How and why ($) to improve web performance

Practical tips for 2023

Dev Romagna, March 22, 2023

cognizant
netcentric

cognizant netcentric

1 second

loading time improvement

-14%

users leaving the website at landing

(bounce rate)

+13%

users reaching website goals

(conversion rate)

Source: How Renault improved its bounce and conversion rates by measuring and optimizing Largest Contentful Paint

BCE ECB EZB EKT EKP 2002
500
500 EURO
EURO
200

Takeaways

- Basic understanding of web performance
- How web performance impacts on business
- Actionable improvements for web performance

Andrea Verlicchi

- Lots of websites
- Front-end development
- Web performance consultant

Experts
2023

Google Hackathon, London

Roadmap

1. Web performance, Search Engine Optimisation, and business impact
2. Measuring web performance

   Quiz game!

3. Improvement tips for 2023

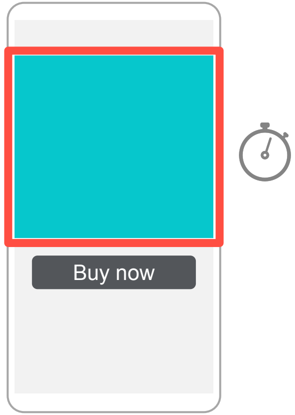
1

Web performance, SEO, and business impact

cognizant
netcentric

Google Core Web Vitals

Loading

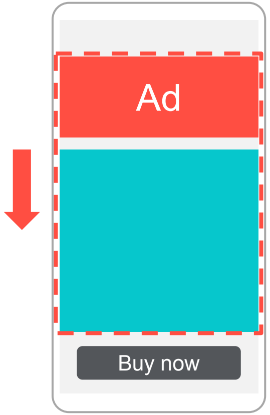


Largest Contentful Paint (LCP)

Good | Ok | Poor

2.5 s | 4.0 s

Visual Stability

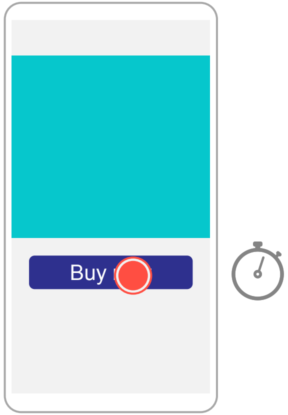


Cumulative Layout Shift (CLS)

Good | Ok | Poor

0.10 | 0.25

Interactivity

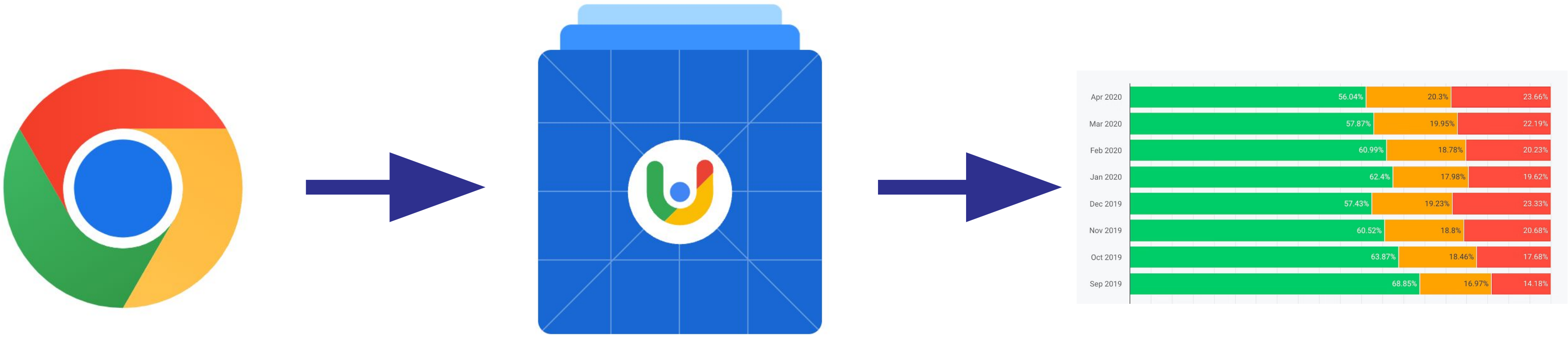


First Input Delay (FID)

Good | Ok | Poor

100 ms | 300 ms

How does Google know?

