DevFest

Online

# Measure, guide, fix: Core Web Vitals with an AI agent

Ivan Kristianto
GDE Web Performance



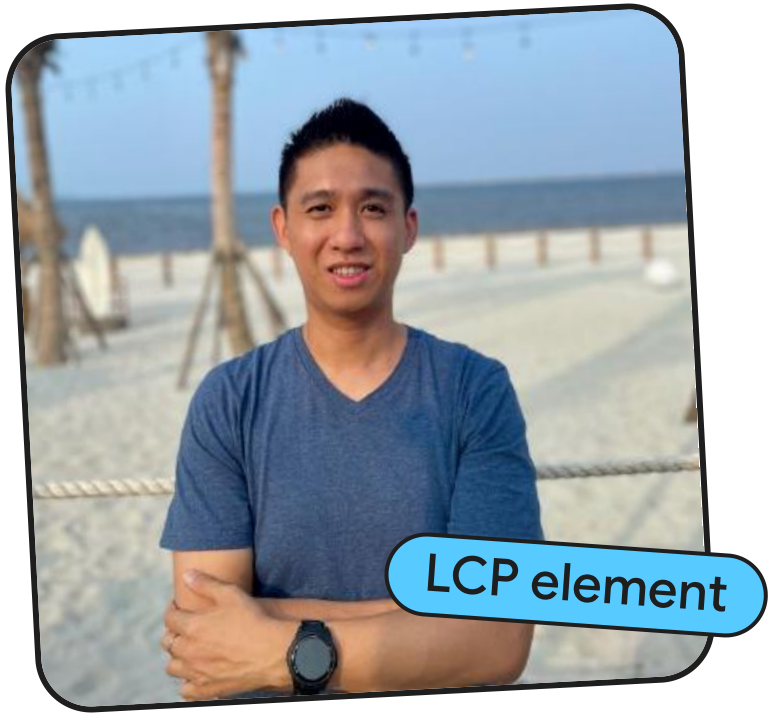
Google Developer Groups

# Hi, I'm Ivan

## GDE, Web Performance

ivankristianto.com

| | |
|---|---|
| Day job | Senior Web Engineer, Human Made |
| GDE since | 2018 |
| Podcast | Co-host of *ngobrolin web* |
| Middle name | Performance |

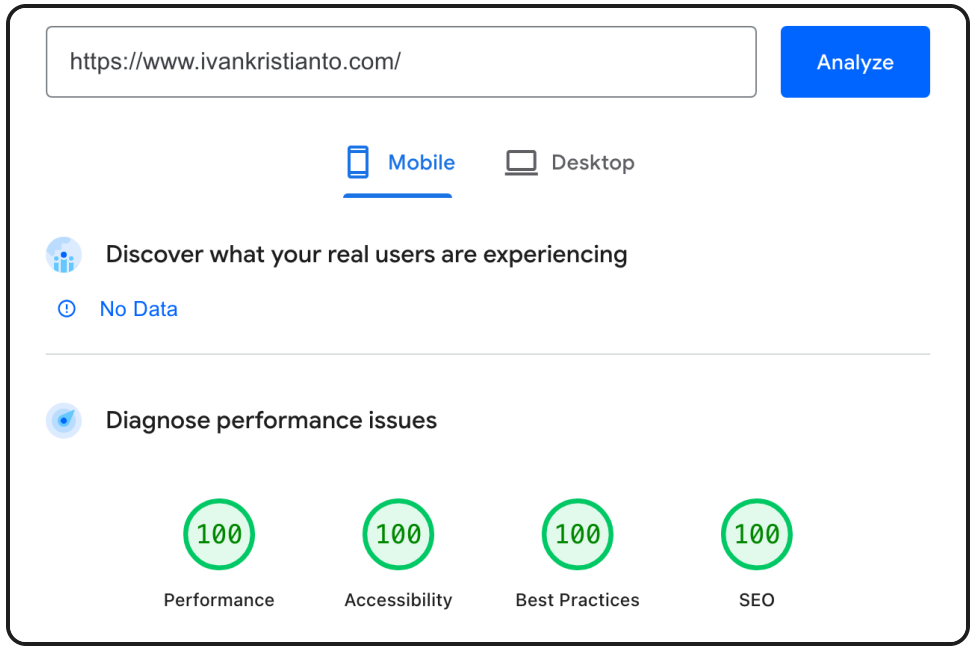



My AI agent's first fix looked right. The page stayed slow.

