A WHITEPAPER BY CLASSY LLAMA

GOOGLE'S NEW GAME

Get Ahead of Core Web Vitals

Presented by

CLASSY LLAMA

Austin Hovey, Sales Development
417.866.8887
sales@classyllama.com

CONTENTS

INTRO	What is the Core Web Vitals Update and why should I care?	3
PART 1	What do I need to know about Core Web Vitals?	4
PART 2	What does this mean for me?	6
PART 3	How can I find out what my site needs for Core Web Vitals?	7
PART 4	How do I optimize my website for Core Web Vitals?	8
CONCLUSION	Where do I go from here?	15

The Rules are Changing

What are Core Web Vitals why should I care?

So, Google is changing up its search ranking game ... again.

Starting in mid-June 2021, Google will start rolling out a change in the way it ranks sites in search results pages that includes a set of metrics Google is calling Core Web Vitals, which are part of a set of seven "search signals" related to page experience.

The idea is basically this:

Google wants to reward websites that customers enjoy browsing so everyone will make better websites.

Sounds great, right? Well, yes and no.

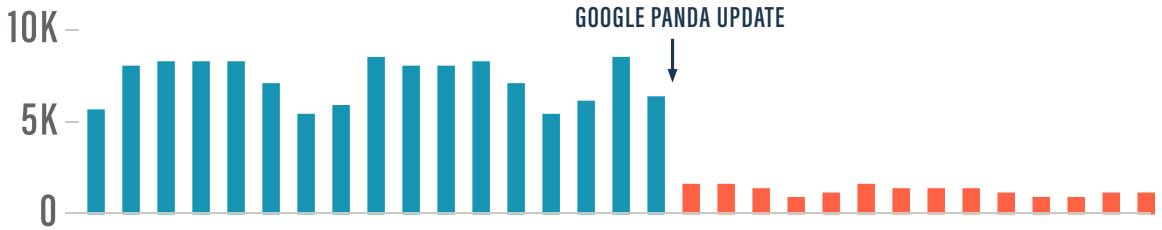
The problem with this algorithm shift is if your site doesn't meet Google's standards for page experience by optimizing your Core Web Vitals scores, you could lose traffic and even sales. In fact, over the years, we've seen several big updates to Google's algorithm that threw a huge wrench in traffic and sales for lots of online businesses. It was so painful to watch site owners suffer those consequences that we're super serious about stopping as many businesses as we can from going through similar problems this time around.

For example, Google's Panda update in 2014 was particularly brutal. Google admitted to over 500 changes that caused many sites' rankings to drop off a cliff, and their traffic and revenue followed suit.

While it's great that Google's focus on creating enjoyable online experiences paves the way for better websites all across the internet, **historically algorithm updates have really damaged a lot of businesses ... we don't want you to go through that!**

That's why we've put this guide together: To help you get ahead of Google's game and make sure your business doesn't lose any momentum (and possibly gains some ground by doing work your competitors aren't!).

Read on to find out more about what Google's Core Web Vitals update is, what it means for you, and what to do about it next.