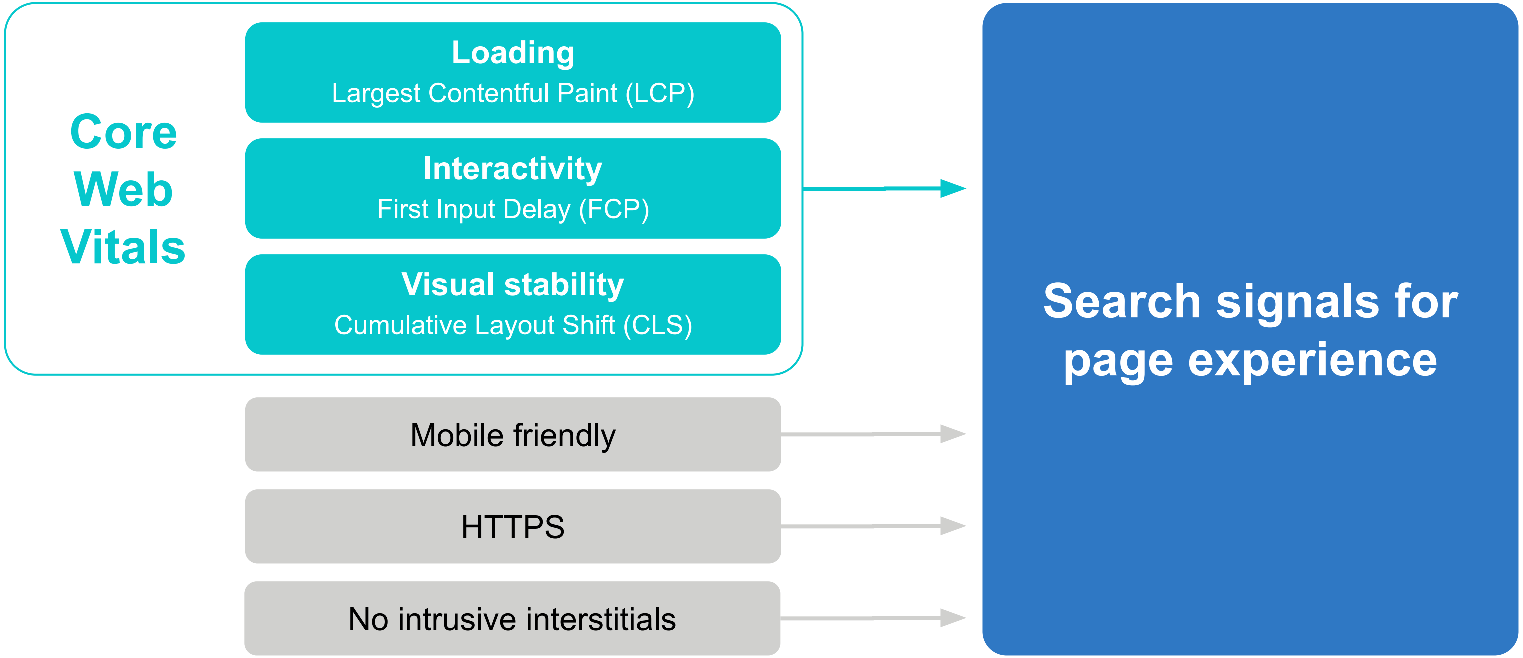


Google Chrome

Chrome User Experience
(CrUX)