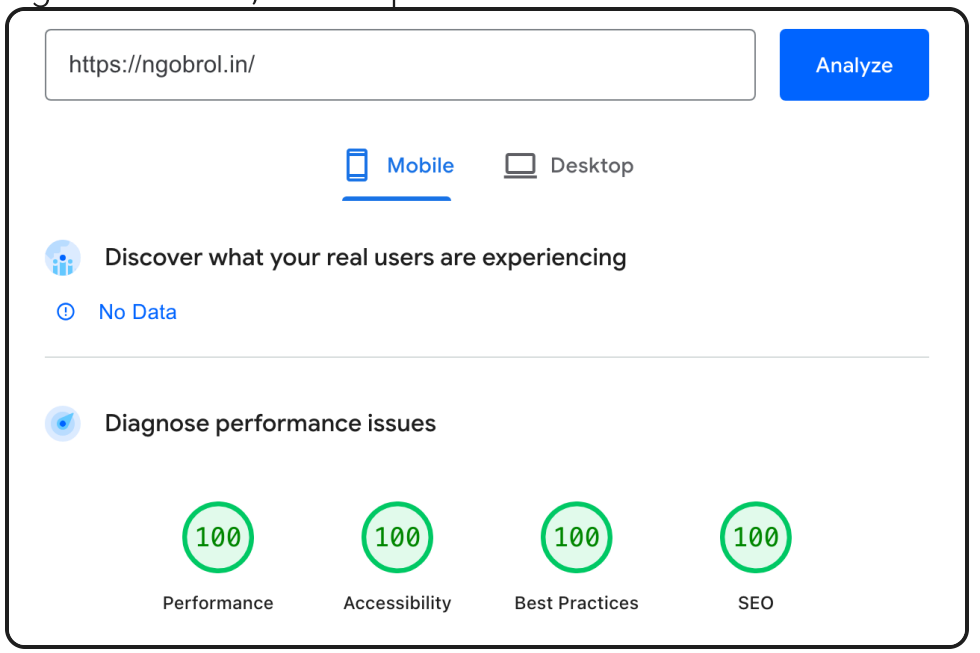
Google Developer Groups

# Proof: 100 on mobile

ivankristianto.com




ngobrol.in, the podcast




PageSpeed Insights, Mobile, 15 November 2026. Lighthouse lab test, one run each.

Lab test only. No real-visitor data.

Google Developer Groups

# Try the broken site

LCP The main image shows up late.

INP Typing in search feels stuck.

CLS An announcement pushes the page down.

Run `npm run sample:serve`, open 127.0.0.1:4173/starter/



Google Developer Groups

# Three numbers for how a page feels

| Metric | What a visitor feels | Good |
| --- | --- | --- |
| LCP | How long until the main content shows | 2.5 s or less |
| INP | How long the page takes to react to a tap or a key | 200 ms or less |
| CLS | How much the content jumps while you read | 0.1 or less |

Together they are the Core Web Vitals.



Google Developer Groups

# Modern Web Guidance

- AI coding agents learned from years of old web code. Some of it is out of date.
- Short guides by web experts from Chrome, Edge, Firefox and MDN.
- Version 1.0 came out on 10 November 2026, with 203 guides.
- Two commands: one adds it to your project, one to Antigravity. The agent finds it by itself.

In its own words:

"web APIs evolve rapidly and training weights contain obsolete patterns"

Plainly: browsers change fast, and what the AI learned gets old.

```
npx modern-web-guidance@latest install
agy plugin install \
https://github.com/GoogleChrome/modern-web-guidance
```

Overview video: youtu.be/PsrHq5F_BAc (Chrome for Developers, 7 min)



Google Developer Groups

# How the agent uses a guide

**Search**
a few words about the task

→

**Read**
one guide, all of it

→

**Check**
its code against the guide

→

**Tell**
which guide it used

```
npx -y modern-web-guidance@latest search "prioritize hero image"
# optimize-image-priority, html, performance, ...
```

No guide named? Then no guide was used.



