BRIGHTEDGE

The Global Leader in Enterprise
SEO and Content Performance

Google Page Experience Update
Understanding Core Web Vitals

What is the May 2021 Google Page Experience Update?

It is a major update to Google's algorithms that will bring a wide variety of different metrics together to form a *page experience ranking factor,* including the new metrics that Google is calling Core Web Vitals as additional ranking factors.

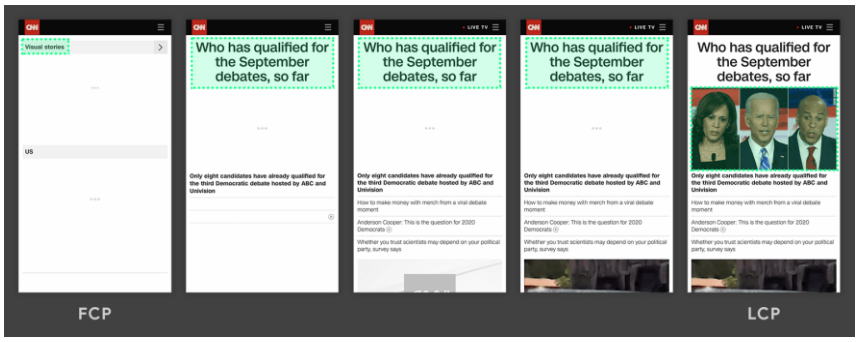
What are Core Web Vitals?

Core Web Vitals are three metrics that score a user's experience loading a webpage. These metrics score how quickly page content loads, how quickly a browser loading a webpage can respond to a user's input, and how unstable the content is as it loads in the browser. These three metrics are going to be bundled alongside Mobile Friendliness, Safe Browsing, HTTPS and Intrusive Interstitials into a signal Google is calling the "Page Experience Signal".

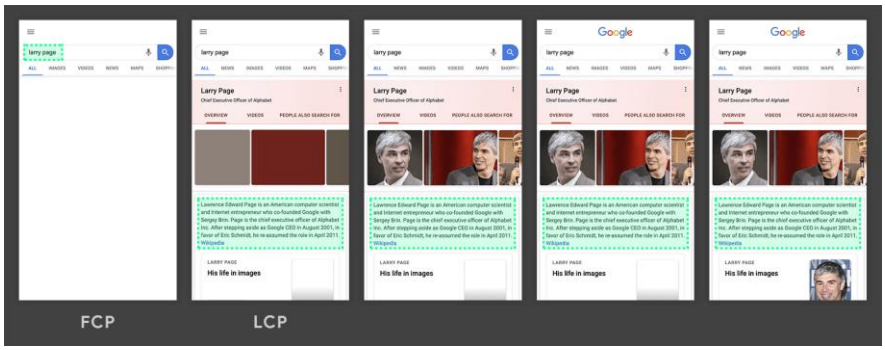
LCP – Largest Contentful Paint

This metric scores the time in which a page's main content, for example the largest visible image or text block, has rendered. If you have a large image on your site, or a video background that takes a long time to load, you'll likely need to improve this metric.

The two images below do a good job of showing what the browser considers the largest element on the page (highlighted in green). In the first example, the H1 is the largest element until it's time to load the image, in which case it is considered the largest contentful paint. This is an important distinction, as webmasters have been told for years to delay the load times for large elements. If that large element is still being loaded "above the fold" it counts against the LCP.



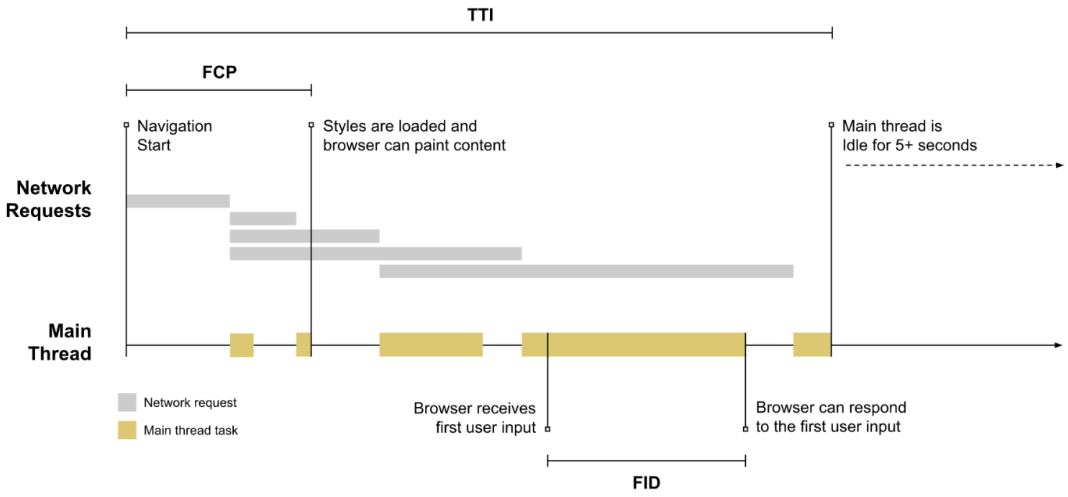
Example of delayed LCP



Example of immediate LCP

FID - First Input Delay

First input delay (FID) measures when a user first interacts with a webpage to when the browser can respond to that actions. It helps quantify user frustration related to the moment a user sees a visual element into a user-centric metric.