CrUX Report

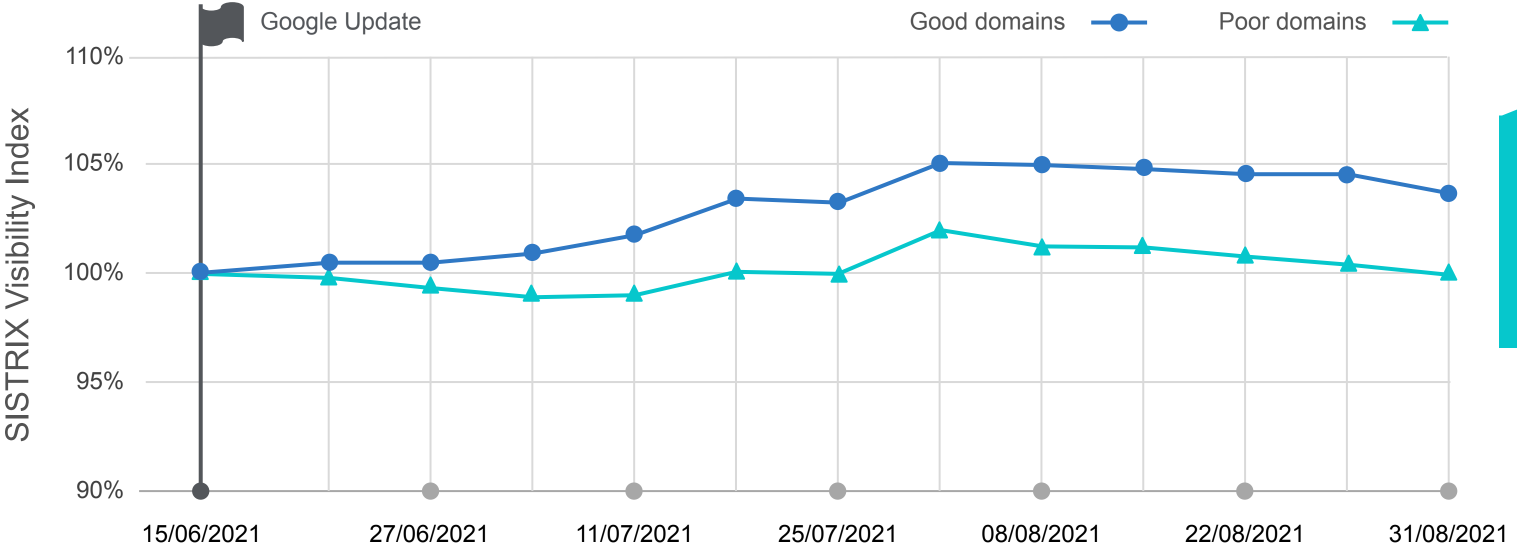
The Core Web Vitals impact your Google Search ranking



Core Web Vitals
Loading
Largest Contentful Paint (LCP)
Interactivity
First Input Delay (FCP)
Visual stability
Cumulative Layout Shift (CLS)
Mobile friendly
HTTPS
No intrusive interstitials
Search signals for page experience

Source: https://developers.google.com/search/blog/2020/11/timing-for-page-experience

The Core Web Vitals impact the Google search ranking



4%

Increase in ranking

Source: https://www.sistrix.com/blog/core-web-vitals-is-a-measurable-ranking-factor/

Success = Traffic × Conversion

Good performance

High conversion + low traffic cost

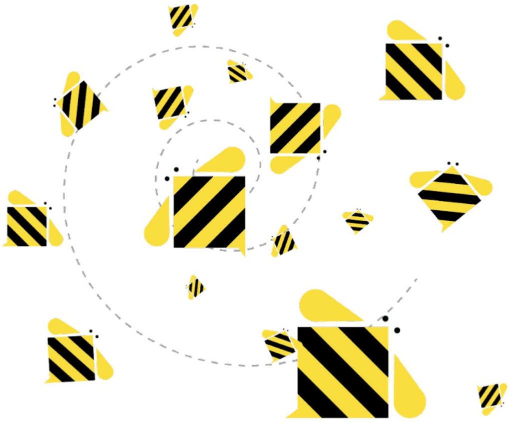
Low conversion + high traffic cost

Poor performance

2

Measuring web performance

Field data



The big picture

Lab data

Detailed view

Field data

- Monitor page navigations as they are happening
- Real users, devices, connections, locations
- See data in a dashboard