Google Developer Groups

# What a guide contains

1. Steps to follow
2. Rules, including what not to do
3. Which browsers support it

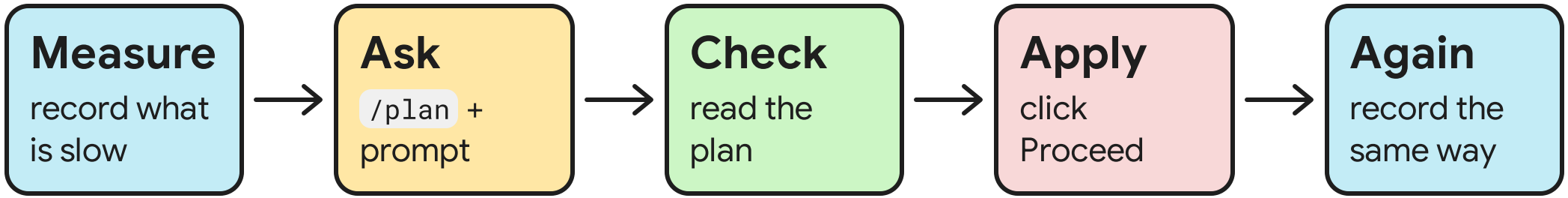
```
## How to implement
1. Identify the LCP image
2. Add fetchpriority="high" to it
## Best practices
DO NOT combine fetchpriority="high"
with loading="lazy"
## Fallback strategy
Baseline since 2024-10-29.
Chrome 103, Edge 103, Firefox 132, Safari 17.2
```

Shortened from the optimize-image-priority guide. It can't see our page, so we measure first.



Google Developer Groups

# One loop, three times




**Check the plan, every time:**

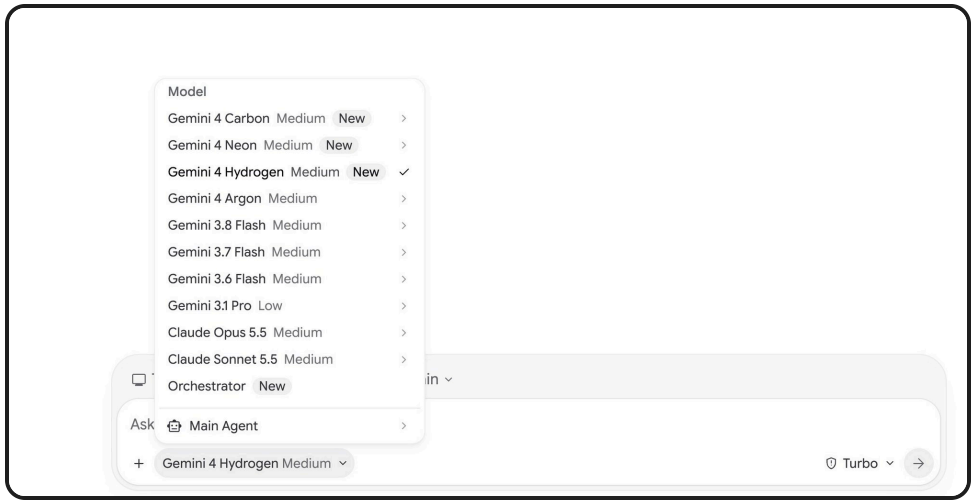
1. Does it fix what we measured?
2. Does the page still work the same?
3. Does it name the guide it used?

No App? In a terminal: `agy --mode=plan`

Google Developer Groups

# Get ready

- [ ] Antigravity App open and signed in
- [ ] Model set to Gemini 4 Hydrogen
- [ ] Guidance in `.agents/skills/`
- [ ] Guidance in `agy plugin list`
- [ ] Sample running at 127.0.0.1:4173/starter/
- [ ] The agent can open the sample