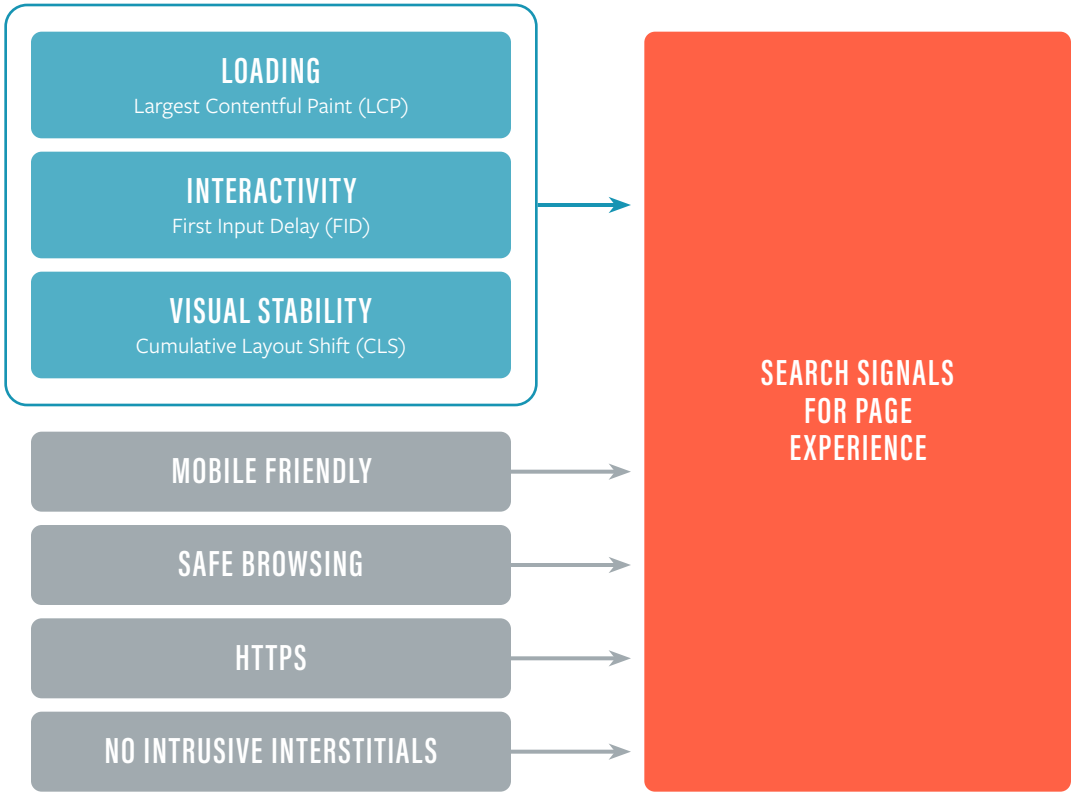
PART 1

The Vital Details

What do I need to know about Core Web Vitals?

Thankfully, Google is being pretty cool about this update and is sharing lots of information (and fun, colorful graphs) to keep us all in-the-know. Sometimes their way of explaining things can be a little ... um ... academic, to put it nicely, which is why we're gonna break it down for you in the simplest terms possible.

So, first of all, there are three metrics known as Core Web Vitals that are the main pillars of this update. Overall, the way Google ranks for page experience takes some other signals into account (as you can see in the graphic below), but the three Core Web Vitals are the new kids on the block.



What are these “vitals”?

Google gave them some fancy, technical names to help us SEO pros sound smart, but it’s really pretty simple. Core Web Vitals are all about these three things:

Loading

How fast does your website show up on the screen?

LCP

Largest Contentful Paint

Less than 2.5 sec

More than 4.0 sec

This metric measures loading performance. To provide a good experience, the main content on your page should load within 2.5 seconds.

Interactivity

How fast or easy is it to click around on your website?

FID

First Input Delay

Less than 100 ms

More than 300 ms

This metric measures the time before interactivity is possible. To provide a good user experience, pages should have an FID of less than 100 milliseconds.

Visual Stability

Do elements on your website jump around unexpectedly?

CLS

Cumulative Layout Shift

Less than 0.1

More than 0.25

This metric measures visual stability, meaning elements on your page will stay in a consistent place while a user browses. To provide a good user experience, your web pages should maintain a CLS of less than 0.1.

The cool thing about these metrics is that:

1. Each of them are directly connected to a part of the user’s experience.

2. You can measure them “in the field” (testing real website visitors interacting on your site).

3. They reflect what it’s actually like to use your website.

Basically, the better your site experience is for users, the better you will rank—which is a win/win for you and your customers, no doubt.

PART 2

Good News / Bad News

What does this mean for me?

More good news:

Some of the standards that affect Google's new Core Web Vitals scores are things that good common sense already told you your website needs. For example, if you've already been optimizing your web pages for things like page speed, mobile friendliness, and great user experience, you're likely already ahead of the game.

If that's the case for you, it's possible that the impact of the Core Web Vitals will be small to nonexistent for your business.

Woohoo!

Not sure? You can run a check on your website to see if it aligns with the Core Web Vitals standards. If it does, you're good to go!

Woohoo x2!!

BUT (and this is a big "but"), **if your website hasn't been specifically optimized for user experience or if your site is built on a platform that isn't optimized, your business could be negatively affected**, and it'll continue to get worse the longer you wait to act.

Not only that, Google is also working on adding a marker to each website on its Search Result Pages that highlights websites with good page experiences.

If you don't make sure your page has good Core Web Vitals scores, your search results won't include that marker. Eventually, that'll negatively affect your click-through rates when people start choosing other websites that Google says have better page experiences.

The sooner you can find out the health of your website in the Web Vitals category, the better chance you'll have of keeping the rankings you have ... and possibly gaining ground against competitors who didn't read our guide.

PART 3

Taking Stock

How can I find out what my site needs for Core Web Vitals?

Google is being super cool and helpful here by providing five different tools to help you check and monitor your website for Core Web Vitals. All of these tools make it easy to understand your site's health through a green/yellow/red color-coded system, along with numbered metrics and explanations of what they mean (Thanks, Google!).