Spot problems

Lab data

- Test a given URL
- Emulated speed, single location
- Check results immediately
- In-depth analysis, video

Investigate causes, test resolutions

Field data

Chrome User Experience Report (CrUX)

SpeedCurve

mPulse

cognizant web vitals monitoring

Lab data

Pagespeed insights

Lighthouse

WebPageTest by catchpoint

Quiz game!

Here is when I opened Kahoot :)

3

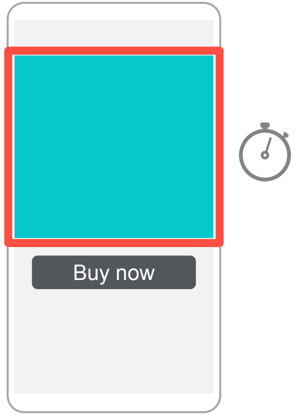
Improvement tips for 2023

cognizant netcentric

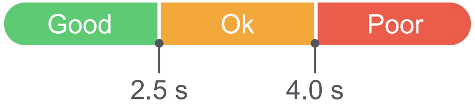
- Largest real-world impact
- Relevant and applicable to most sites
- Realistic to implement

Google Core Web Vitals

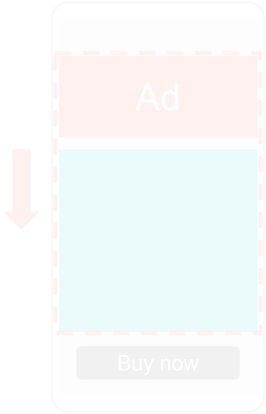
Loading



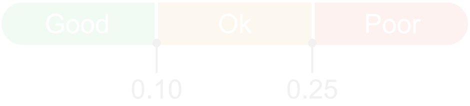
Largest Contentful Paint (LCP)



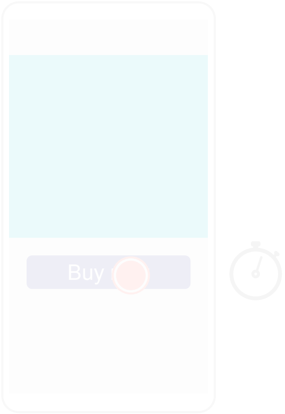
Visual Stability



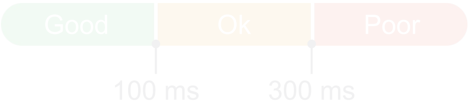
Cumulative Layout Shift (CLS)



Interactivity



First Input Delay (FID)



Ensure the LCP resource is discoverable 🔍 from the HTML source

72% of mobile pages

LCP element = an image

39% of those images

not discoverable from HTML source

Sources: Web Almanac

- Load the image using an `<img>` element with the `src` or `srcset` attribute
- Prefer server-side rendering (SSR) over client-side rendering (CSR)
- If your image needs to be referenced from an external CSS or JS file, include it in the HTML source via a `<link rel="preload">` tag.

Ensure the LCP resource is prioritized 🔺

- Add `fetchpriority="high"` to the `<img>` tag of your LCP image
- Never set `loading="lazy"` on the `<img>` tag of your LCP image
- Defer non-critical resources when possible

Use a CDN 🌐 to optimize document and resource server-time (TTFB)

Serve your content as geographically close to your users as possible.

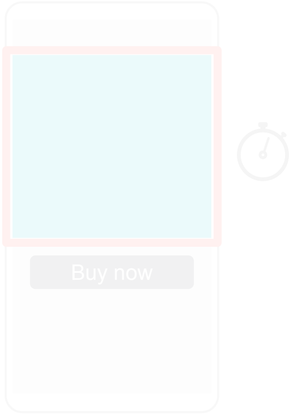
Cache that content so recently-requested content can be served again quickly.