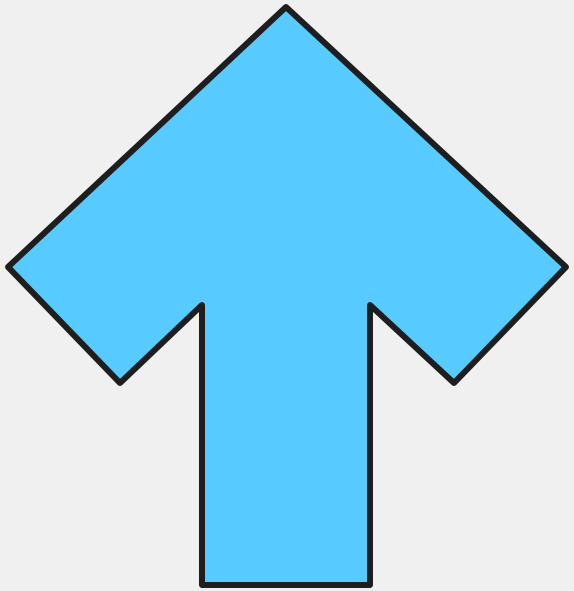



Antigravity App

Stuck? Watch my screen. You can repeat it all after the session.

Google Developer Groups

01



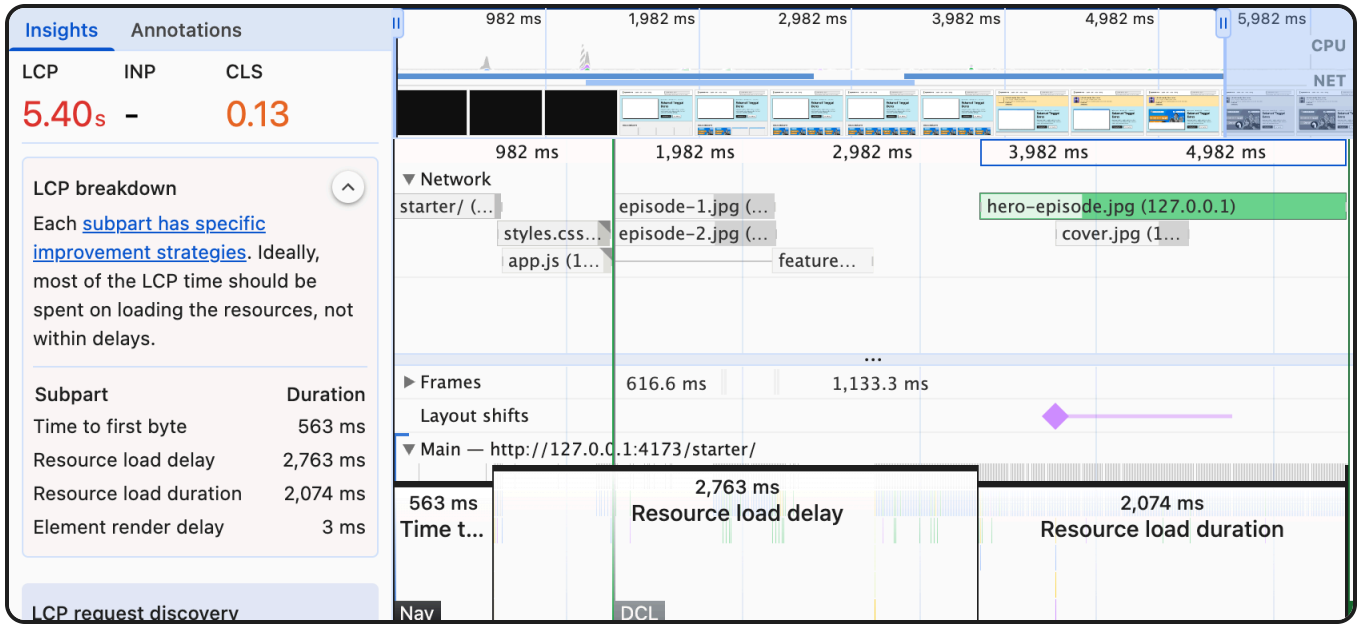
# The main image shows up late

LCP: how long until the main content shows.

Google Developer Groups

# Measure: record a reload

1. DevTools, **Performance** panel
2. Network: **Slow 4G**, like a phone
3. Click Record and reload
4. Find the **LCP**: which image, and when did it start downloading?




Own capture, Chrome 157 DevTools, Slow 4G, 4x CPU slowdown

Watch first, then do the same.

