# Core Web Vitals

Performance & User Experience Metrics

# What are the Core Web Vitals?

**3 core metrics** specified by Google to measure **performance** & the **user experience** for web pages.
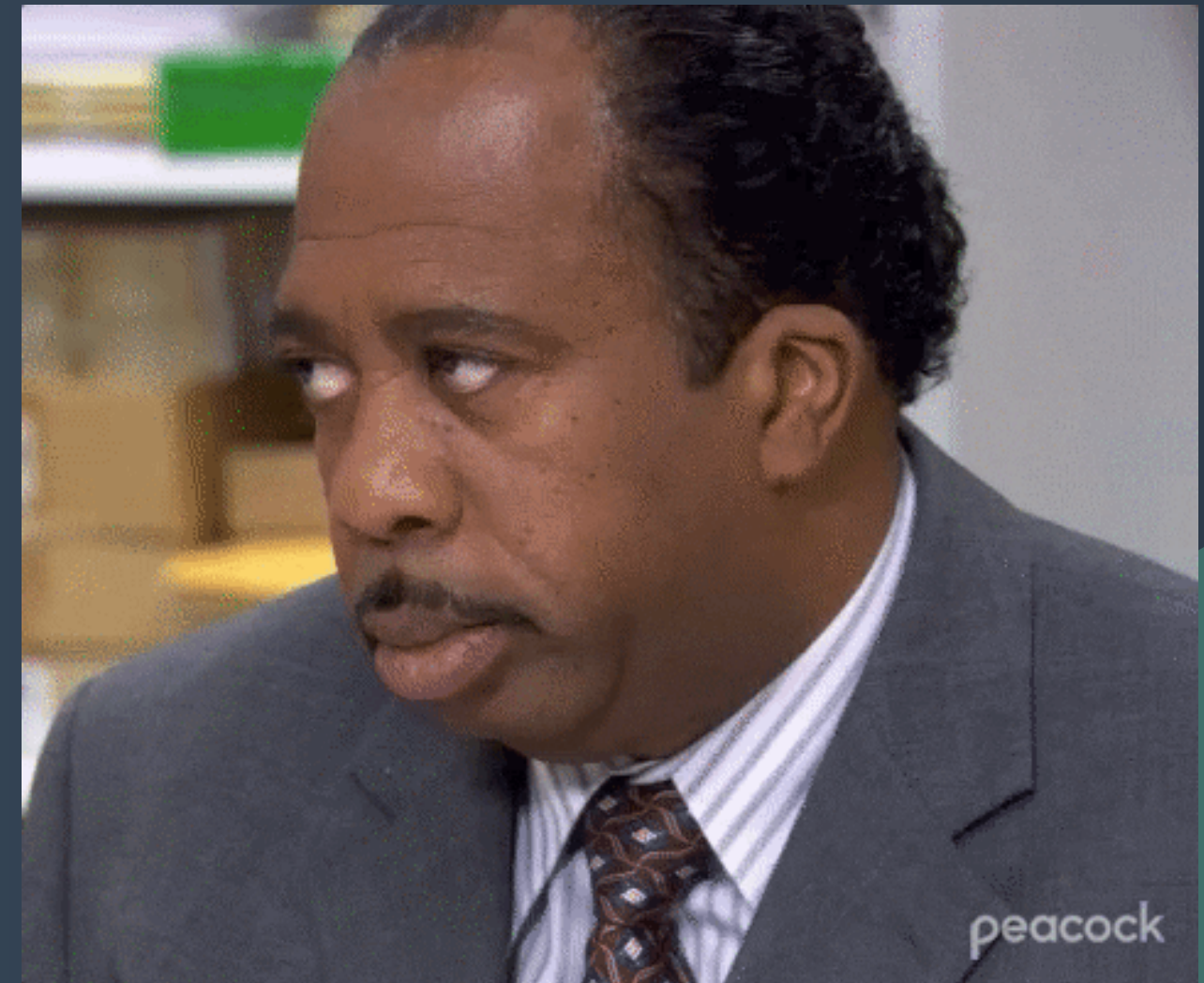
- **Largest Contentful Paint (LCP)** - loading

- 👆 **First Input Delay (FID)** - interactivity

- ⬇ **Cumulative Layout Shift (CLS)** - visual stability

niji

# Why should I care?

- Better web vitals indicate a **better user experience**

- Monitoring those vitals can help **identify performance issues**

- **Google Search rankings** take the vitals into account to determine a site's page ranking

Learn more: Google Search - page experience

# Metrics Definition

# Largest Contentful Paint (LCP)

Measures the time at which the **largest content** on the page is **rendered**.
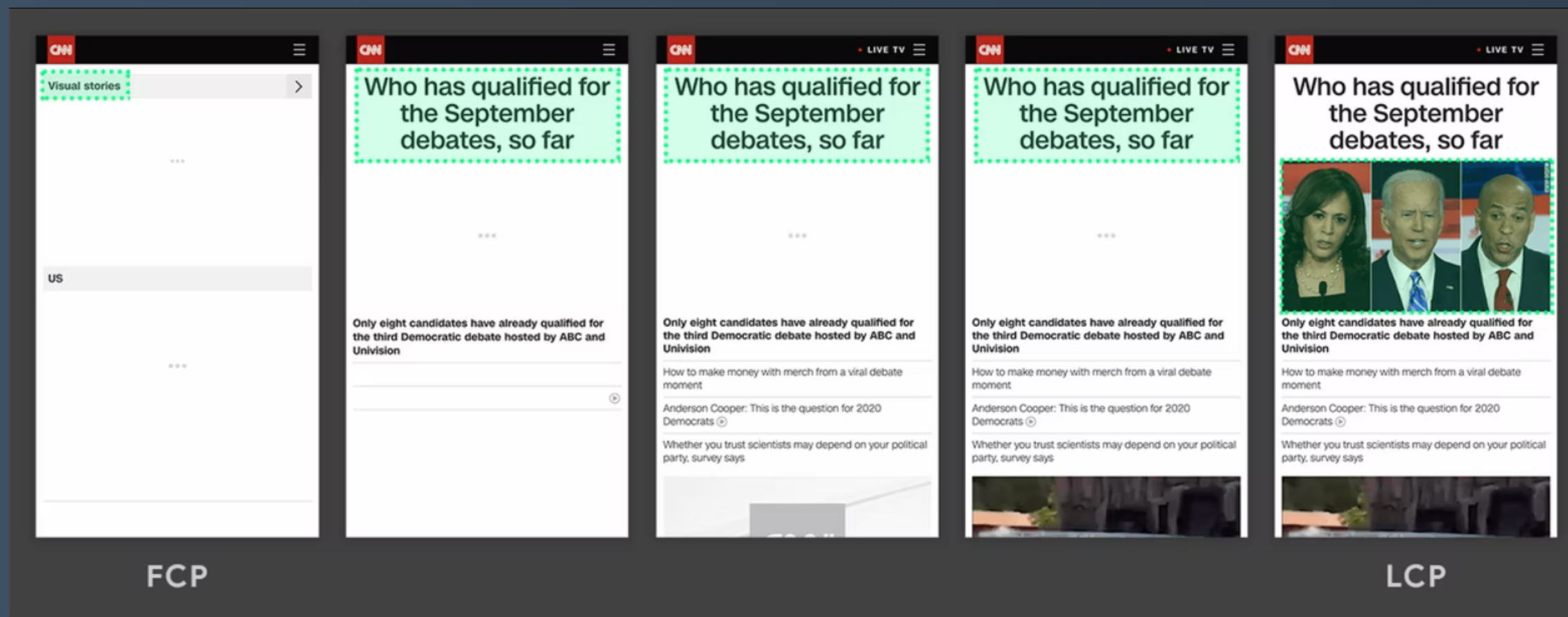
-> page **loading speed**