Sources: Web Almanac

- Increase how long content is cached for.
- Cache content indefinitely, and purge the cache on updates.
- Move dynamic logic from your origin server to the edge

Google Core Web Vitals

Loading

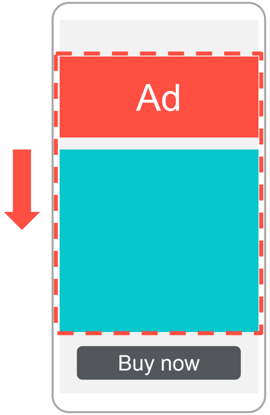


Largest Contentful Paint (LCP)

Good | Ok | Poor

2.5 s | 4.0 s

Visual Stability

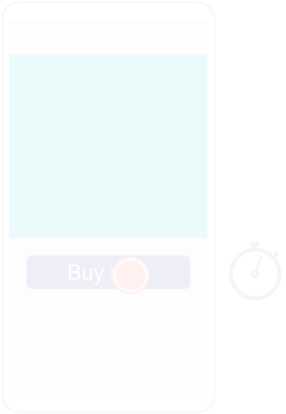


Cumulative Layout Shift (CLS)

Good | Ok | Poor

0.10 | 0.25

Interactivity



First Input Delay (FID)

Good | Ok | Poor

100 ms | 300 ms

cognizant netcentric

Set explicit sizes 🖼️ on any content loaded from the page

0px initial default height

for unsized images

72% of pages

have at least one unsized image

Sources: Web Almanac,

- Explicitly set `width` and height attributes (or equivalent CSS properties) on images
- Use the `aspect-ratio` CSS property to reserve space for other lazy loaded content (ads, embedded videos, etc.)
- If aspect is unknown, use `min-height`

Ensure pages are eligible for back/forward cache (bfcache) 💭

35% of pages

ineligible for the bfcache

Sources: Web Almanac

- Check if your pages are eligible for the bfcache using bfcache tester in DevTools
- Work on the reasons why they are not

Sources: bfcache tester in DevTools

Avoid animations and transitions 🎞️ that use layout-inducing CSS properties

15% less likely to have "good" CLS if you animate any CSS property that could affect layout

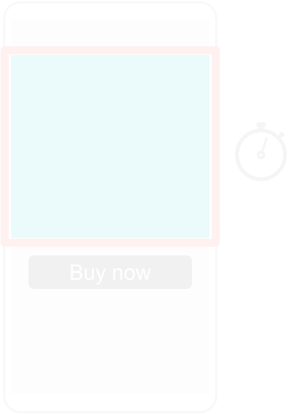
Absolutely positioned elements that animate top or left will cause layout shifts

Sources: Web Almanac, Optimize LCP / Make it discoverable

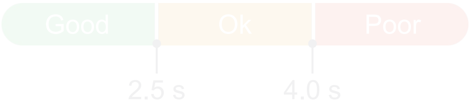
- Never animate or transition CSS properties that require browsers to update page layout
- Instead of animating `top` or `left`, animate `transform:translateX()` or `transform:translateY()`

Google Core Web Vitals

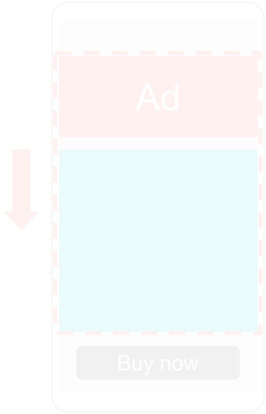
Loading



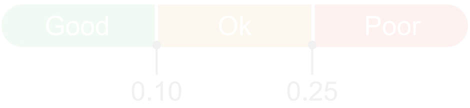
Largest Contentful Paint (LCP)