Google Developer Groups

# Ask: type /plan, then this

```
LCP case. Open http://127.0.0.1:4173/starter/ and record a reload
with Slow 4G network throttling. Find the LCP element and when
its download starts. Use Modern Web Guidance on loading the hero
image early. Explain why it shows up late and propose the smallest
fix. Keep the same image at the same size.
```

Copy it from workshop/prompts.md



Google Developer Groups

# Check the plan

**A good plan:**

- ☐ Puts the image in the HTML
- ☐ Same image, same size
- ☐ Names its guide

```
async function showHero() {
  const response = await fetch(FEATURED_URL);
  const featured = await response.json();
  await wait(FEATURED_DELAY_MS);
  heroImage.alt = featured.hero.alt;
  heroImage.src =
      new URL(featured.hero.src, response.url).href;
}
```

```
<img id="hero-image" width="1280" height="720" alt="">
```

Expected guide: optimize-image-priority. Plan looks right? Click Proceed.



Google Developer Groups

# Measure again

- Same recording, Slow 4G, fresh reload
- The image download starts right away
- Same image, same size

Before

| Subpart | Duration |
|---|---|
| Time to first byte | 563 ms |
| Resource load delay | 2,782 ms |
| Resource load duration | 2,074 ms |
| Element render delay | 3 ms |

After

| Subpart | Duration |
|---|---|
| Time to first byte | 563 ms |
| Resource load delay | 23 ms |
| Resource load duration | 1,750 ms |
| Element render delay | 3 ms |

LCP breakdown, own capture, Chrome 157 DevTools, Slow 4G

**5.4 → 2.4 s**

Lab, headless Chromium, Slow 4G, 4x slower CPU, middle of 5 runs

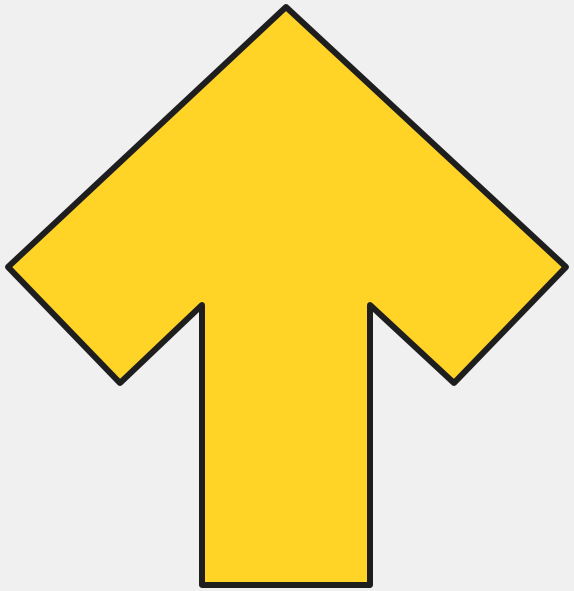
LCP, before → after

Your machine helps find the cause. Real visitors decide the score.