*A good LCP is <2.5s*
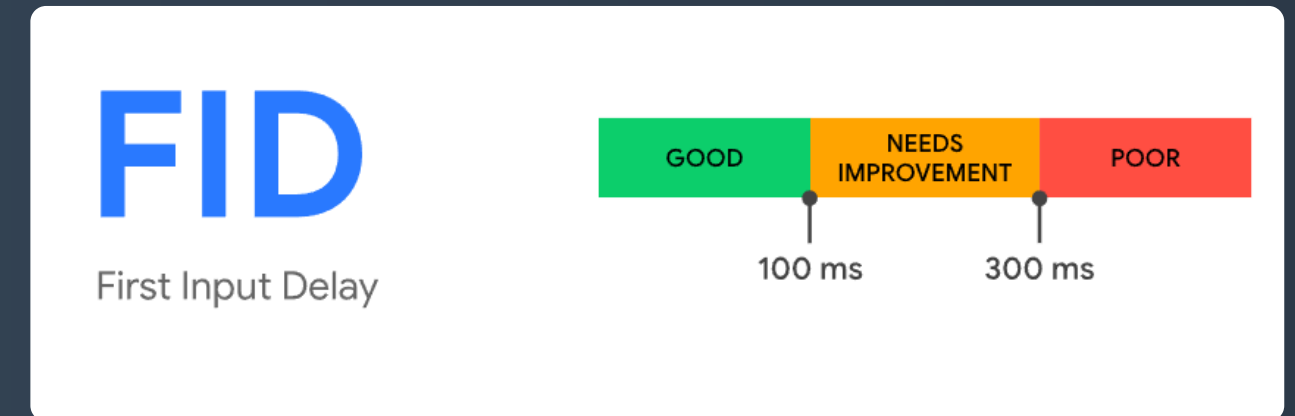
# First Input Delay (FID)

*"FID measures the delay between when a user first interacts with a page [...] and when the browser is actually able to begin processing event handlers in response to that interaction"* - web.dev - FID

*ex: click a link, tap a button, use a JS-powered control...*

–> page **interactivity & load responsiveness**
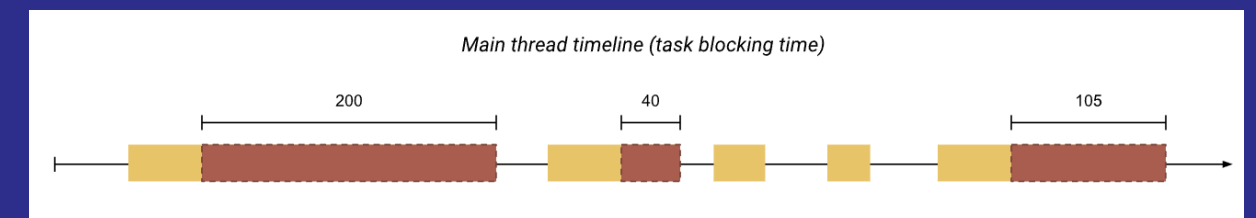
FID (and TBT) can help identify **performance issues**