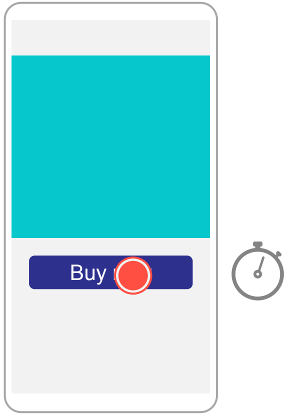
Visual Stability



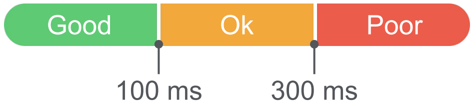
Cumulative Layout Shift (CLS)



Interactivity

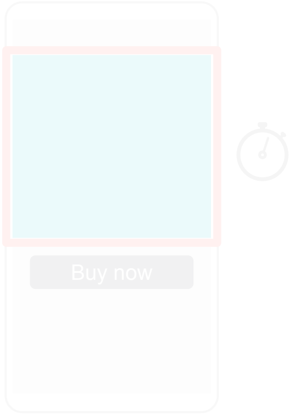


First Input Delay (FID)

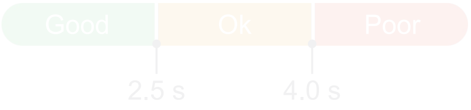


Google Core Web Vitals

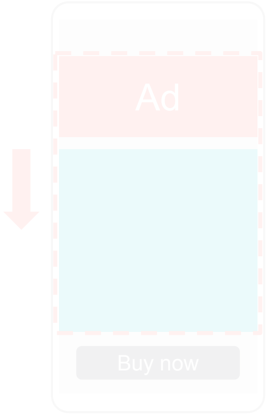
Loading



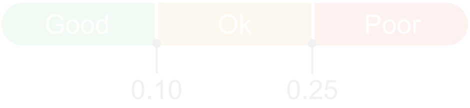
Largest Contentful Paint (LCP)



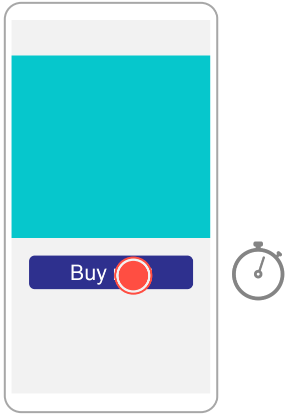
Visual Stability



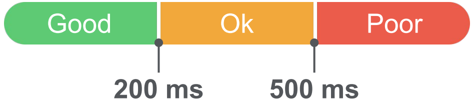
Cumulative Layout Shift (CLS)



Interactivity



Interaction to Next Paint (INP)



Avoid or break up 💥 long tasks

Tasks include rendering, layout, parsing and compiling and executing scripts.

50ms of main-thread blocking threshold for a task to be considered “long”.

19 is the median number of long tasks on mobile

Sources: Web Almanac

- Break up long tasks into smaller ones, yielding often to the main thread
- Consider using APIs such as `isInputPending` and the Scheduler API.

Avoid unnecessary 🟨 JavaScript

460 kb / page

median of JS code served to each page

This Javascript

creates an environment where tasks are competing for the main thread’s attention

Sources: Web Almanac, Optimize LCP / Make it discoverable

- Use the coverage tool in Chrome DevTools to find unused code
- If unused because it will be used later, move to separate bundle - code splitting
- Using a tag manager? Periodically check your tags.

Avoid large 🐙 rendering updates

Javascript is not the only thing that can affect your website's responsiveness.

Rendering can be expensive and can interfere with your website's ability to respond to user inputs.

Sources: Web Almanac, Optimize LCP / Make it discoverable

- Avoid using `requestAnimationFrame()` for doing any non-visual work.
- Keep your DOM size small.
- Use CSS containment (`contain` property)

Let’s wrap up

https://web.dev/top-cwv-2023/

cognizant®
netcentric

Need an expert?

webperformance@netcentric.biz