Google Developer Groups

02



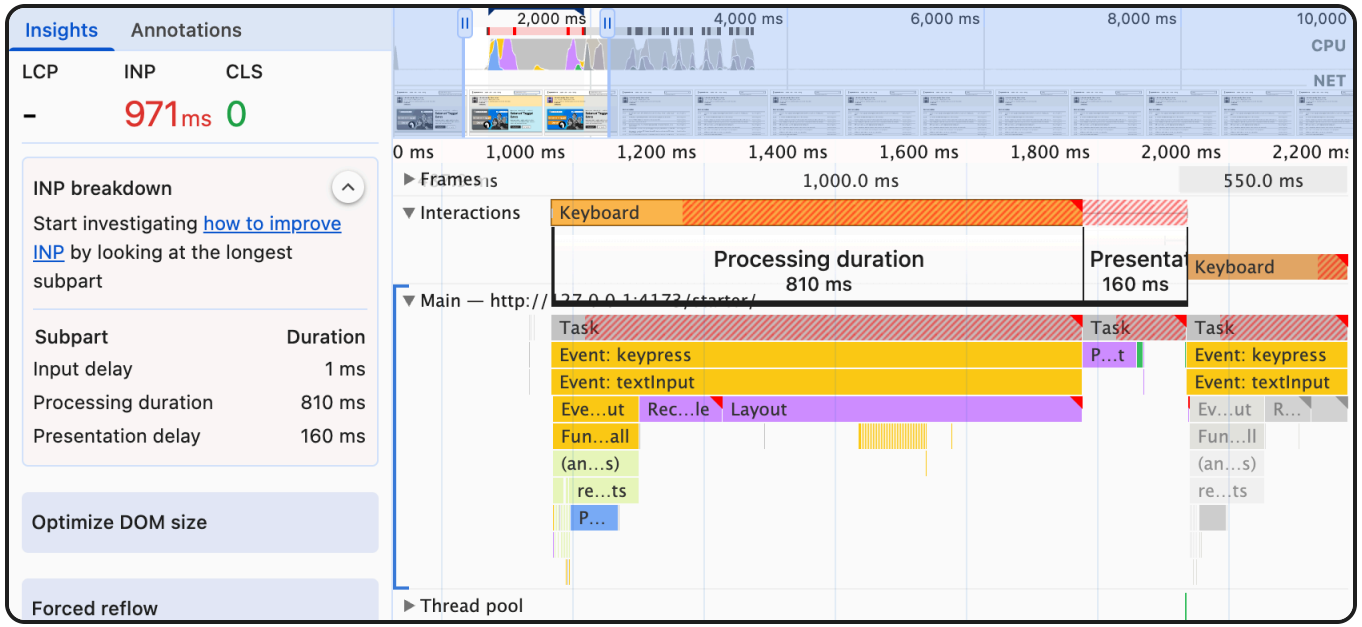
# Typing in search feels stuck

INP: how long the page takes to react to a tap or a key.

Google Developer Groups

# Measure: record while typing

1. Reload and wait
2. Record, type **coding**, stop
3. Find the **key press**: what work runs under it?




Own capture, Chrome 157 DevTools, 4x CPU slowdown

INP needs a real key press. A reload alone won't show it.

Google Developer Groups

# Ask: type /plan, then this

```
INP case. Open http://127.0.0.1:4173/starter/, record typing
"coding" in the search box, and find what makes the page slow
to react. Use Modern Web Guidance on breaking up long tasks.
Propose the smallest fix. Keep every search result, and show
only the latest search when someone types fast.
```

Copy it from workshop/prompts.md



Google Developer Groups

# Check the plan

**A good plan:**

- ☐ Keeps every result
- ☐ Typing fast shows only the latest search
- ☐ Names its guide

```
let html = '';
let deadline = performance.now() + 50;
for (const line of matches) {
  html += lineCard(line);
  if (performance.now() >= deadline) {
    await yieldToMain();
    if (token !== renderToken) return;
    deadline = performance.now() + 50;
  }
}
results.innerHTML = html;
```

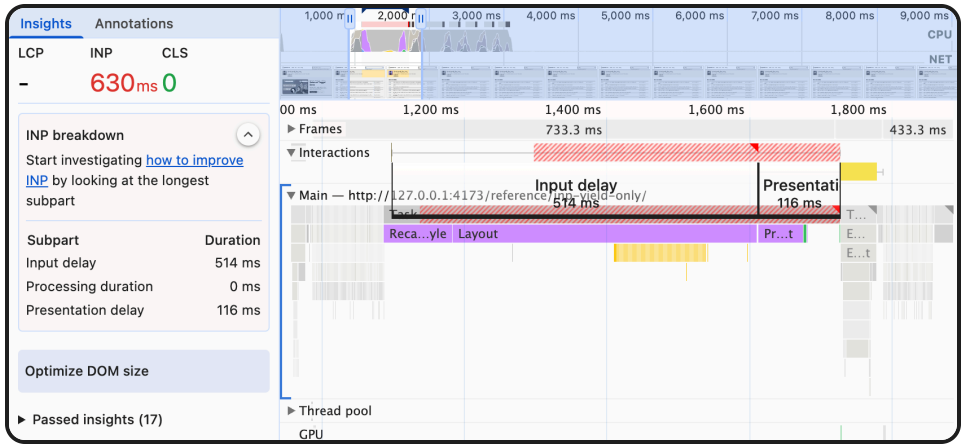
Expected guide: break-up-long-tasks. Plan looks right? Click Proceed.