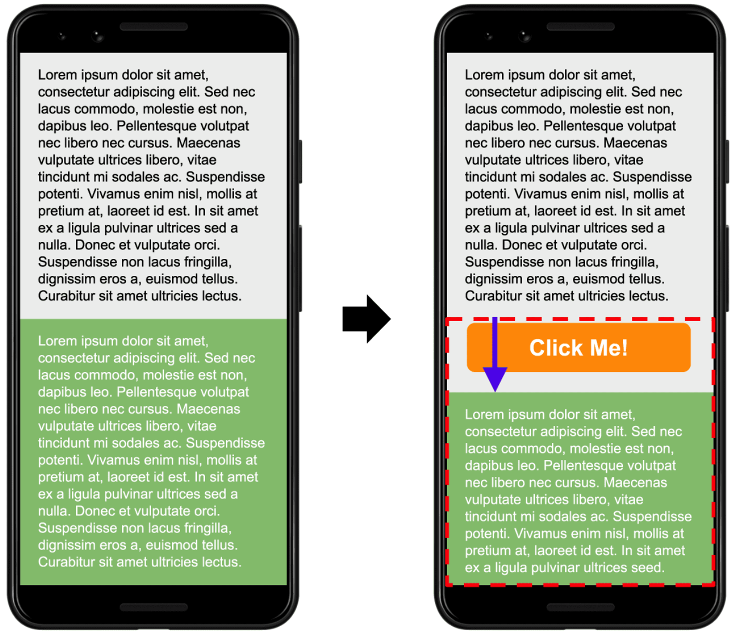


CLS - Cumulative Layout Shift

This measures when resources and content are being called on to the page after critical content (text) has loaded causing the content to shift, without first being predicated by a user interaction with the page.

Images that are without defined heights and widths and/or have been deferred to after critical content is loaded, ads that push down content after critical content is loaded, and sidebar widgets that push the main article to the left or right, are common causes for a negative CLS score.

In the image at right, you can see a common instance of a layout shift. On the left, the content is loaded, and the user can start interacting with the page. On the right, a button that was loaded after much of the page was rendered shifts the content in the green area down the page.



Example of unexpected layout shift.

Fixing your Core Web Vitals

It would be rare for two separate sites to suffer from the same issues, so it's important to focus on analyzing each domain individually to prioritize updates that will be the most beneficial.

There are common issues many websites will need to contend with. Here is a short list of fixes for common issues:

Common activities to address LCP

- Apply instant loading with the PRPL pattern
  - **P**re-render most important resources
  - **R**ender the initial route as soon as possible
  - **P**re-cache remaining assets
  - **L**azy load other routes and non-critical assets
- Optimizing your Critical Rendering Path
  - minimize number of critical resources
  - reduce the total number of roundtrips to load page
  - download all critical assets as early as possible
- Optimize CSS files
  - minify the code
  - use gzip
  - remove unused CSS
- Optimize image file sizes and compression
  - Resize images down to the largest pixel size it will be displayed at on screen
  - Keep image file sizes to below 100kb whenever possible

    - Make use of next-gen image formats for optimum image compression ratios
- Optimize or remove web fonts
- Optimize or reduce your JavaScript (for client-rendered sites)
    - minify the code
    - use gzip
    - remove unused JavaScript

Common activities to address FID

- Reduce third-party code
    - Assess widgets, ad blocks, plugins, etc.
    - Remove code that doesn't add clear value to your site or is redundant
    - Use async or defer on <script tags> when appropriate
    - Use rel="preconnect" when establishing a connection to a third-party domain
        - Use rel="dns-prefetch" for less critical third-party domains
- Reduce JavaScript execution time
    - Only send the code that your users need by implementing code splitting
    - Minify and compress your code
    - Remove unused code
    - Reduce network trips by caching your code with the PRPL pattern
- Minimize main thread work
    - Optimize third-party JavaScript
    - Debounce your input handlers
    - Use web workers
- Keep request counts low and transfer sizes small

Common activities to address CLS

- Include the size attributes on your images and video elements or reserve the space with CSS aspect ratio boxes
- Never insert content above existing content, except in response to a user interaction
- Use transform animations instead of animations of properties that force layout changes