There's some overlap between the info you'll get from each of these, but check out this chart below to find the tool that best meets your need:

TOOL	WHAT IT IS	WHEN TO USE IT
Google Search Console https://search.google.com/search-console/	Within Search Console, a new Core Web Vitals report can evaluate pages across an entire site. The report is based on real-world data and identifies groups of pages that need help.	This tool is best used to get a high level overview of how your site is performing because the report will give specific examples of issues on some pages/URLs, not every single one.
Page Speed Insights https://developers.google.com/speed/pagespeed/insights/	This tool gives you a report on specific pages. Through analysis of real-world data, you'll also get advice on what to do to improve your scores on a particular URL.	This report is useful for checking on specific pages/URLs. Use it when you: • Only need to check loading speed. • Need an accurate report of loading times experienced by visitors.
Lighthouse https://developers.google.com/web/tools/lighthouse	Lighthouse differs from PageSpeed Insights in that it uses lab data only, rather than real-world data. In addition, Lighthouse goes beyond measuring loading times by delivering more than one score, providing SEO and accessibility information.	Use Lighthouse when you: • Want to run audits programmatically. • Need to evaluate other aspects of your website as well as load times. • Want to incorporate the Lighthouse API into your own systems.
Chrome DevTools Performance Panel https://developer.chrome.com/docs/devtools/	Chrome DevTools has been updated to help fix visual instability (CLS). Chrome DevTools also measures Total Blocking Time (TBT), which is useful for improving First Input Delay (FID).	This tool will give you the most detailed information, with a decidedly developer-oriented focus. Use this tool if you really want to get in the weeds of Core Web Vitals metrics, but remember **the information in this tool may not be 100% accurate** because there are no real users involved.
Web Vitals Extension https://bit.ly/3vor5Xf	A Chrome Web Vitals extension measures Core Web Vitals in real-time. Once enabled, the extension shows you scores for all three Core Web Vitals coded in red, yellow, and green.	This tool is best used to quickly gauge your site health because it only offers scores on the Core Web Vitals, rather than advice for improvement.

PART 4

Now Let's Fix It

How do I optimize my website for Core Web Vitals?

Here are the most common issues for each of the three Core Web Vitals and some ways you (or your developers) can fix them.

PART 4.1

How to Fix LCP

LCP (Largest Contentful Paint) is the Core Web Vital metric related to loading speed.

The most common causes of poor LCP are:

1. Slow server response times
2. Render-blocking JS and CSS
3. Slow Resource load times
4. Client-side rendering

1. Slow Server Response Time

ISSUES

The longer it takes a browser to receive content from the server, the longer it takes to display anything on the screen. (A faster server response time directly improves every page-load metric.)

FIXES

First, improve how and where your server handles your content. Use Time to First Byte (TTFB: the time it takes for a user's browser to receive the first byte of page content) to measure your server response times.

Optimize your server code: Are you running queries that take your server a significant amount of time to complete? Or are there operations happening server-side that delay the process to return page content? Analyzing and improving the efficiency of your server-side code will reduce the time it takes for the browser to receive the data.

Other solutions that improve slow server response times:

- Route users to a nearby CDN
- Cache assets
- Serve HTML pages cache-first
- Establish third-party connections early

(Feeling Lost? Feel free to use one of our audit services to demystify the tech-speak! Connect to us at ClassyLlama.com/contact to get started today.)

2. Render-Blocking JavaScript and CSS

ISSUES

Scripts and stylesheets are both render-blocking resources which delay FCP, and consequently LCP.

FIXES

To address this, remove any unused JavaScript or CSS, or defer any JavaScript and CSS that's **not** critical to speed up loading times for the main content of the page.

Ensure that only the minimal amount of necessary CSS and JavaScript is blocking rendering on your site with the following tips:

- Minify CSS and JavaScript ("minify" is cool tech-speak for "make it smaller or use less of it")
- Defer CSS and JavaScript that's not critical

3. Slow Resource Load Times

ISSUES

Unoptimized CSS or JavaScript are often to blame for slow load times, but there are many other types of resources that can also affect load times. The types of elements that affect LCP are:

- <img> elements
- <image> elements inside an <svg> element
- <video> elements
- An element with a background image loaded via the url() function (as opposed to a CSS gradient)
- Block-level elements containing text nodes or other inline-level text elements

The time it takes to load these elements if rendered above-the-fold will have a direct effect on LCP.

FIXES

There are a few ways to ensure these files are loaded as fast as possible:

- Optimize and compress images
- Preload critical resources
- Compress text files bigger than 1.4 KB (if compression will decrease size by at least 10%)
- Consider making CSS modular, so each page only loads what it needs.