*ex: main thread blocked while parsing/executing a large JS file*



*A good FID is <100ms*

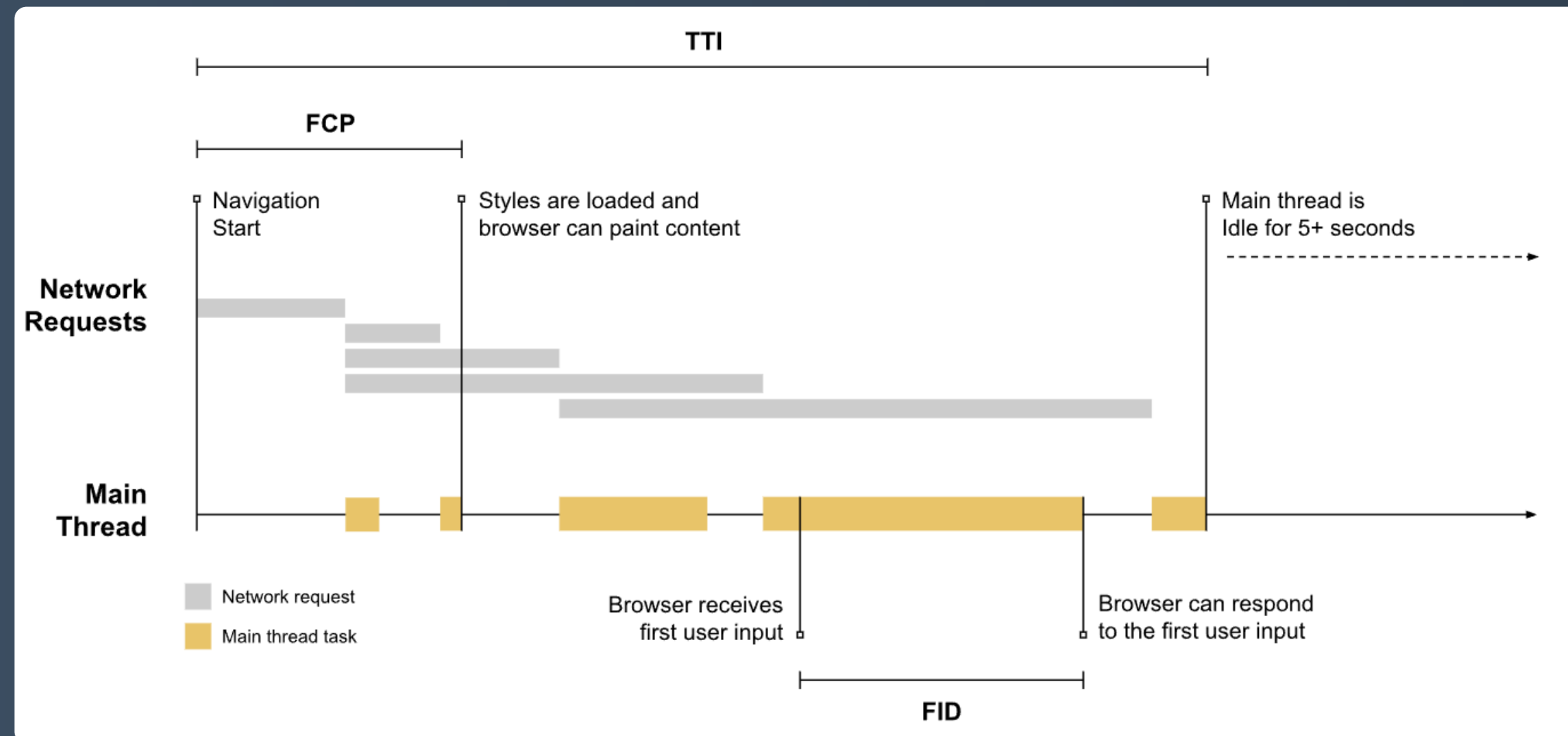**Total Blocking Time (TBT, tasks >50ms)** is a substitute to FID when lab testing



*TBT representation on main thread timeline*

# First Input Delay (FID)

Schema representing FID:

*- delay after first user input before the
browser can respond to the event*
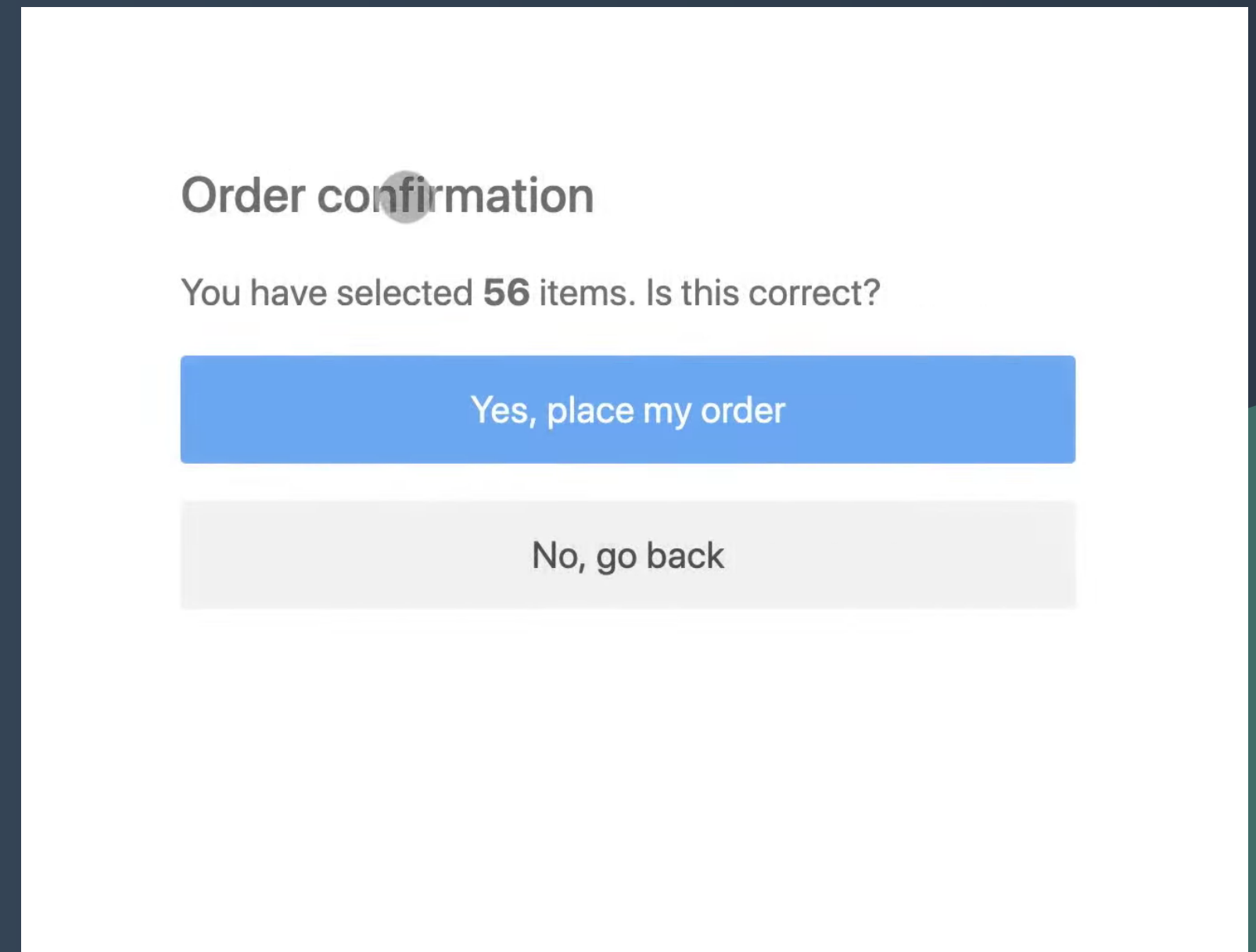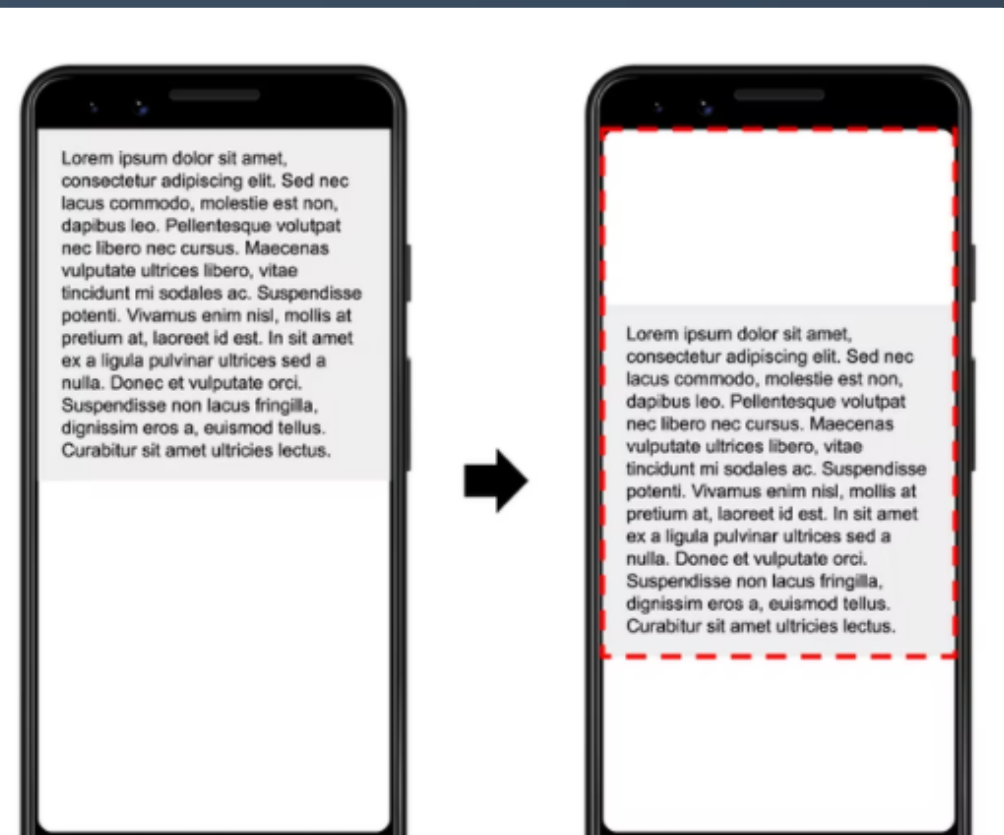
*See web.dev - FID*

