

Tuning Google's Core Web Vitals in RiSE/EMS

Presenter: Tipton Ramsay



Understanding Core Web Vitals

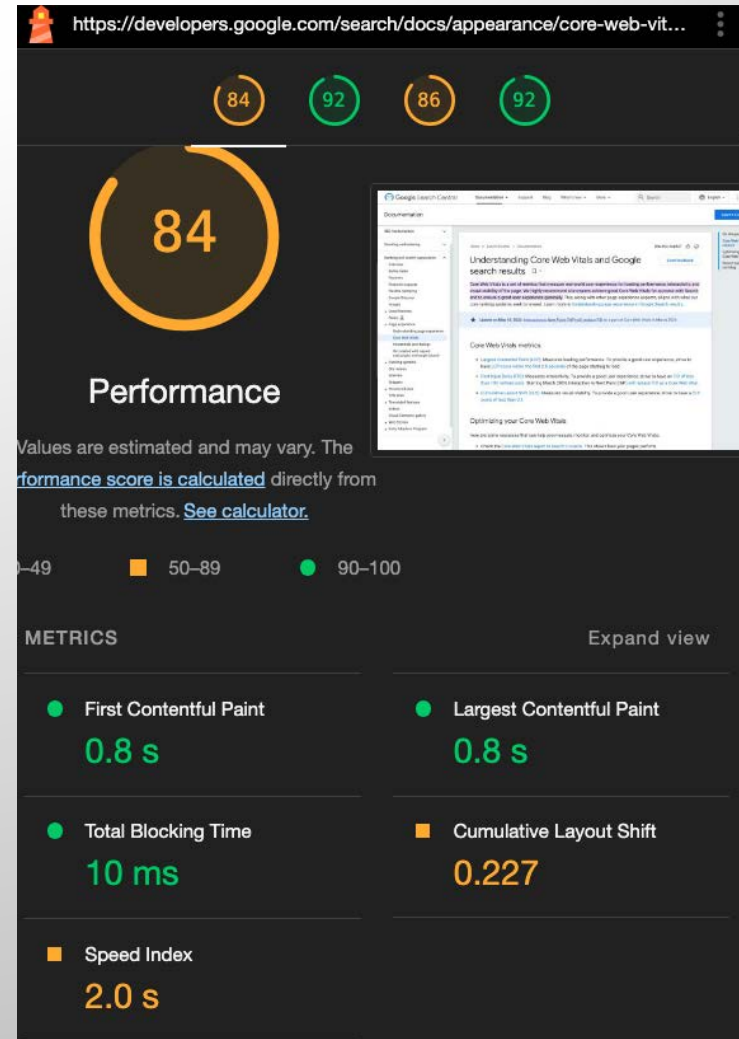
From [developers.google.com](https://developers.google.com/search/docs/appearance/core-web-vitals):

“**Core Web Vitals** is a set of metrics that measure real-world user experience for **loading performance**, **interactivity**, and **visual stability** of the page.”

“We highly recommend site owners achieve good Core Web Vitals for success with search and to ensure a great user experience generally. **This, along with other page experience aspects, aligns with what our core ranking systems seek to reward.**”

Learn more at <https://developers.google.com/search/docs/appearance/page-experience>.

- Do the best you can – **Look at Google’s own page scores!**
- Keep cost/benefit in mind
- Some suggestions can’t be done in the context of your environment or within your level of access.



Google's Take

- Google frames this all as “looking to reward content that provides a good page experience” and encouraging site owners and content developers to **not just focus on only 1 or 2 aspects of the user experience.**
- Encourages “**Self Assessment**”. If you can answer yes to the following questions you're probably in good shape.
 - Do pages have good Core Web Vitals? **THAT'S WHAT YOU'RE HERE FOR!**
 - Are pages served in a secure fashion? **HOPE SO!**
 - Does content display well for mobile devices when viewed on them? **DO YOU KNOW? HAVE YOU TESTED?**
 - Does the content have an excessive “ads” that distract from or interfere with the main content? **MAINTAIN PROPER VISUAL HIERARCHY**
 - Do pages have excessive or intrusive interstitials? **THINK POP UPS OR MODAL WINDOWS**
 - Yada yada yada
- Like SEO, this will be an **ongoing adherence to a set of best practices and not a bunch of set and forget tasks.**

What Actually Are The “Core Web Vitals”?

