# Core Web Vitals Field Guide

How to measure, diagnose and actually fix LCP, INP and CLS on real sites with real traffic.

The three metrics are measured on real visits, in the 75th percentile. That means the slowest quarter of your visitors decide whether you pass. A fast homepage on your laptop proves nothing.

## The thresholds

| Metric | What it measures | Good | Poor |
|---|---|---|---|
| LCP | When the largest element in view finishes rendering | 2.5s or less | over 4.0s |
| INP | Responsiveness across the visit, from tap to paint | 200ms or less | over 500ms |
| CLS | How much the layout jumps while loading | 0.1 or less | over 0.25 |

## Where to measure

- **Search Console, Core Web Vitals report** for field data grouped by URL pattern. This is your source of truth.
- **PageSpeed Insights** for a single URL: field data at the top, lab trace underneath for diagnosis.
- **Chrome UX Report API or BigQuery** when you need to compare against competitors or track a trend over months.
- **A real device on a throttled connection** for the last word. Emulation tells you what to look at, not how it feels.

## Fixing LCP

Find the element first. It is usually the hero image, a background video, or a heading waiting on a web font. Then attack it in this order: remove the round trip, then the bytes, then the render blocking.

- Serve the hero from the same origin or a fast edge, and preload it. No lazy loading on the hero, ever.
- Compress to WebP or AVIF, size images to their display box, and set width and height attributes.
- Cut render-blocking CSS and defer non-critical JavaScript. A hero waiting on a script tag is a self-inflicted wound.
- Cache HTML at the edge where the content allows it, and use a CDN for static assets.
- Watch for fonts: use font-display swap, self-host, preload the single weight you actually use.

## Fixing INP

INP is the metric most sites fail quietly, because the page looks fine while taps feel delayed.

- Break long tasks. If a click handler runs for 300ms, the response is 300ms late by definition.
- Defer third-party tags: chat widgets, heatmaps, A/B tools and tag managers are the usual culprits.
- Hydrate less. Ship interactivity where it is needed rather than making the whole document a client app.
- Give feedback immediately: a pressed state or skeleton that paints before the work finishes reads as instant.
- Audit event handlers on scroll and input. Debounce, throttle, or move work off the main thread.

## Fixing CLS

- Reserve space for every image, embed and ad slot with explicit dimensions or aspect ratio boxes.
- Never inject a banner, cookie bar or promo above existing content after paint. Make room at first render.
- Load fonts with matching fallback metrics, or accept one deliberate shift rather than several.

- Avoid animating properties that trigger layout. Transform and opacity only.

## A diagnosis routine that works

- 1. Pull the failing URL group from Search Console and pick the worst real template, not the worst test result.
- 2. Record a performance trace on a throttled mobile profile and identify the largest task and the LCP element.
- 3. Fix one thing, then re-measure the field data. Field data lags by up to 28 days, so keep a lab trace alongside it.
- 4. Re-check after deployment week, not deployment hour. Bots and repeat visitors skew the first days.

Rule of thumb: performance work that is not measured in the field is decoration. Pick the two templates that carry the most revenue, fix those to green, then let the rest follow the same pattern.