# Cumulative Layout Shift (CLS)

**What is a Layout Shift?**

A layout shift is when **a visible element changes position**, usually due to an element being added to the DOM (above it).

Some layout shifts are *fine* (ex: shortly after user events).
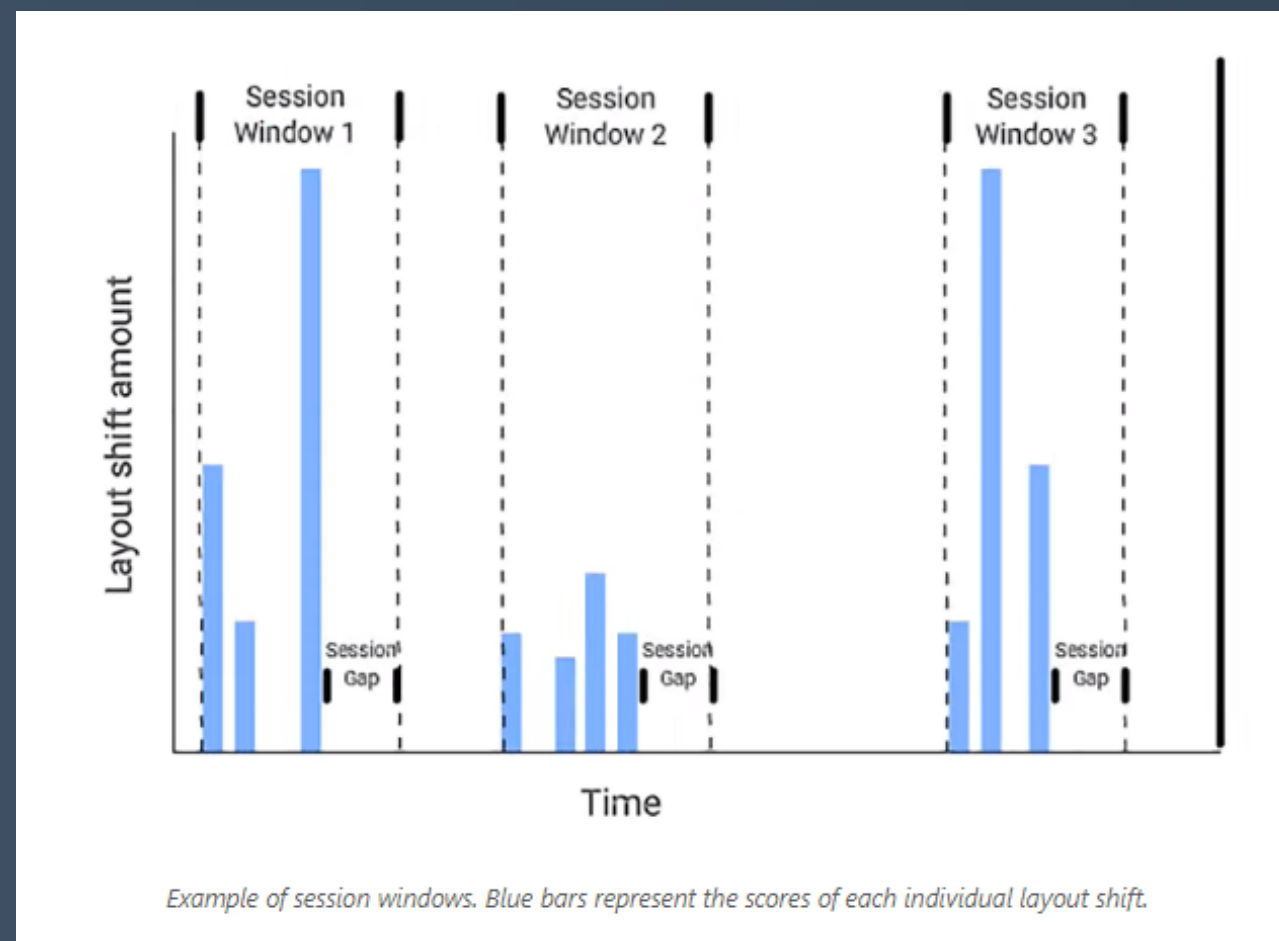




A layout shift in action... 😠

# Cumulative Layout Shift (CLS)

The **CLS** is a measure of the **largest burst of layout shift scores** for every unexpected layout shift that occurs during the entire lifespan of a page.