Are you thinking, "Will someone wake me from this nightmare of developer speak?!" Then check out our audit services to pull the parachute cord.

Get connected at ClassyLlama.com/contact to get help.

4. Client-Side Rendering

ISSUES

Many sites use client-side JavaScript logic to render pages directly in the browser.

If you're building a site that is mostly rendered on the client, you should be aware of the effect it can have on LCP if a large JavaScript bundle is used. If optimizations aren't in place to prevent it, users may not see or interact with any content on the page until all the critical JavaScript has finished downloading and executing.

FIXES

When building a client-side rendered site make sure to:

- Minimize critical JavaScript
- Use server-side rendering
- Consider restructuring page to use server-side rendering

Tools for Measuring LCP

PageSpeed Insights
https://developers.google.com/speed/pagespeed/insights/

Search Console
https://search.google.com/search-console/welcome

Lighthouse
https://developers.google.com/web/tools/lighthouse

WebPageTest
https://www.webpagetest.org/

Chrome DevTools
https://developer.chrome.com/docs/devtools/

PART 4.2

How to Fix FID

FID (First Input Delay) is the Core Web Vital metric related to Interactivity. FID refers to how long it takes a site to react to a user interacting with it (for example, clicking a button or scrolling.)

The most common causes of poor FID are:

1. A lot of heavy JavaScript execution on page load
2. Interactive elements that rely on JavaScript

1. A lot of heavy JavaScript

ISSUE

The browser cannot respond to user interactions while the main thread is busy. (This tends to last longer on a site that uses client-side rendering.)

FIXES

A) Optimize your page for interaction readiness:

There are a number of common causes for poor FID and Total Blocking Time (TBT) scores in web apps that rely heavily on JavaScript:

- First-party script execution can delay interaction readiness.
- Data-fetching can impact aspects of interaction readiness.
- Third-party script execution can also delay interaction readiness.

Any of these may benefit from the fixes below:

B) Break up Long Tasks:

If you've already attempted to reduce the amount of JavaScript that loads on a single page, it can be useful to break down long-running code into smaller tasks.

Long Tasks are JavaScript execution periods where users may find elements to be unresponsive. Any piece of code that blocks the main thread for 50 ms or more can be characterized as a Long Task. Long Tasks are a sign of potential JavaScript bloat (loading and executing more than a user may need at first). Splitting up long tasks can reduce input delay on your site.

C) Use a web worker:

A blocked main thread is one of the main causes of input delay. Web workers make it possible to run JavaScript on a background thread. Moving non-UI operations to a separate worker thread can cut down main thread blocking time and consequently improve FID.

D) Reduce JavaScript execution time:

Limiting the amount of JavaScript on your page reduces the amount of time that the browser needs to spend executing JavaScript code. This speeds up how fast the browser can begin to respond to any user interactions.

To reduce the amount of JavaScript executed on your page:

- Defer unused JavaScript
- Minimize unused polyfills

2. Interactive elements that rely on JavaScript for their behavior

ISSUE

Some custom actionable elements rely on JavaScript event listeners to respond to user input. If the JavaScript that handles these has not been loaded or executed yet (even if the main thread has finished), then the elements will still be non-interactive. If these are important elements, the user is more likely to experience a delay with them.

FIXES

First, use native, semantic actionable elements wherever possible. (For example, if clicking an element causes the browser to navigate to a different URL, use an actual `a` element with an `href` attribute, instead of a button that's powered by JavaScript. This way, the browser can handle the navigation as soon as the user clicks the element, with no JavaScript required.)

If controls still require JavaScript (which is inevitable sometimes), consider separating the needed scripts and preloading them (or consider deferring loading for all other scripts)

Tools for Measuring FID

Lighthouse 6.0 does not include support for FID since it is a field metric. However, Total Blocking Time (TBT) can be used as a proxy. Optimizations that improve TBT should also improve FID in the field.

Chrome User Experience Report provides real-world FID values aggregated at the origin-level.

Okay. Deep breaths. If none of this is making sense, we can help! Our audit services will get you on track. Get connected at ClassyLlama.com/contact to get help.

PART 4.3

How to Fix CLS

CLS (Cumulative Layout Shift) relates to Visual Stability. Layout shifts, or elements moving around as they're loading, can be pretty distracting to users. When a user attempts to click on an element on the page and suddenly it shifts to a different position, preventing the click, it can be jarring and frustrating.

