-ok/-warn/-danger/-accent`) recolors a component's border by meaning · `.raised` elevation.
- **Layout** — `.cols-2` `.cols-3` `.cols-4` equal grids · `.row` centered flex row · `.stack` column · `.spread` space-between · `.grow` · `.center` · `.narrow` width-capped centered block · `.slide.top`.

Mark words as **on track**, **at risk**, **blocked**, or **the point**; keep asides muted and identifiers `mono`.

An `.xl` line for the moment that deserves it.

And a `.sub` detail line that is safe by construction.

05 Components

Opt-in patterns that cover almost every slide you'll design. All of them inherit tokens, both themes, and the utility overrides above.

`.card`

The surface box. Add `.tint-<hue>` for a wash or `.edge-<hue>` to recolor the border; inside a card an `h3` becomes a title. Cards sharing a `.cols-*` / `.row` equalize heights.

Plain card

Surface, border, quiet.

`.tint-green`

A wash that says **good**.

`.edge-danger`

The border carries the meaning.

`.stat-row` · `.stat`

Big numbers. Each `.stat` holds a mono `.value` and a muted `.label`; recolor a value with a hue class.

5.26%

house edge

-\$17

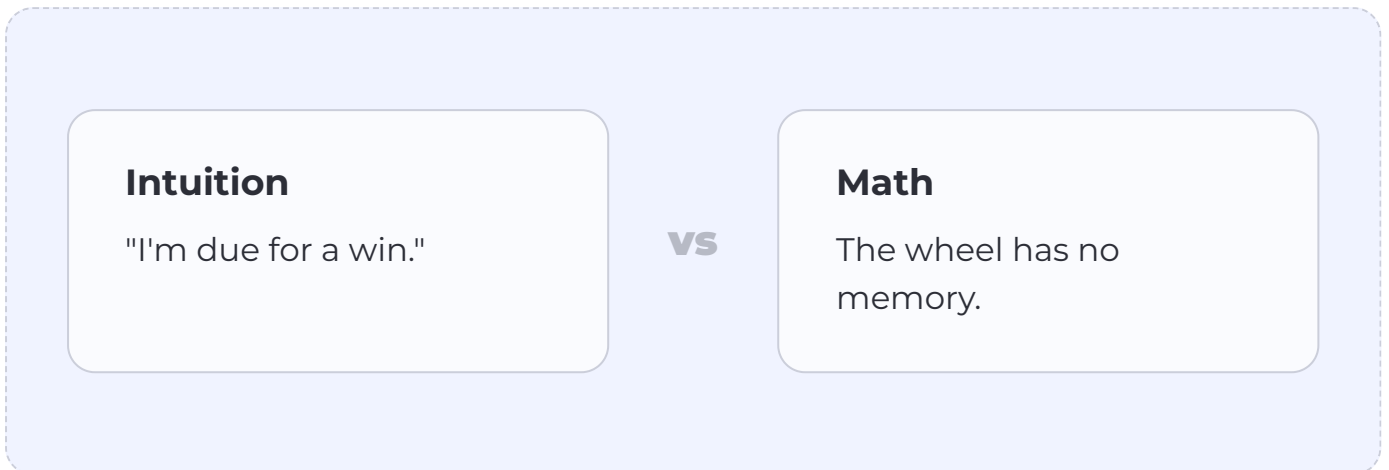
expected / hour

37:1

payout

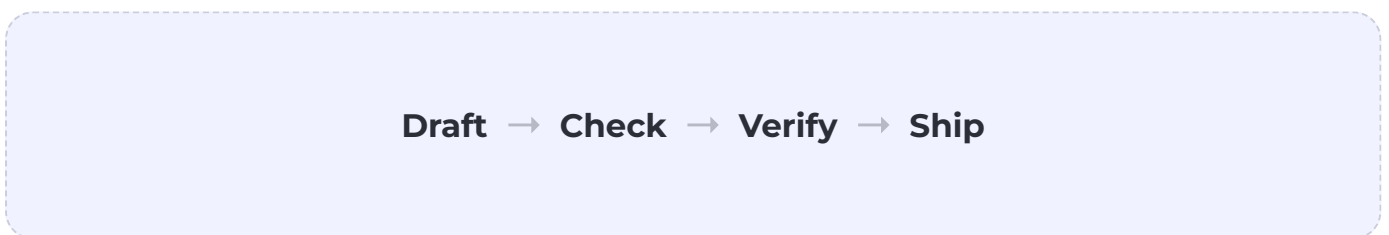
.compare

Two children with a divider — "vs" by default, or anything via `data-vs="→"`.