–> page **visual stability** over time


*CLS — Cumulative Layout Shift*

GOOD | NEEDS IMPROVEMENT | POOR
0.1 | 0.25

*A good CLS is <0.1*


*Example of session windows. Blue bars represent the scores of each individual layout shift.*

*source: web.dev - CLS*

# Improving Web Vitals Scores
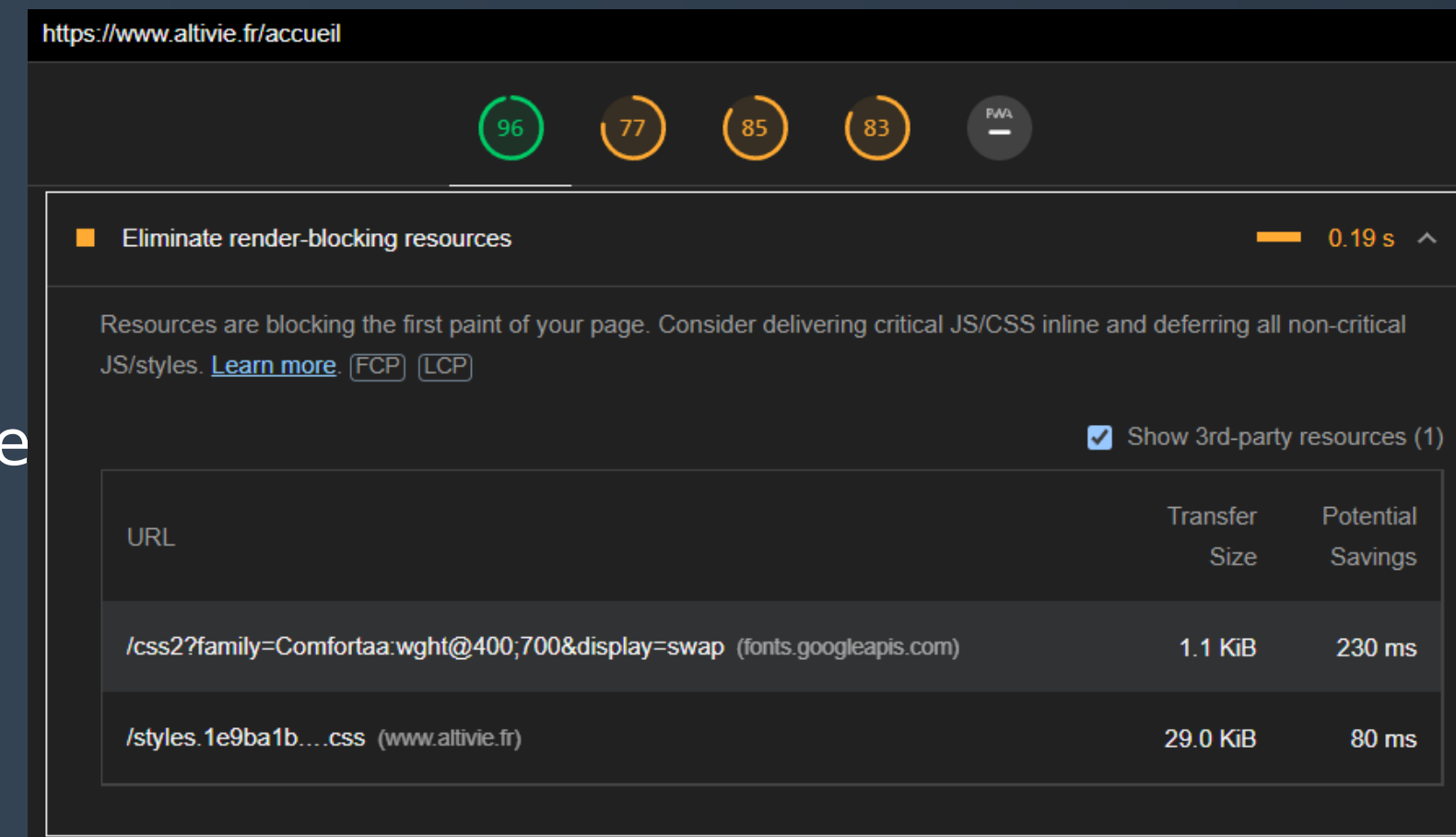
# Improve the LCP score (1/3)

Make sure your resources are **rendered quickly**.

- use SSR or SSG when possible for a faster load

- use progressive loading techniques

    *ex: lazy-loading on images, blurred placeholders*

- load JS/CSS in a way that is not blocking the HTML parse

    *ex: CSS / JS code splitting, load critical files first*



*PageSpeed Insights report: Render-blocking resources section*

Efficiently load 3rd party JS

Defer non-critcal CSS

Optimize 3rd-party JS (lab tutorial)

# Improve the LCP score (2/3)

Make sure your resources are **discovered quickly**.

- load resources direclty from the main HTML document

    *ex: inline fonts, inline critical CSS*
- use resource hints to set the loading priority for important resources

    *ex: preconnect, prefetch, prerender on links, see W3 Resource Hints*



*Adding resource hints & async/defer to improve page load performance*

# Improve the LCP score (3/3)

Make sure your resources are **sent quickly**.