- **Largest Contentful Paint (LCP):** Measures loading performance. *How quickly the page loads.*
- ***First Input Delay (FID):** Measures interactivity. *How quickly a user can start interacting with the page.*
*Starting March 2024, Interaction to Next Paint (INP) will replace FID (First Input Delay).
- **Cumulative Layout Shift (CLS):** Measures visual stability. *Changes to layout, not initiated by the user, that can be annoying.*

Largest Contentful Paint (LCP)

The Largest Contentful Paint (LCP) is the amount time it takes to render the largest image or text block visible within the viewport (above the fold).

Page elements that impact LCP:

- Images and videos without explicit width and height attributes
(I know, I know. Total contradiction to the advice to make sure things are responsive. I will explain.)
- Elements with **background images** versus CSS colors/gradients
- **Block level** elements that change size as child elements are loaded, resized, or manipulated

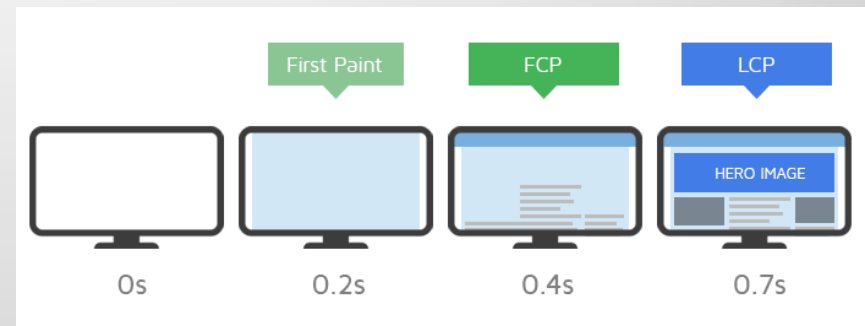


Image Credit: <https://gtmetrix.com/largest-contentful-paint.html>

First Input Delay (FID)

First Input Delay (FID) aims to quantify the user experience as they try to interact with an unresponsive page.

- FID is mainly affected by the process and timeline of **how a page loads content**, linked assets, and runs local scripts.
- Long first input delays typically occur between First Contentful Paint (FCP) and Time to Interactive (TTI) because the page has rendered some of its content but isn't yet reliably interactive.

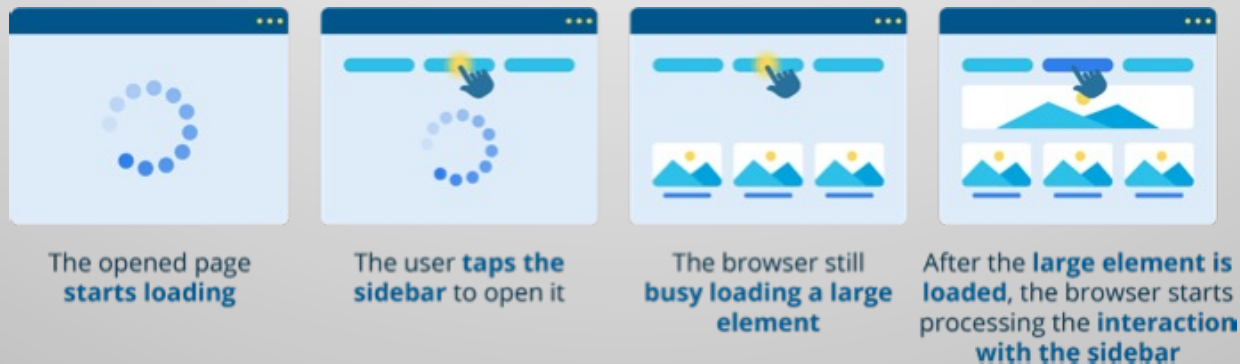


Image Credit: <https://product.webpagetest.org/core-web-vitals/first-input-delay>

Cumulative Layout Shift (CLS)

Cumulative Layout Shift (CLS) measures visual stability and how often a user might experience an unexpected shift in the layout of the page.

We've all experienced this!

- Example: Click a link from google > page loads > we start reading > suddenly an ad appears pushing content down and we've now lost our place.
- Example: Page loads > we see the link we want > go to click it > layout shifts suddenly and you end up clicking on the wrong link.