Google Developer Groups

# Round 1: still slow

- Same typing, fresh reload
- The code now has `scheduler.yield()`
- The recording still shows one long block




Own capture, Chrome 157 DevTools, 4x CPU slowdown

**632 ms** Lab, headless Chromium, 4x slower CPU, middle of 5 runs
slowest key press, round 1

Code that looks right is only a guess until you measure it.

Google Developer Groups

# Round 2: let the browser draw

Let the browser draw each batch before starting the next one.

I lead this one

```
for (let start = 0; start < matches.length;
    start += BATCH_SIZE) {
  const html = matches.slice(start, start + BATCH_SIZE)
      .map(lineCard).join('');
  results.insertAdjacentHTML('beforeend', html);
  await nextFrame();
  await yieldToMain();
  if (token !== renderToken) return;
}
```

**32 ms**

Lab, headless Chromium, 4x slower CPU, middle of 5 runs

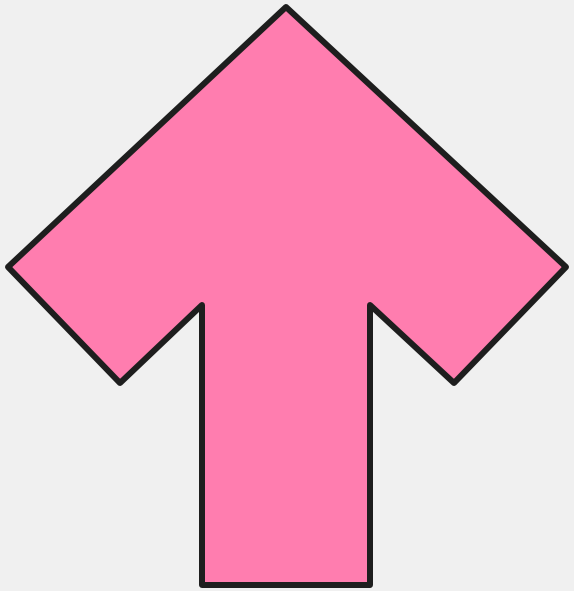
slowest key press, round 2

Round 2 prompt: workshop/prompts.md



Google Developer Groups

03



# The page jumps

CLS: how much the content jumps while you read.

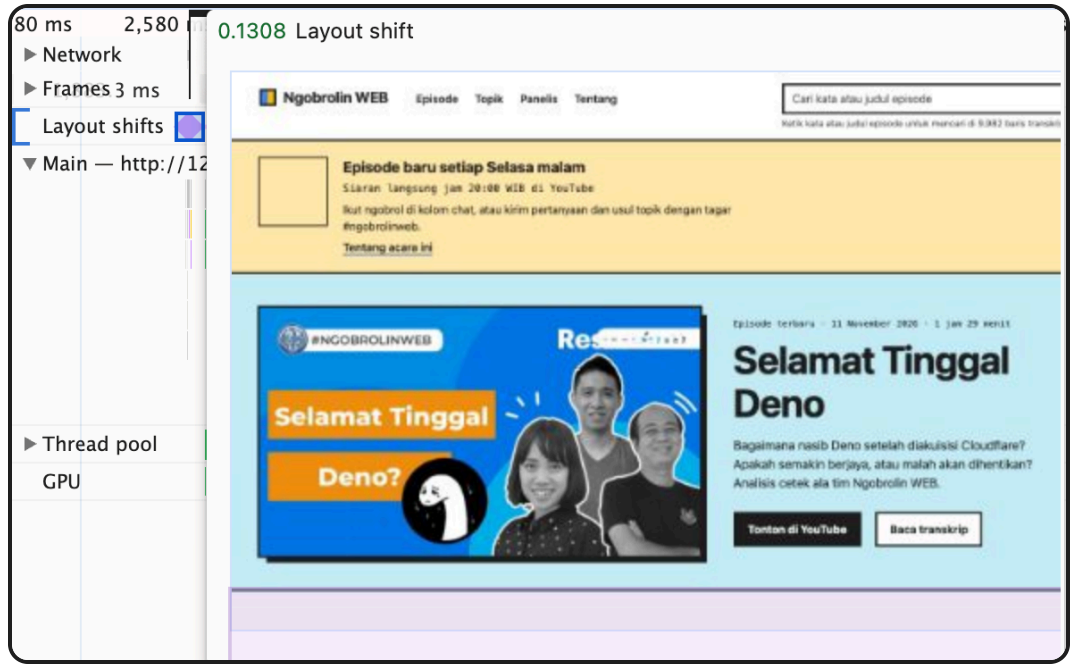




Google Developer Groups

# Measure: record a reload, hands off

1. Record and reload
2. Don’t touch anything
3. Find the **layout shift**: what moved, and what pushed it?




Own capture, Chrome 157 DevTools

Touch the page and the shift won’t count. Hands off.