- Use CDNs (closer to the edge user, faster response times)
- Optimize server response times
- Use compression for resources

  *ex: gzip / brotli compression formats*

- Optimize images and use newer formats

  *ex: WebP, AVIF (not fully supported)*

niji

# Improve the FID score

- avoid loading and running heavy Javascript during the initial page load

- use **code splitting** & **lazy-loading**, remove dead code, analyze your **bundle sizes** and 3rd party code

- use LightHouse to identify scripts running **long main-thread tasks** and the **Total Blocking Time** metrics
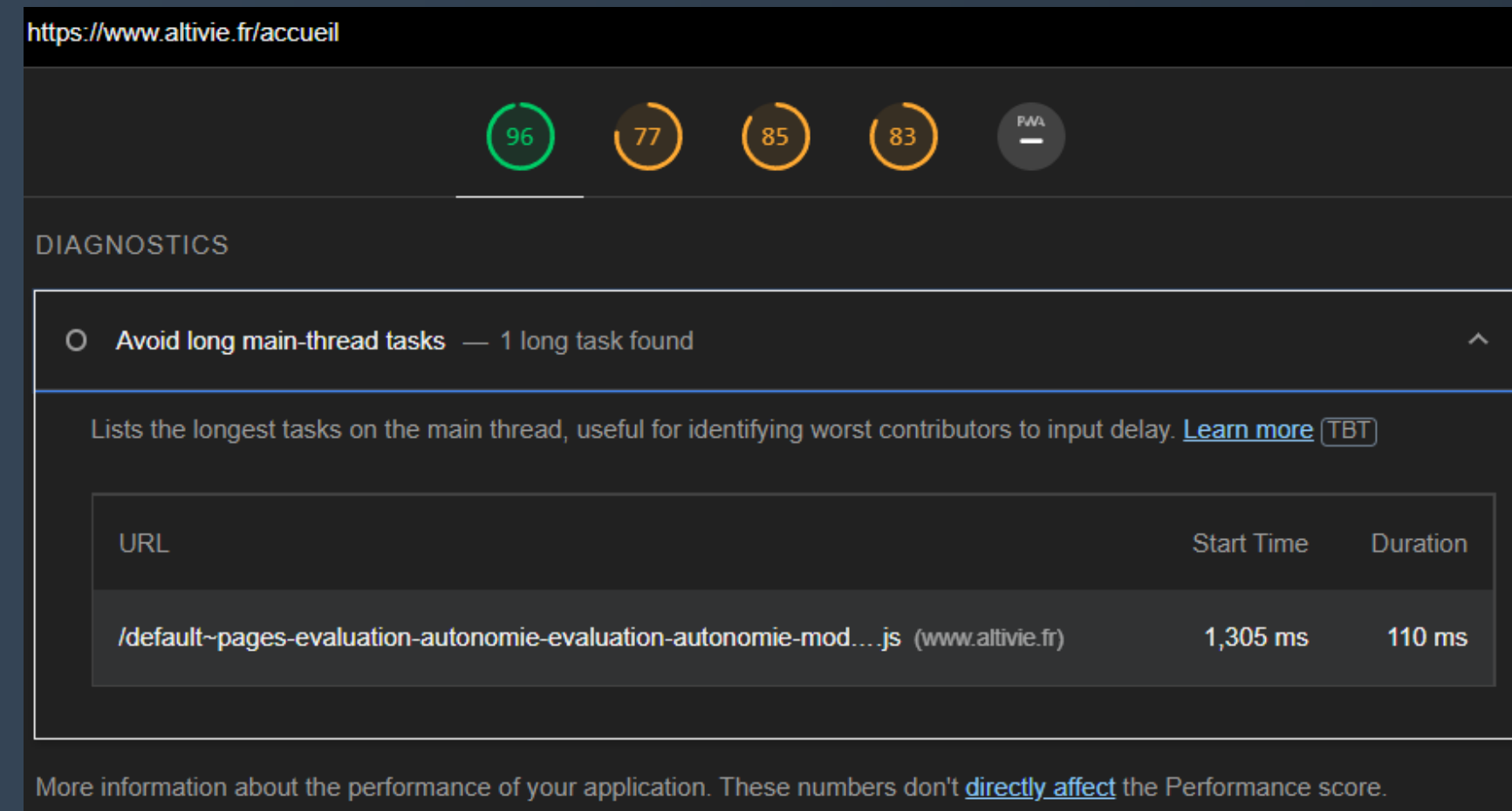


*PageSpeed Insights report: "Avoid Long Main Thread Tasks" section*

# Improve the CLS score

- Set **width/height atttributes** on all images, videos, iframes

- Avoid CSS animations that cause layout shifts

- Don't add any element above the loaded content unless its a response to the user interaction
  - *ex: opening a filter box or a menu*

- Use LightHouse's *"Avoid large layout shifts" section* to identify elements creating a layout shift
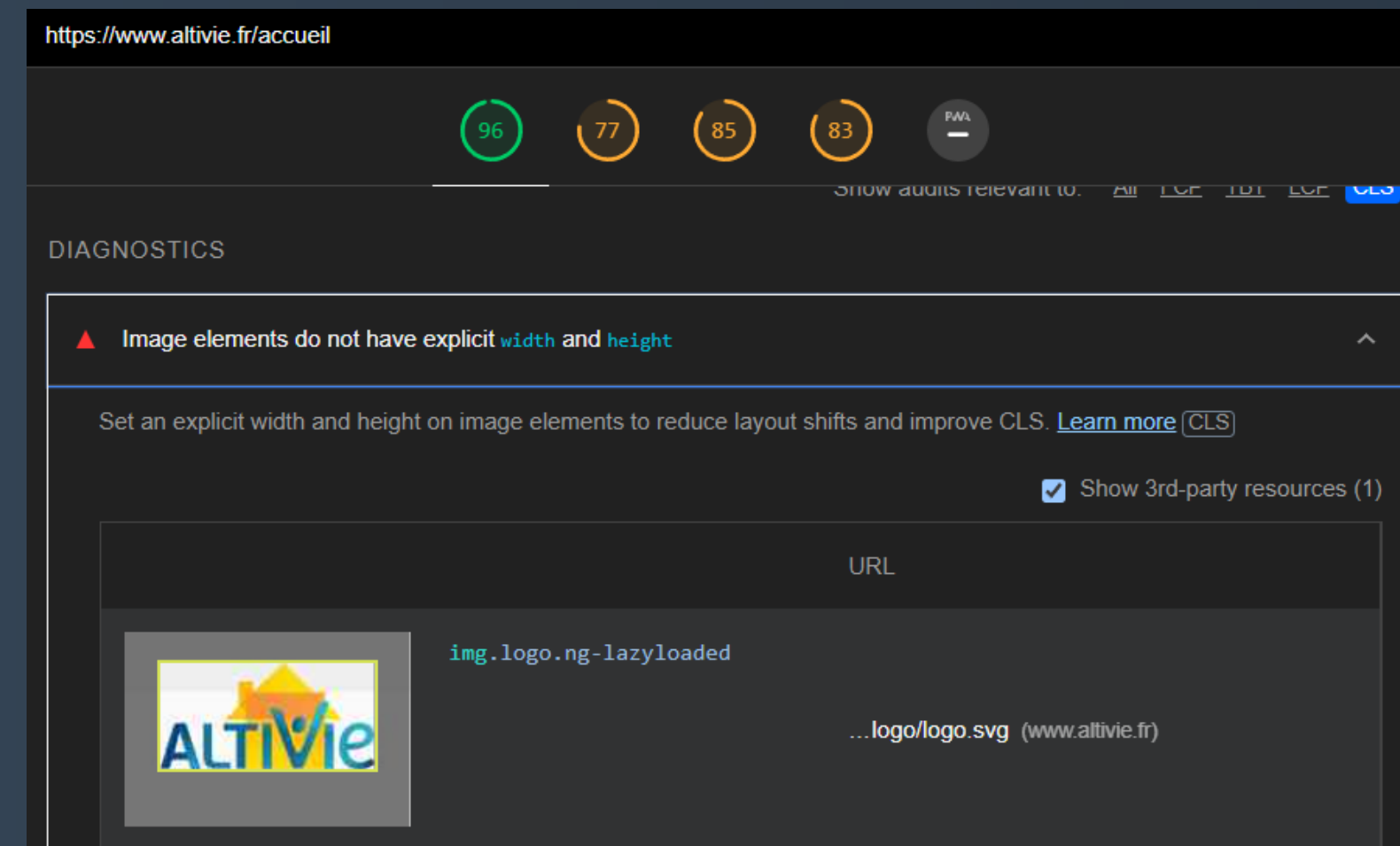