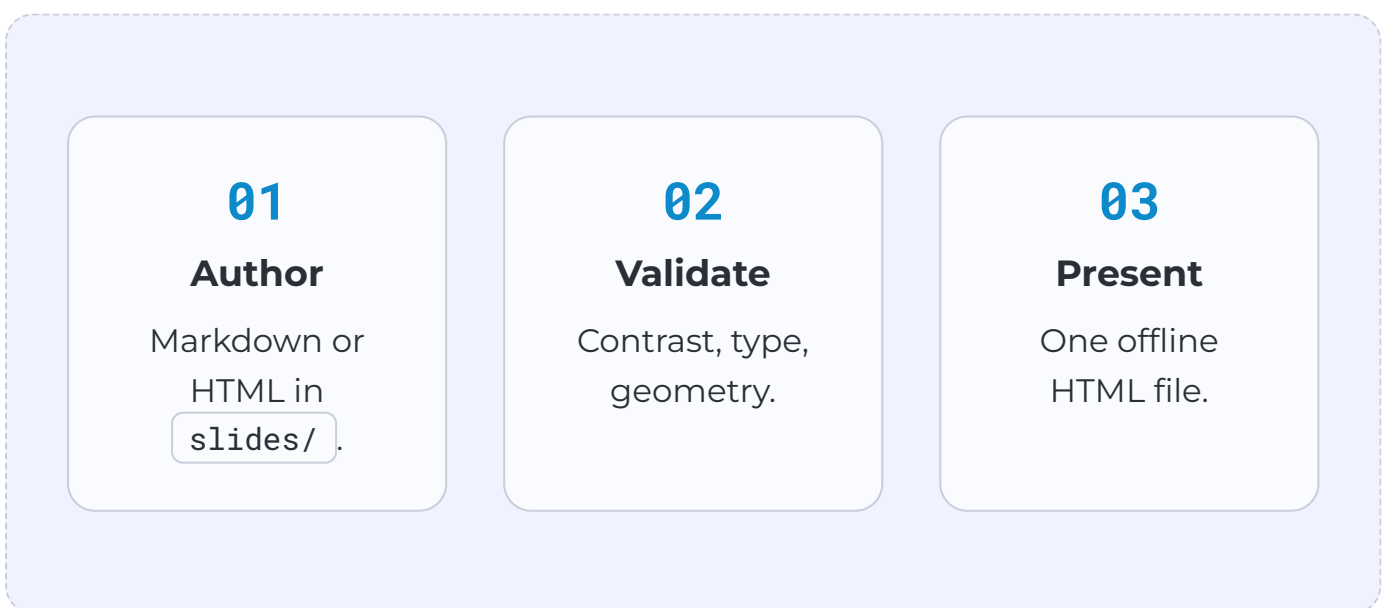
.flow

Process row; arrows appear between children automatically.



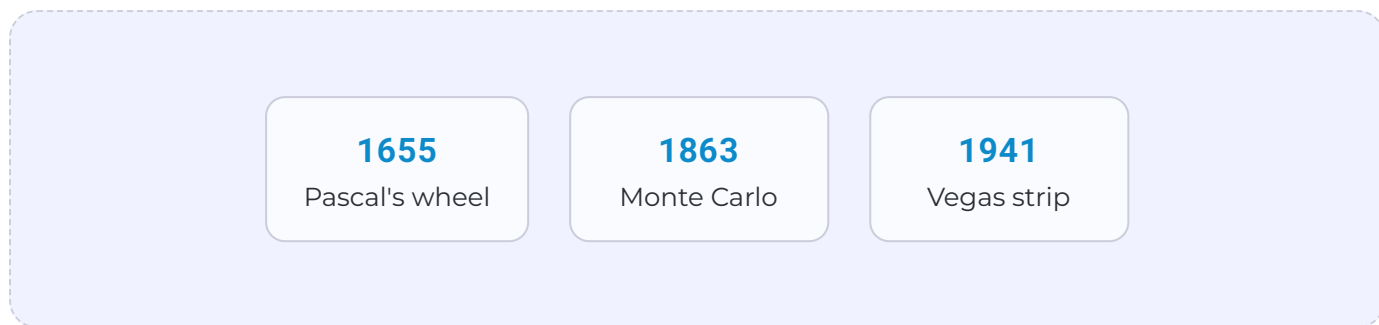
.steps

Auto-numbered card grid; children are plain divs with `h3` + `p`.