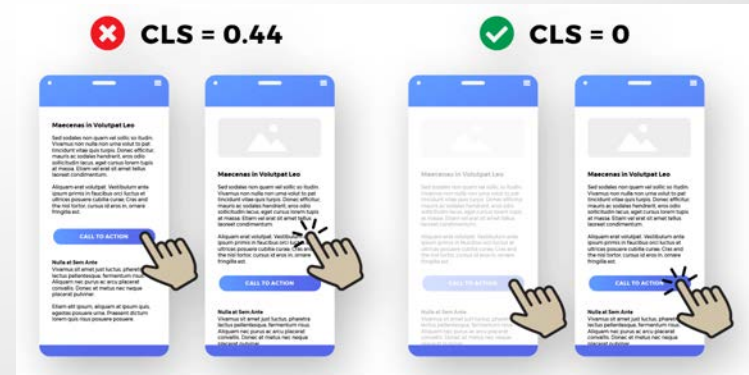


Photo Credit: <https://parachutedesign.ca/blog/cumulative-layout-shift-cls/>

Practical Approach to Managing

Take this all as “best case”, or “in a perfect world”.

With access limitation of a given platform, the inclusion of code libraries like Bootstrap, jQuery, etc. and prerender page manipulates performed by these included libraries, **many of the recommendations are out of our control.**

We probably can't get perfect scores, but that should not be the goal. The goal is to be in the driver's seat and not a helpless passenger. **Do all you can...** and constantly re-evaluate.

Note mobile vs. desktop results. Google PageSpeed Insights uses an "emulated 3G network" for mobile testing.



Specific Limitation of RiSE/EMS

Below are Script's and CSS files that are automatically included in your site. There is no way to remove or selectively include **without risk of error**.

Included Script libraries

- Modernizr.min.js
- Jquery.min.js
- jquery-migrate.min.js
- jquery-ui.min.js
- WebResource.axd?d=pynGkmcFUV13...
- Telerik.Web.UI.WebResource.axd?_TS...
- Asi.js?v=20402048

Included CSS libraries

- 10-UltraWaveResponsive.css?v=734064678
- 99-GreatBarrierReef_Responsive.css
- WebResource.axd?d=QKX1AO1vX8ebWK...
- WebResource.axd?d=_s8C6V0hVmZ50IR...
- css?family=Raleway|Abel

FCP/LCP Recommendations

(First Content Paint/Last Content Paint)

- Eliminate render blocking resources
 - **REALITY:** The “resources” in this case are the scripts being automatically included by the CMS. But good to be aware and not make the problem worse when including custom scripts.
- Reduce unused CSS
 - **REALITY:** The base theme includes a ton of CSS that the system might need for specific use cases. We don't want to remove in case a certain page ends up needing it. Also, the cost/benefit to devise a method to selectively include CSS only on pages where it is used would impact the average user in a way that would not likely be worth the minor gain it would achieve.
- Reduce unused JavaScript
 - **REALITY:** Same as CSS. But might be something consider when custom scripting. Are you carrying around scripts globally that could be selectively included. But again, the impact on ease of use for content editors needs to be considered.

FCP/LCP Recommendations

(First Content Paint/Last Content Paint)

- Minify CSS and JavaScript
 - **REALITY:** We have no way to do this for items automatically included. And doubtful that any custom CSS added is large enough to see any meaningful gain.
- Ensure text remains visible during webfont load
 - **REALITY:** Make sure a fallback system fonts is included in font-family CSS declarations. For example: font-family: "Custom Webfont", sans-serif;
- Initial server response was too long.
 - **REALITY:** Completely out of your control. But good to make sure your host provider is not dropping the ball.

FID Recommendations

(First Input Delay)

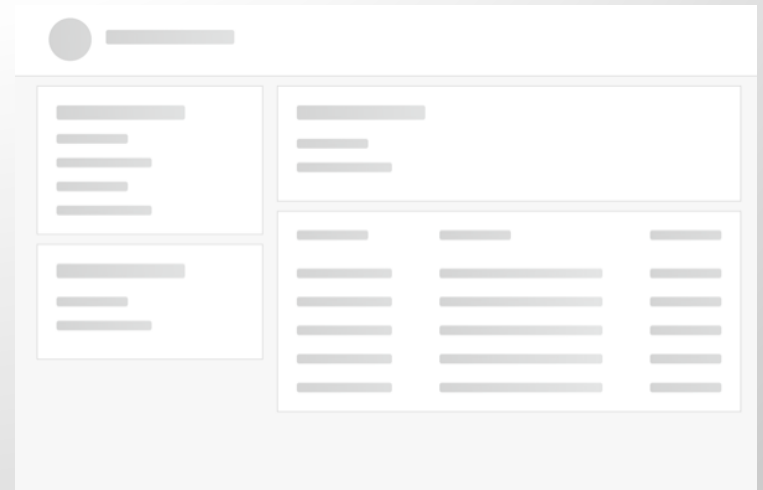
- Reduce JavaScript execution time
- Minimize main thread work
- Reduce the impact of third-party code
- Avoid excessive DOM size
- Avoid serving legacy JavaScript to modern browsers