*PageSpeed Insights report: CLS diagnostic on images*
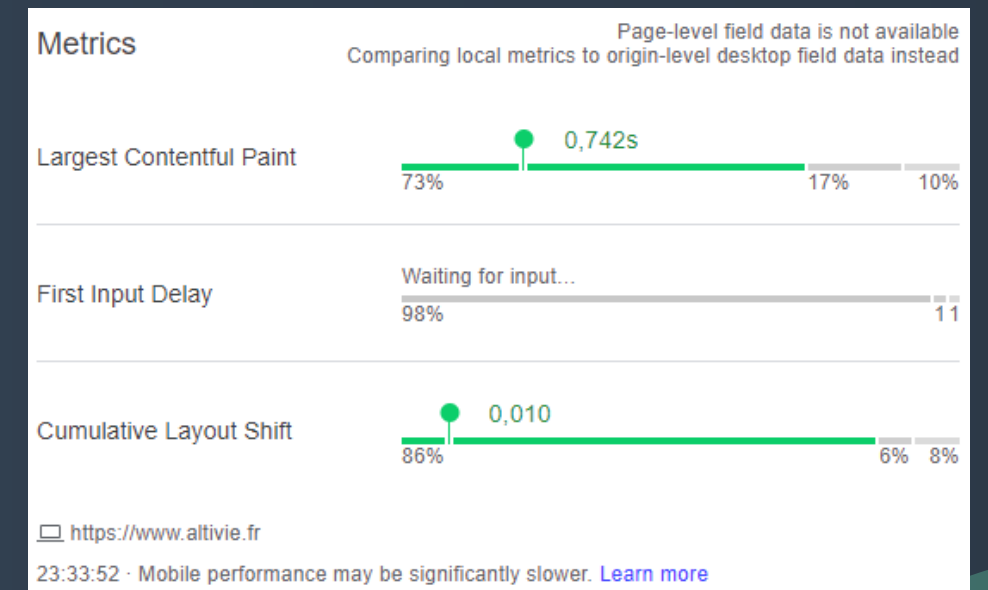
# Measuring Web Vitals

# How to measure the vitals?

- Page Speed Insights - simplest way, get a full report

- Web Vitals Chrome Extension - during navigation, in your browser

- LightHouse in ChromeDevtools - generates lab data

*Web Vitals via Chrome Extension*



Others:

- Google Search Console

- Web-Vitals library - npm package (145 kB)

Web Vitals Report - web app with Google Analytics plugin

*Logging Web Vitals within the code with WebVitals package*

```
import { getLCP, getFID, getCLS } from 'web-vitals';

//
getCLS(console.log);
getFID(console.log);
getLCP(console.log);
```