Google Developer Groups

# Ask: type /plan, then this

```
CLS case. Open http://127.0.0.1:4173/starter/ and record a reload
without touching the page. Find what moves and what pushes it.
Use Modern Web Guidance on preventing layout shifts.
Propose the smallest fix that keeps the announcement.
Check it at a desktop width and a phone width.
```

Copy it from workshop/prompts.md



Google Developer Groups

# Check the plan

**A good plan:**

- ☐ Keeps the announcement
- ☐ Saves its space from the start
- ☐ Works at a phone width too
- ☐ Names its guide

```
const notice = document.createElement('aside');
notice.className = 'announcement';
// …
document.querySelector('main').before(notice);
```

```
.announcement {
  /* … */
  margin-block: 0;
  padding-block: 1.5rem;
  /* … */
}
```

Plan looks right? Click Proceed.

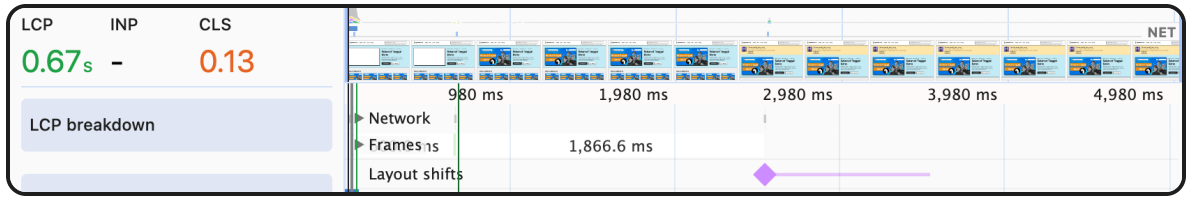


Google Developer Groups

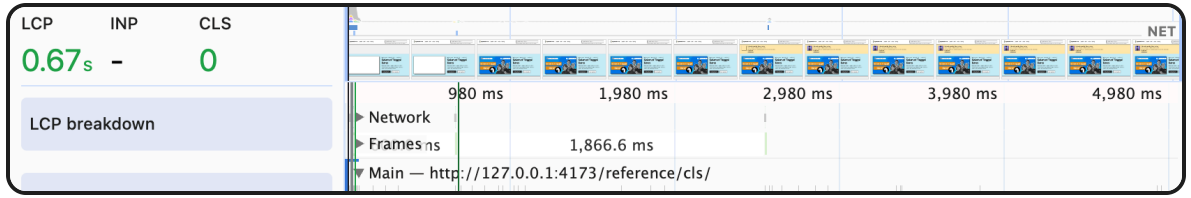
# Measure again

- [ ] Same recording, hands off
- [ ] No shift when it arrives
- [ ] Phone width too (0.237 → 0 in the lab)

Before




After




Own capture, Chrome 157 DevTools, 4x CPU slowdown

**0.131 → 0**

Lab, headless Chromium, 4x slower CPU, middle of 5 runs

CLS, desktop width

The announcement still shows. It just has its own space now.

Google Developer Groups

# What we found

LCP The browser couldn't see the image until JavaScript set it. In the HTML, it loads right away.

INP Splitting the work wasn't enough. The browser also needed a chance to draw between batches.

CLS The announcement arrived with no space saved. Saving the space stopped the jump.

Each time, the second recording decided.



Google Developer Groups

# Take the loop home

1. **Record twice.** Before the change and after it. The second recording decides.
2. **Read the plan.** The guide makes the agent better. Only you can check it against your page.
3. **Laptop finds, visitors decide.** Find the cause on your machine. Real visitors set the score.

Sample, prompts and agent rules
`github.com/ivankristianto/google-devfest-2026`

`Notes: ivankristianto.com`

Google Developer Groups