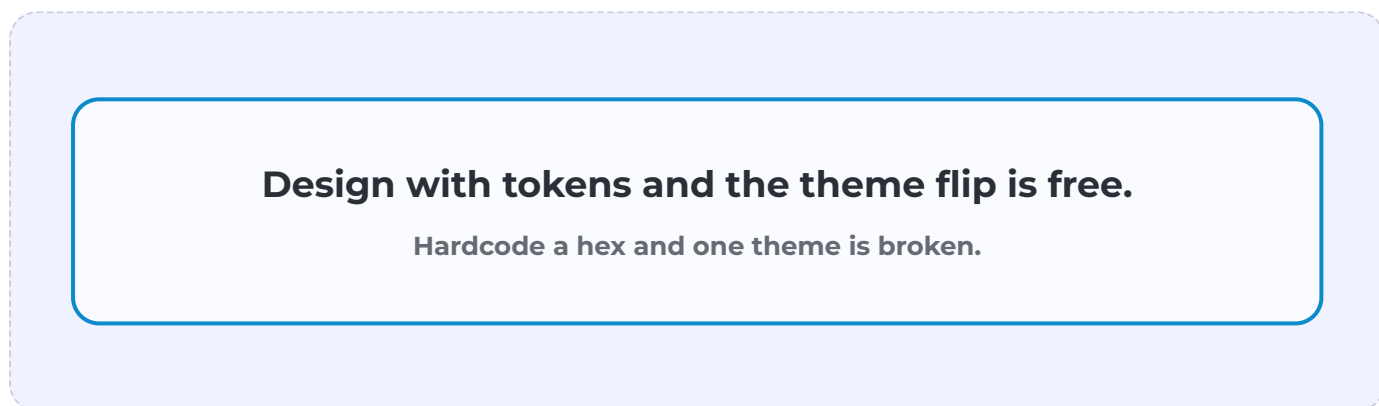
.timeline

Milestone row of `.item`s, each with a mono `.year` and a `.label`.



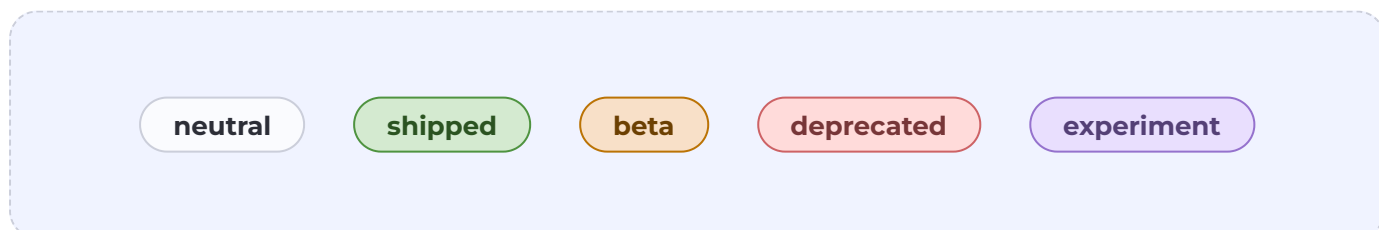
.key-thought

The boxed takeaway — one per slide, maximum.



.pill

Inline badge; combine with `.ok / .warn / .danger / .accent` or `.tint-<hue>` for the matched tint/on-tint pair.



.concept

Definition-slide header: a big `.icon` over a shouty accent `.label`; follow with a muted tagline.



HOUSE EDGE

The casino's built-in margin on every bet.

.slide.title

The title-slide recipe: `h1` + `h3` (auto-accented) + a `.meta` block for presenter and date.

Beating the House

Why the math always wins

Eugene Mirotin · July 2026

Also in the box: `.block-label` (top-left section label) · `figure.corner` (pinned reference figure) · `.img-row` (captioned image rows) · `.interactive-badge` + `.step-hint` (the affordances for stepped slides) · `table.mono` / `table.compact` · and the step-reveal attributes `data-step` / `data-until-step`, whose styling reserves space so reveals never reflow a centered slide.

06 Theming & guardrails

The theme is a single attribute — `data-zerp-theme` set to `dark`, `light`, or `system` — and only token values change with it. Presenters pick a default at build time; viewers override it with the sun/moon toggle in the deck chrome (or the `t` key), which pins the opposite scheme and, on a second press, hands control back to the default. Because both themes always ship, everything you design must work in

both — that's the real reason raw colors are banned. (Sole exemption: depicted real-world objects — a playing card's white face stays white.)

The system is enforced, not aspirational. `zerp check` loads every slide in a real headless browser, against both themes: APCA text contrast, the lem type floor, surfaces that blend into their background, font fallback, and — in the same pass — frame geometry, overflow, and console errors. The published defaults all pass; the checker exists to keep *your* additions honest. Known traps it will catch: `.sm` + `.muted` combined, `faint` used for text, any text under lem.

When you need to go beyond the box, escalate in order: built-in components → utility classes → a slide-scoped `<style>` that uses only `var(--zerp-*)` tokens (deck-wide styles live in a style-only `slides/00-styles.html`). Then re-run the checks.

One last layer rides along for free: **print**. A built deck paginates as one page per slide, hides the presentation chrome, and prints steps in their final state — the PDF export is the same file.

Sources of truth: `src/assets/base-styles.css` (rules) · `scripts/generate-tokens.mjs` (tokens) · `llms.txt` (the full authoring contract). Generated 2026-08-18 against zerp 0.11.0.