# Pagespeed Insights
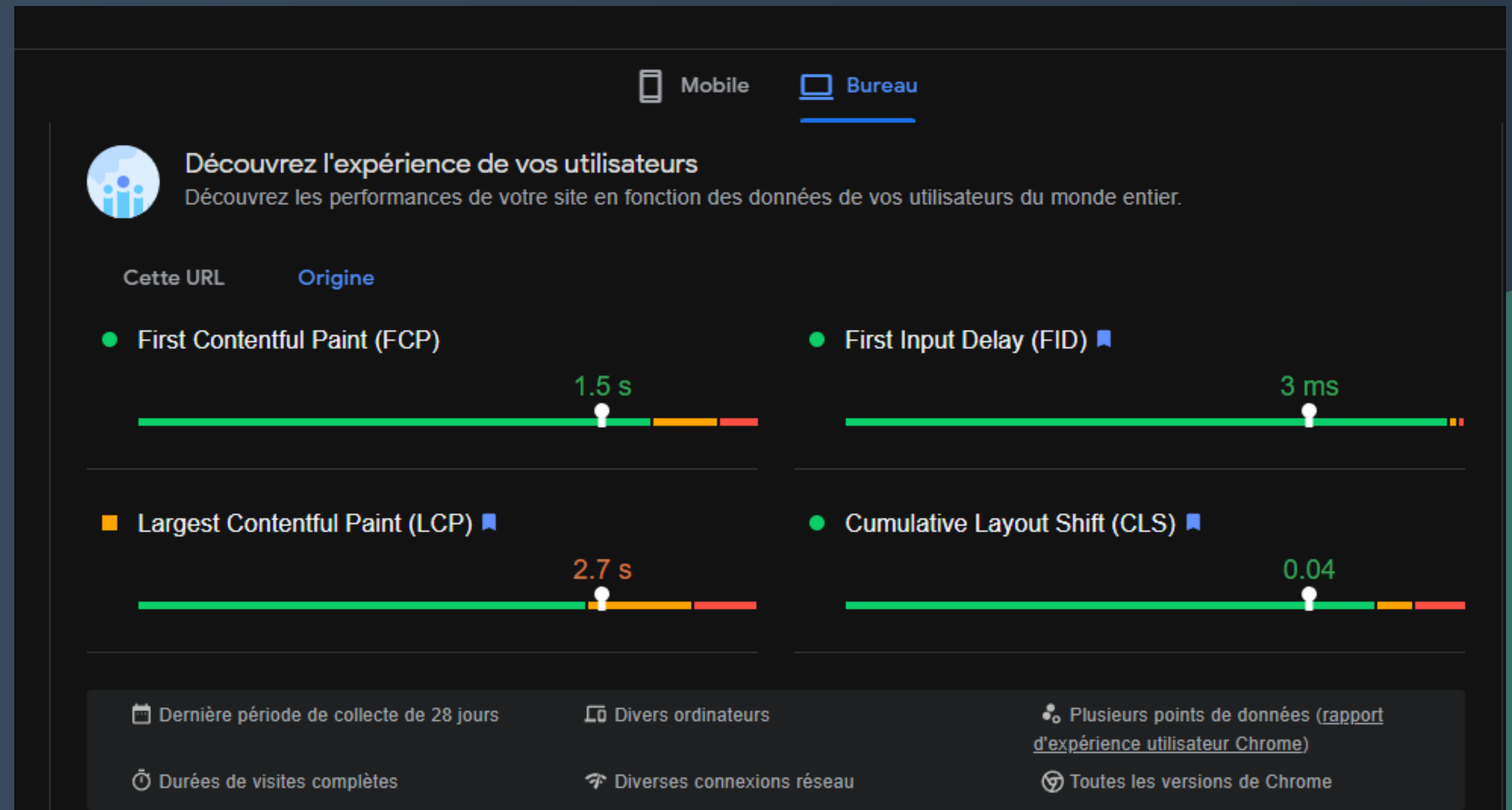
*Pagespeed report: Lab data (~LightHouse)*

- LCP, TBT (equivalent to FID), CLS
- Time To Interactive (TTI, page fully interactive), First Contentful Paint, Speed Index (how quickly the content appears)
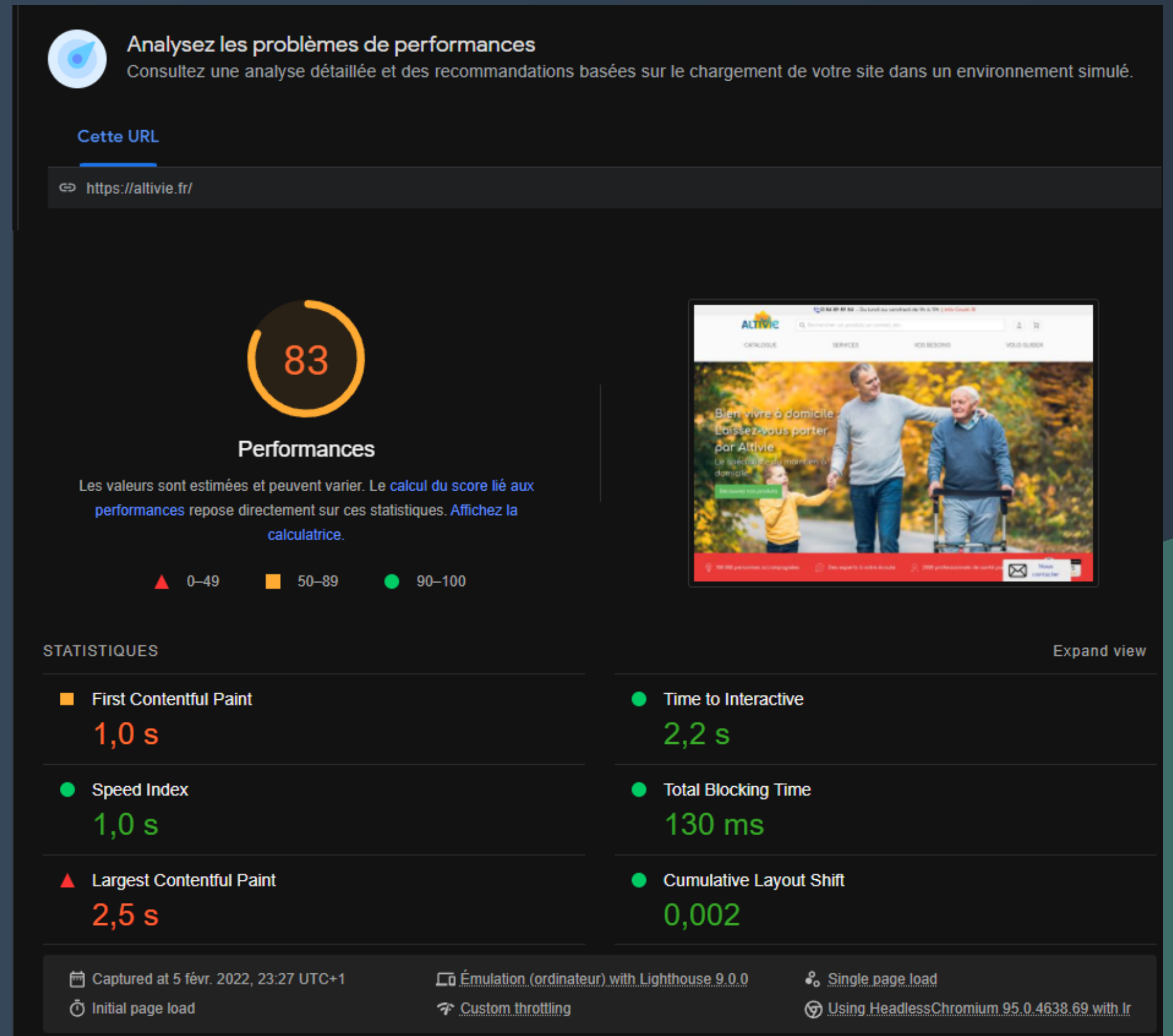
# Pagespeed Insights

*Pagespeed report: real users data*

- FCP, FID, LCP, CLS

# Resources

## Web.dev Google Documentation

- Vitals
- Learn Web Vitals
- Metrics
- Improving load times
- Web Vitals patterns

## Blog posts

- Efficiently loading 3rd-party JS
- Defer non-critical CSS

## Videos

- Measure what matters 22/03/2021, Google Chrome Developers channel (8min)
- News on Core Web Vitals 18/05/2021, Google Chrome Developers channel (14min)

## Others / Tools

- Page Speed Insights
- GitHub Web-vitals library
- Web.dev - Lab example on improving 3rd-party JS

niji