The most common causes of poor CLS are:

1. Images without dimensions
2. Ads, embeds, and iFrames without dimensions
3. Dynamically injected content
4. Web Fonts causing FOIT/FOUT
5. Actions waiting for network response before updating DOM

1. Images without Dimensions

ISSUE

Image elements are loaded on the page before the actual image itself is loaded, so unless the size is explicit, the browser doesn't know how much space to give the image element until after the image content is loaded separately.

FIXES

Include width and height size attributes on your image and video elements. This ensures that the browser can allocate the correct amount of space in the document while the media is loading.

Set a default aspect ratio for images, which will help prevent layout shifts. If you are having a hard time understanding aspect ratios, aspectratiocalculator.com is a helpful tool.

2. Ads, Embeds, and iFrames without Dimensions

ISSUES

Ads are one of the largest culprits to layout shifts, due to dynamic sizes, which often causes them to push visible content down the page as they're loading.

FIXES

Statically reserve the largest possible size space for the ad slot by styling the element before the ad tag library loads, or choose the most likely size for the ad slot based on historical data.

When the ad slot is visible, show a placeholder to avoid collapsing the reserved space if there is no ad returned.

Embeddable widgets allow you to embed content in a page (e.g. Google Maps or videos from YouTube). These embeds often aren't aware in advance just how large an embed will be. As a result, platforms offering embeds do not always reserve enough space for their embeds and can cause layout shifts when they finally load. To work around this, you can minimize CLS by precalculating sufficient space for embeds as a placeholder.

iFrames have the same problem as images, where the browser can't know how much space to allocate until after the inner content is loaded.

3. Dynamically Injected Content

ISSUE

Most users have experienced layout shifts due to the website elements that pop in at the top or bottom of the viewport when you're trying to load a site. Similar to ads, this often happens with banners and forms that shift the rest of the page's content.

FIX

If you need to display these types of UI affordances, reserve sufficient space in the viewport for it in advance such as with a placeholder so that when it loads, it does not cause content in the page to shift around unexpectedly.

4. Web Fonts Causing FOUT/FOIT

ISSUE

When a web font is used, the browser has to download it, just like other external resources. Meanwhile, the browser defaults to a system ("fallback") font. After a moment (or several), when the web font is loaded, the font style suddenly changes. This can have one or both of two effects:

1. Flash of Unstyled Text (FOUT): The fallback font not only looks stylistically different, but likely has different proportions, causing text to wrap at different points when the fallback is swapped with the web font.
2. Flash of Invisible Text (FOIT): Sometimes, the text is hidden completely until the web font is rendered.

FIX

One way to mitigate this effect is to include the custom font files directly on your server whenever possible, along with your other resources (such as CSS and JavaScript).

This typically allows the browser to load them more quickly than when it has to connect to a 3rd-party service.

5. Features Waiting for Network Response Before Updating DOM

ISSUE

Some content loads via AJAX, which is a technique that opens a new request to the server (even after the rest of the page is done loading). When there's a visible content update that depends on the conclusion of this request, there will be an inevitable delay while waiting for the server response. (AJAX is used heavily in client-side rendering.)

FIX

First, consider using client-side rendering where possible, to minimize the amount of content being loaded via AJAX.

If AJAX is required (or desired for other performance reasons, such as reducing initial page load), consider implementing a loading indicator, or some other kind of placeholder, that reserves the necessary space for the pending content.

Tools for Measuring CLS

Lighthouse 6.0 and above include support for measuring CLS in a lab setting. This release will also highlight the nodes that cause the most layout shifting.

Want some help fixing your CLS results? Get connected with our experts at ClassyLlama.com/contact.

Wrapping Up

Where do I go from here?

Phew! Was that enough information or what? Yikes!

Right now, the Core Web Vitals Update might seem insanely, annoyingly, worryingly complicated. If you're feeling panicked about what to do to help your business, we totally get it.

But here's the good news: We want to help you with Core Web Vitals, and it doesn't have to cost you a dime.

We do NOT want to see Google's update mess up your business so we've developed an arsenal of tools to get you ready:

Free Report

Send us your URL and get a free Core Web Vitals Readiness Report straight to your inbox so you can find out what issues you have.

Webinars

We're hosting some free webinars where experts in Core Web Vitals will give you the skinny on everything you need to know and answer your questions.

Paid Guidance and Fixes

We also have paid services that include things like correcting issues on your site for you, or bulking up your SEO.

You don't have to go through this on your own!
Classy Llama is here to help.

Click one of the options below to get started:

GET A FREE WEBSITE REPORT

REGISTER FOR A WEBINAR

CORE WEB VITALS SERVICES

Or email us at sales@classyllama.com.