REALITY: As long as other best practices (clean HTML, SEO, image optimizations, etc.) are being followed, these items are outside our control and should not need attention.

CLS Recommendations

(Cumulative Layout Shift)

- Add explicit width="" and height="" attributes on image elements
 - **REALITY:** This practice will help avoid CLS, but does not reduce the need for pre-processing large images prior to uploading. Make sure content editors understand this.
- Avoid large layout shifts
 - **REALITY:** Be aware and troubleshoot specific issues. Some might be caused by third party widgets, e.g. paid advertising.
- If animating page elements use CSS transform property
 - **REALITY:** Animations are likely relying on a library for this. Consider updating or selecting new.



Best Practices = Cheap Insurance

- Resize and optimize images prior to unload
- Add height and width attributes on images
- HTML attributes to help set priority
 - loading="lazy": Lazy load images below "the fold"
 - fetchpriority="": low, high or auto for <link> tags
(not supported by all browsers)
 - preload="": for media files
 - async / defer : for <script> tags
- Understand loading order and avoid bottlenecks
- Know what included third-party scripts are doing
- Be smart with copy/paste solutions – only what the page needs
- Avoid bloat where possible.
- **NOTE:** Lighthouse "Best Practices" have gotten very technical. Do what you can!

Tools To Test Your Site

- Chrome Browser Developer Tools
 - **Lighthouse** tab
 - **Performance** tab
- Web based Page speed insights
 - <https://pagespeed.web.dev/>
- **REMEMBER**, these test a single page. Not the entire site.

BEST PRACTICES

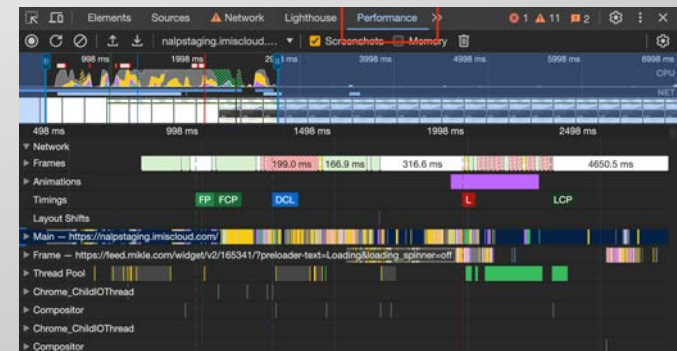
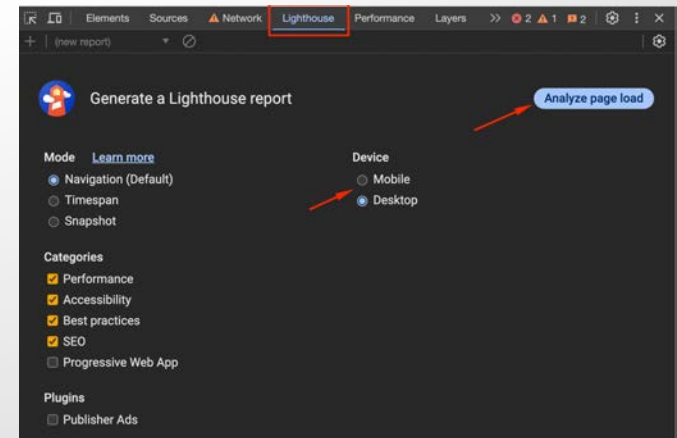
to avoid unnecessary issues

Due Diligence

to identify where problems exist

Trial and Error

to test and select effective solutions



Q & A

Please feel free to contact us with any follow-up questions.

info@atsol.org

www.atsol.org

720.945.7252

